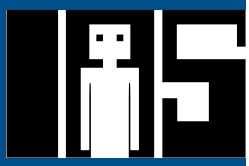
Bayesian Inference for Regression Models using Nonparametric Infinite Mixtures

Bayessche Inferenz für Regressionsmodelle mit Unendlichen Parameterfreien Mischverteilungen

Master-Thesis von Peter Nickl aus Weiden i. d. Opf.

Januar 2020

TECHNISCHE
UNIVERSITÄT
DARMSTADT

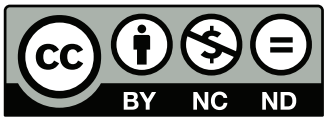


Bayesian Inference for Regression Models using Nonparametric Infinite Mixtures
Bayessche Inferenz für Regressionsmodelle mit Unendlichen Parameterfreien Mischverteilungen

Vorgelegte Master-Thesis von Peter Nickl aus Weiden i. d. Opf.

1. Gutachten: Prof. Jan Peters, Ph.D.
2. Gutachten: M.Sc. Hany Abdulsamad
3. Gutachten: Prof. Dr.-Ing. Matthias Weigold, M.Sc. Stephan Bay

Tag der Einreichung:



Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, _____ , die vorliegende Master-Thesis / Bachelor-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

English translation for information purposes only:

Thesis Statement pursuant to § 22 paragraph 7 and § 23 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, _____ , have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.

I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.

In the submitted thesis the written copies and the electronic version for archiving are pursuant to § 23 paragraph 7 of APB identical in content.

Matrikelnummer:	E-Mail (privat):
_____	_____
Datum / Date:	Unterschrift/Signature:
_____	_____

Abstract

In this thesis, we deal with Bayesian inference methods for regression using nonparametric infinite mixtures. The regression method we are working with is called Dirichlet Process Mixtures of Generalized Linear Models (DP-GLM). The general idea of this approach is to cluster inputs that share the same relation to the associated outputs. In each of the clusters, a separate linear regression model is learned. A Dirichlet process prior is used to infer the number of clusters and thus the number of linear models from the data. DP-GLM can be regarded as an infinite mixture of linear regression models. As we put priors on all parameters of the regression model, a quantification of the uncertainty of the predictions can be achieved. The downside of this approach is that the posterior is too complex to be inferred analytically. Therefore, we use the Bayesian inference methods of variational inference and Gibbs sampling. Gibbs sampling is a stochastic method for generating samples from the true posterior by iteratively sampling from the conditional posteriors of the parameters. Variational inference is a deterministic approximation scheme. We use the mean field assumption to calculate iterative update equations of a factorized approximate posterior. The resulting algorithm has a similar structure to the expectation maximization algorithm. The approximate posterior is used to derive a posterior predictive distribution for predicting on new input data. We show that the combination of variational inference and Gibbs sampling for learning the parameters of DP-GLM leads to good prediction accuracy on a broad range of different datasets. Among others, we use the method for learning the forward kinematics of a robot arm with simulated data, as well as for learning the inverse dynamics of a real SARCOS robot arm.

Zusammenfassung

In dieser Arbeit beschäftigen wir uns mit Bayes'schen Inferenzmethoden für Regression mit unendlichen parameterfreien Mischverteilungen. Die Regressionsmethode, die in dieser Arbeit im Fokus steht, heißt Dirichlet-Prozess-Mischverteilungen aus generalisierten linearen Modellen (DP-GLM). Die Idee dieses Ansatzes ist es, Eingangsdaten zu clustern, die ein ähnliches Verhältnis zu den zugehörigen Zielvariablen haben. In jedem der Cluster wird ein separates lineares Regressionsmodell gelernt. Wir verwenden den Dirichlet-Prozess, um die Anzahl der Cluster automatisch aus den Daten abzuleiten. DP-GLM kann als eine unendliche Mischverteilung aus linearen Regressionsmodellen betrachtet werden. Da alle Parameter des Regressionsmodells als Zufallsvariablen behandelt werden, kann eine Quantifizierung der Unsicherheit von Vorhersagen für Testdaten erreicht werden. Der Nachteil dieses Ansatzes ist, dass die A-posteriori-Verteilung der Parameter zu komplex ist, um analytisch berechnet zu werden. Wir verwenden daher die Bayes'schen Inferenzmethoden der Variationsinferenz und des Gibbs-Stichprobenverfahrens. Das Gibbs-Stichprobenverfahren ist eine stochastische Methode zur Stichprobenauswahl aus der wahren A-posteriori-Verteilung durch iteratives Ziehen aus der A-posteriori-Verteilung der Parameter, konditioniert auf die restlichen Parameter. Variationsinferenz ist eine deterministische Approximationsmethode. Wir verwenden die Molekularfeldannahme, um iterative Aktualisierungsgleichungen einer faktorisierten approximativen A-posteriori-Verteilung zu bestimmen. Die Struktur des resultierenden Algorithmus ist ähnlich zur Struktur des Erwartungs-Maximierungs-Algorithmus. Wir leiten eine prädiktive Verteilung für die Vorhersage neuer Eingangsdaten ab. Wir zeigen, dass die Kombination von Variationsinferenz und des Gibbs-Stichprobenverfahrens zum Erlernen der A-posteriori-Verteilung von DP-GLM zu einer guten Vorhersagegenauigkeit bei einer Vielzahl von verschiedenen Datensätzen führt. Unter anderem verwenden wir das Verfahren zum Erlernen der Vorwärtskinematik eines Roboterarms mit simulierten Daten, sowie zum Erlernen der inversen Dynamik eines echten SARCOS Roboterarms.

Contents

Figures and Tables

List of Figures

List of Tables

Abbreviations

List of Abbreviations

Notation	Description
CRP	Chinese restaurant process
DP	Dirichlet process
DP-GLM	Dirichlet process mixtures of generalized linear models
DP-GMM	Dirichlet process Gaussian mixture model
ELBO	Evidence lower bound
EM	Expectation maximization algorithm
GLM	Generalized linear models
GMM	Gaussian mixture model
GP	Gaussian process
I-projection	Information projection
i.i.d.	Independent and identically distributed
KL	Kullback-Leibler divergence
LWPR	Locally weighted projection regression
LWR	Locally weighted regression
M-projection	Moment projection
MCMC	Markov chain Monte Carlo
MOE	Mixture of experts
RFWR	Receptive fields weighted regression
VB	Variational Bayes
VBEM	Variational Bayes expectation maximization
VI	Variational inference
VLB	Variational lower bound

1 Introduction

In recent years, as more and more applications are becoming mature, machine learning and the field of artificial intelligence in general have become quite visible in our daily lives and the public dialogue. An outstanding example of this fact is the tremendous progress in bringing fully autonomous cars to the streets. In 2018 Waymo LLC, a subsidiary of Alphabet Inc., started the first commercial self-driving taxi service in Arizona, USA [1]. Other increasingly apparent applications are chat-bots, computer vision (e.g for quality assurance in a production setting), speech recognition systems, and intelligent robots. Behind all these efforts there is the goal to enable machines to learn from past experience and to be able to execute complex tasks with human or even superhuman performance.

The increased economic and academic activity in machine learning to a large extent is due to the sharp increase in the performance of deep artificial neural networks as universal function approximators. In 2012 AlexNet [2], a convolutional neural network, lowered the error of image classification on the ImageNet dataset [3] significantly. This brought the feasibility of deep neural networks for a broad range of tasks to the attention of a wider audience. Subsequently, other impressive applications followed. In 2015 AlphaGo became the first computer program to beat a professional human on a full Go board with 19 by 19 fields by using deep neural networks and Monte Carlo tree search [4]. This was a significant step as the game previously had been considered too complex to be mastered by a machine.

Despite the recent successes of neural networks in these and many other applications, in the past research was focused on solving pragmatic practical tasks rather than the theoretical understanding of how deep networks can achieves such accurate predictions. Additionally these networks suffer from different problems, e.g. overfitting the test data, which raises the need to use regularization techniques like dropout. In contrast to that, the Bayesian theory of machine learning has a very strong theoretical foundation. By putting priors on the model parameters, Bayesian regression methods are able to quantify the uncertainty of predictions. For proper choice of the priors, Bayesian methods are a principled approach to automatic regularization and model selection. Due to these strengths of the Bayesian approach there is a trend towards Bayesian neural networks, where uncertainty is incorporated in the weights of the neurons. This is still an active area of research.

In this thesis, as an alternative approach to neural networks as function approximators, we focus on a Bayesian regression method called Dirichlet Process Mixtures of Generalized Linear Models (DP-GLM) [5]. In contrast to neural networks, the uncertainty of predictions is incorporated in the model formulation naturally. The idea of DP-GLM is to model a complex response distribution by simpler locally linear models. First the input data is clustered in regions, where inputs have the same kind of relation to the associated output. Then a separate linear model is learned for each cluster. When making predictions the complexity of the predictive distribution arises by marginalizing out which local model is active in which region of the inputs. The number of clusters can be automatically inferred from data using the Dirichlet process as a prior, without specifying the number of linear models in advance as a model assumption. This approach enables the model to be just as complex as required by the data. As the number of clusters is theoretically unbounded and the number of clusters does not need to be specified in advance, such clustering methods are referred to as nonparametric infinite mixtures. As a Bayesian method DP-GLM naturally delivers an uncertainty of the predicted targets.
A second major advantage of DP-GLM is that the regression method can represent input-dependent variance. As variance can vary across the linear models, it can be a function of the inputs in the global regression estimate. This makes DP-GLM a good choice for heteroscedastic datasets and distinguishes the method from other state-of-the-art function approximators like Gaussian processes (GPs). In the standard version of the algorithm, GPs can not model input-dependent variance. Furthermore, due to the use of a Kernel function for obtaining the covariance matrix, standard GPs are limited to smooth functions. By using locally linear models DP-GLM allows a greater flexibility in modelling discontinuous functions.
As a further advantage of DP-GLM it is worth mentioning that linear models have a certain appeal in a control context. In contrast to neural networks as function approximators, linear models retain the ability to interpret the results of learned models from a classical theoretical control perspective.

1.1 Contributions

The ultimate goal of DP-GLM is to make predictions on test data. As DP-GLM is a Bayesian regression method, we need to evaluate the joint posterior distribution of the parameters for obtaining the posterior predictive distribution. In DP-GLM

there are many parameters, that are treated as random variables and have to be learned. The parameters are the location and scale of the clusters, the regression parameters of the linear models, as well as the mixing weights. There are also latent cluster assignment variables. The amount and potentially high dimensionality of the hidden variables prohibit to calculate the evidence in Bayes' formula for obtaining the posterior analytically. The integration over all possible values of the parameters is intractable. Due to this reason we use the Bayesian inference techniques of Gibbs sampling and mean field variational inference.

Variational inference (VI) [6] is a deterministic method that approximates the true posterior by a tractable, but simplified distribution. A common simplification is the mean field assumption, which states that the posterior is fully factorized. By introducing such assumptions, an approximate posterior can be obtained analytically. To our knowledge variational inference has not been applied to the DP-GLM regression method before. The mean field variational inference algorithm is an iterative procedure that has a similar structure as the expectation maximization algorithm. For obtaining the update equations of the factors of the approximate posterior, we extend the variational Bayesian treatment of Gaussian mixture models [7, 8] to an infinite mixture of linear regression models as in DP-GLM. We show our derivations in detail.

The second inference method we are concerned with is Gibbs sampling. Gibbs sampling belongs to the class of Markov chain Monte Carlo (MCMC) methods. In contrast to variational inference it is a stochastic sampling procedure. The idea is to iteratively draw samples from the conditional distributions of the parameters. After a burn-in period this procedure converges to generating samples from the true joint posterior. Inaccuracies can arise as it is a stochastic method and only a finite number of samples from the posterior can be generated. Furthermore the convergence of the algorithm is hard to assess. One contribution of this thesis is to derive and implement a Gibbs sampling algorithm for DP-GLM. We extend a Gibbs sampling algorithm of Gaussian mixture models [9, 10, 11] to the case of an infinite mixture of regression models. A similar collapsed Gibbs sampling algorithm has been applied in the original DP-GLM paper [5].

We evaluate our method on different datasets that are chosen to demonstrate the broad applicability of DP-GLM in combination with the implemented inference algorithms. In our experiments we use Gibbs sampling as an initialization for the mean field variational inference algorithm. VI is prone to getting stuck in bad local optima and thus is dependent on good initializations. A weakness of Gibbs sampling, on the other hand, is that the convergence is hard to assess. This is no problem in mean field VI, as a quantity called variational lower bound can be used as a convergence criterion for the algorithm. By Gibbs sampling as initialization different regions of the area can be explored. The samples will be from different modes of the posterior. Variational inference then locks on the mode it has been initialized on. If the algorithm is started multiple times, this approach lowers the risk of mean field VI to get stuck in bad local optima.

We examine the prediction accuracy of the algorithm for different settings of the hyperparameters and for different sample sizes. We use a highly heteroscedastic dataset of the cosmic microwave background [12, 5] to demonstrate the capability of DP-GLM to cover input-dependent noise. We also use simulated sinus data, simulated forward kinematics data of a robot arm with one joint simulated forward kinematics data of a robot arm with three joints. We additionally evaluate the prediction accuracy of DP-GLM on the inverse dynamics dataset of a real SARCOS robot arm [13, 14, 15].

1.2 Overview

The structure of the thesis is as follows.
In chapter 2 we introduce foundations from the field of machine learning that serve as a basis for the rest of the thesis. We first motivate the Bayesian approach to regression by comparing the approach to classical least-squares regression. A section about Bayesian linear regression shows which priors can be assumed for the mean and the variance of the linear models of DP-GLM. Basics on graphical models are the prerequisites for understanding the graphical models that occur in this thesis. In the end of the foundational chapter we review finite mixture models and describe how we can extend finite mixtures to nonparametric infinite mixtures using the Dirichlet process. We introduce different representations of the Dirichlet process, such as the stick-breaking construction, the Pólya urn sampling scheme and the Chinese restaurant process.

In chapter 3 we gather several different ideas from literature for regression with locally linear models. Locally weighted regression is an approach that started in the statistics community and subsequently was used for control purposes. Locally weighted projection regression is an improvement of locally weighted regression to be viable in an high-dimensional online learning setting. In the mixture of experts model several experts (e.g. linear models) specialize in certain regions of the input. In the section on DP-GLM we discuss the probabilistic generative model and the ideas behind this regression model.

In the chapters 4 and 5 we outline the Bayesian inference methods of Gibbs sampling and variational inference. For both approaches we start with a general definition of the inference methods. We then sketch the inference algorithms for the finite and infinite Gaussian mixture model and show how to extend these results to the case of an infinite mixture of regression models. For mean field variational inference our derivations are shown in detail. In chapter 4 we show how to calculate the posterior predictive distribution using the approximate posterior that is obtained by the VI algorithm.

In chapter 6 we apply DP-GLM to different data sets. The algorithm uses Gibbs sampling as an initialization for mean field variational inference. The main criteria for assessing the algorithm are the prediction accuracy on test data, as well as the number of linear models that are used to achieve this accuracy.

In chapter 7 we summarize the content of this thesis and give an outlook on potential future research questions.

2 Foundations

2.1 Motivating the Bayesian Approach to Regression

This thesis builds upon the definition of *Bayesian* probabilities, which stands in contrast to the *classical* or *frequentist* interpretation of probability. In the classical definition probabilities are seen as frequencies of repeatable random events. From the Bayesian point of view probabilities quantify a degree of uncertainty of the outcome of a random variable. [7, 8, 16]
We want to exemplify the benefits of the Bayesian approach by means of a simple regression problem. We are given a training set $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ of N observations of input variables x together with corresponding observations $\boldsymbol{t} \equiv (t_1, \ldots, t_N)^{\mathrm{T}}$ of the target values t. The aim is to make a prediction $\hat{y}$ for new input data $\hat{x}$. We consider the following polynomial regression model:

$$y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j + \epsilon$$

$\boldsymbol{w} = (w_0, \ldots, w_M)^{\mathrm{T}}$ are the regression parameters, M is the degree of the polynomial function and ϵ is the noise term that is assumed to be normally distributed. In order to find parameter values for fitting the function to observed data we can minimize an error function. In the frequentist approach of least squares regression the aim is to minimize the squared deviations of the estimated targets $y(x_n, \boldsymbol{w})$ from the ground truths y_n. The following error function is commonly used:

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}) - y_n\}^2$$

In our example this error function can be evaluated analytically in closed form. Specifying the order of the polynomial here remains an open question. One would be able to achieve a perfect fit to the training data and a diminishing error, if the degree of the polynomial is sufficiently large. In this case the parameters are calculated such that the function merely interpolates the training data. This phenomenon is referred to as overfitting and leads to poor performance in generalizing to unseen test data. Figure 2.1 illustrates this issue. In order to avoid overfitting one needs to use regularization terms to the error function that penalize if the model fits the training data too closely.

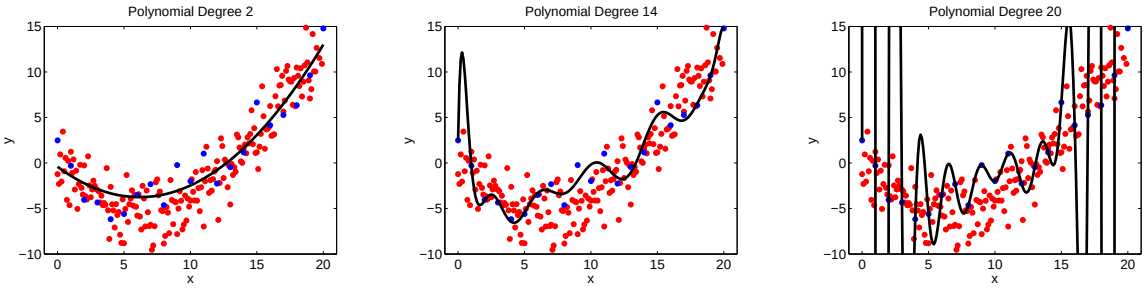
The Bayesian approach alleviates the problem of overfitting in a more principled way. In the Bayesian framework we capture assumptions on the regression coefficients $\boldsymbol{w}$ before having seen any data in the form of a prior probability distribution $p(\boldsymbol{w})$. We model the effect of the observed data through the conditional probability distribution $p(y|x, \boldsymbol{w})$, which is called *likelihood* function. The likelihood expresses how probable the given observation are for different parameter values. Given the likelihood and the prior, we can use Bayes' theorem to quantify the uncertainty in the parameter values after having observed data:

$$p(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w}) \, \mathrm{d}\boldsymbol{w}} \tag{2.1}$$

The quantity $p(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{y})$ on the left hand side is called the *posterior* distribution and is normalized to a proper probability distribution by dividing by the *evidence* in the denominator. This normalization is necessary as the likelihood in general does not fulfil the requirements of a probability density. The likelihood does not sum to one if one does a summation over the data given the parameters. In words one can state Bayes' theorem as follows:

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

For proper choices of the prior $p(\boldsymbol{w})$ the aforementioned problem of overfitting of the frequentist approach can be reduced. The prior on the parameters acts as a regularizer and restricts the values, that the learned parameter values can take to stay close to the prior belief. If more training data is taken into account, the effect of the prior on the posterior is reduced as the importance of the likelihood grows. According to equation 2.1 we average over all possible parameter values to obtain the posterior, thus making extreme outcomes for the parameters unlikely. The frequentist method



(a) (b) (c)

Figure 2.1.: The plots show training data (blue) and test data (red). A polynomial regression function is fitted to the test data using a maximum likelihood estimate. (a) A polynomial degree of 2 generalizes well from the training data to the test data. (b) A polynomial degree of 14 results in overfitting. The regression function is fitted too closely to the training data and does not generalize well to test data. (c) For a polynomial degree of 20 the maximum likelihood approach leads to severe overfitting and a mere interpolation of the training data. Figures created with the `linregPolyVsDegree` Skript of the Pmtk3 probabilistic modeling toolkit for Matlab of Kevin Murphy [17].

of least-squares, on the other hand, is equivalent to a maximum likelihood estimation. By maximizing the likelihood $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$ the most probable parameter setting $\boldsymbol{w}^\star$ given the data is obtained, which is equivalent to trying to explain the training data as good as possible. The Bayesian method however allows to quantify the uncertainty in the parameter values instead of only focusing on the most probable value.

In a regression setting the uncertainty in the parameter value leads to a quantification of the uncertainty in the prediction $\hat{y}$ for test data $\hat{x}$, if we integrate over all possible parameter values. The posterior predictive distribution of a target $\hat{y}$ for a new observation $\hat{x}$ reflects this:

$$p(\hat{y}|\hat{x},\boldsymbol{x},\boldsymbol{y}) = \int p(\hat{y}|\hat{x},\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\,\boldsymbol{w} \tag{2.2}$$

The posterior predictive distribution is the distribution over possible unobserved targets conditional on the observed training data. In equation 2.2 we obtain the predictive distribution of $\hat{y}$ by marginalizing the distribution of $\hat{y}$ given the test input $\hat{x}$ and the parameters over the posterior of the parameters given the training data $\boldsymbol{x},\boldsymbol{y}$.
We now have stated two major advantages of the Bayesian framework. The marginalization over the parameters enables automatic regularization and we are able to quantify the uncertainty of predictions for test data.

Equation 2.1 also reveals the shortcomings of the Bayesian approach. Due the complexity of the underlying probability distributions the integration over the whole parameter space might be intractable. The Bayesian inference methods of Markov chain Monte Carlo, for which Gibbs sampling is a representative, and variational inference have been developed to solve this issue. In Gibbs sampling we can obtain samples from the posterior distribution by iteratively sampling from the tractable marginal distributions. In variational inference we obtain an approximate posterior distribution by making simplifying assumptions on the true posterior, such as a factorized structure. In this way we can make the involved integrations analytically tractable. These Bayesian inference methods will be introduced in detail in Chapter 4 and Chapter 5. [7, 8]

2.2 Bayesian Linear Regression

After motivating the Bayesian approach to regression, we want to introduce Bayesian linear regression in a formal way. We choose a very general problem statement, where there can be multiple inputs and outputs of arbitrary dimension and the noise variance is unknown. An input vector $\boldsymbol{x}$ of length m is multiplied with a regression coefficient matrix $\boldsymbol{\beta}$ for generating an output vector $\boldsymbol{y}$ of length d. A Gaussian noise term $\boldsymbol{\epsilon}$ is added: [18]

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{\beta}\boldsymbol{x} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \mathrm{N}(0,\boldsymbol{V}) \end{aligned} \tag{2.3}$$

This data model can be written as a likelihood using the following conditional probability distribution:

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{V}) \sim \mathrm{N}(\boldsymbol{\beta}\boldsymbol{x},\boldsymbol{V}) \tag{2.4}$$

We are given a data set of exchangeable pairs $D = \{(\boldsymbol{y}_1, \boldsymbol{x}_1), \ldots, (\boldsymbol{y}_N, \boldsymbol{x}_N)\}$. We collect the matrices $\boldsymbol{Y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_N]$ and $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_N]$. The distribution of Y given X then is:

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{V}) &= \prod_i p\left(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\beta}, \boldsymbol{V}\right) \\ &= \frac{1}{|2\pi\boldsymbol{V}|^{N/2}} \exp\left(-\frac{1}{2}\sum_i (\boldsymbol{y}_i - \boldsymbol{\beta}\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{V}^{-1} (\boldsymbol{y}_i - \boldsymbol{\beta}\boldsymbol{x}_i)\right) \end{aligned}$$

We can use the matrix-normal density as a prior on the unknown parameter matrix $\boldsymbol{\beta}$. The matrix-normal density is conjugate to the likelihood. By using this distribution it is thus ensured that the posterior for $\boldsymbol{\beta}$ as well as the predictive posterior for $\boldsymbol{Y}$ are also matrix-normal distributed.
A random d by m matrix $\boldsymbol{\beta}$ is matrix-normal distributed with parameters $\boldsymbol{M}_0$, $\boldsymbol{V}$ and $\boldsymbol{K}_0$ if the density of $\boldsymbol{\beta}$ is:

$$\begin{aligned} p(\boldsymbol{\beta}) &\sim \mathrm{N}(\boldsymbol{M}_0, \boldsymbol{V}, \boldsymbol{K}_0) \\ &= \frac{|\boldsymbol{K}_0|^{d/2}}{|2\pi\boldsymbol{V}|^{m/2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left((\boldsymbol{\beta} - \boldsymbol{M}_0)^{\mathrm{T}}\boldsymbol{V}^{-1}(\boldsymbol{\beta} - \boldsymbol{M}_0)\boldsymbol{K}_0\right)\right) \end{aligned} \tag{2.5}$$

where $\boldsymbol{M}_0$ is d by m, $\boldsymbol{V}$ is d by d, and $\boldsymbol{K}_0$ is m by m. $\boldsymbol{M}_0$ is the mean. The two matrices $\boldsymbol{V}$ and $\boldsymbol{K}_0$ are covariances matrices for the rows and the columns of $\boldsymbol{\beta}$.
In the most general case we also consider $\boldsymbol{V}$ of our model in equation 2.4 as unknown and put an inverse Wishart prior on it:

$$\begin{aligned} p(\boldsymbol{V}) &\sim \mathrm{W}^{-1}(\boldsymbol{S}_0, n) \\ &= \frac{1}{Z_{nd}|\boldsymbol{V}|^{(d+1)/2}} \left|\frac{\boldsymbol{V}^{-1}\boldsymbol{S}_0}{2}\right|^{n/2} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{V}^{-1}\boldsymbol{S}_0\right)\right) \\ \text{where } Z_{nd} &= \pi^{d(d-1)/4} \prod_{i=1}^{d} \Gamma((n+1-i)/2) \end{aligned} \tag{2.6}$$

The inverse Wishart prior is conjugate to the likelihood 2.4 so that the posterior of $\boldsymbol{V}$ will also be an inverse Wishart distribution. After specifiying the priors and the likelihood of the mode, we can find the posterior distributions of both $\boldsymbol{\beta}$ and $\boldsymbol{V}$.
For notational convenience we first define the following expressions:

$$\begin{aligned} \boldsymbol{S}_{xx} &= \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + \boldsymbol{K}_0 \\ \boldsymbol{S}_{yx} &= \boldsymbol{Y}\boldsymbol{X}^{\mathrm{T}} + \boldsymbol{M}_0\boldsymbol{K}_0 \\ \boldsymbol{S}_{yy} &= \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_0^{\mathrm{T}} \\ \boldsymbol{S}_{y|x} &= \boldsymbol{S}_{yy} - \boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{yx}^{\mathrm{T}} \end{aligned}$$

We get the posterior for $\boldsymbol{\beta}$ by taking the likelihood (2.4) times the conjugate matrix-normal prior (2.5):

$$\begin{aligned} p(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{V}) &\propto \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{V}^{-1}\left[\boldsymbol{\beta}\boldsymbol{S}_{xx}\boldsymbol{\beta}^{\mathrm{T}} - 2\boldsymbol{S}_{yx}\boldsymbol{\beta}^{\mathrm{T}} + \boldsymbol{S}_{yy}\right]\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{V}^{-1}\left[\left(\boldsymbol{\beta} - \boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\right)\boldsymbol{S}_{xx}\left(\boldsymbol{\beta} - \boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1} + \boldsymbol{S}_{y|x}\right]\right)\right)\right) \end{aligned}$$

:

We can see that the result is a matrix-normal posterior for $\boldsymbol{\beta}$ with the following parameters:

$$p(\boldsymbol{\beta}|D, \boldsymbol{V}) \sim \mathrm{N}\left(\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}, \boldsymbol{V}, \boldsymbol{S}_{xx}\right) \tag{2.7}$$

The posterior of $\boldsymbol{V}$ is obtained by taking the marginal likelihood times the inverse Wishart prior (2.6).

$$p(\boldsymbol{Y}, \boldsymbol{V}|\boldsymbol{X}) \propto \frac{1}{|\boldsymbol{V}|^{(d+1)/2}|2\pi\boldsymbol{V}|^{(N+N_0)/2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{V}^{-1}\left(\boldsymbol{S}_{y|\boldsymbol{x}} + \boldsymbol{S}_0\right)\right)\right)$$

This posterior for $\boldsymbol{V}$ again has the form of an inverse Wishart distribution:

$$p(\boldsymbol{V}|D) \sim \mathrm{W}^{-1}\left(\boldsymbol{S}_{y|x} + \boldsymbol{S}_0, N + N_0\right) \tag{2.8}$$

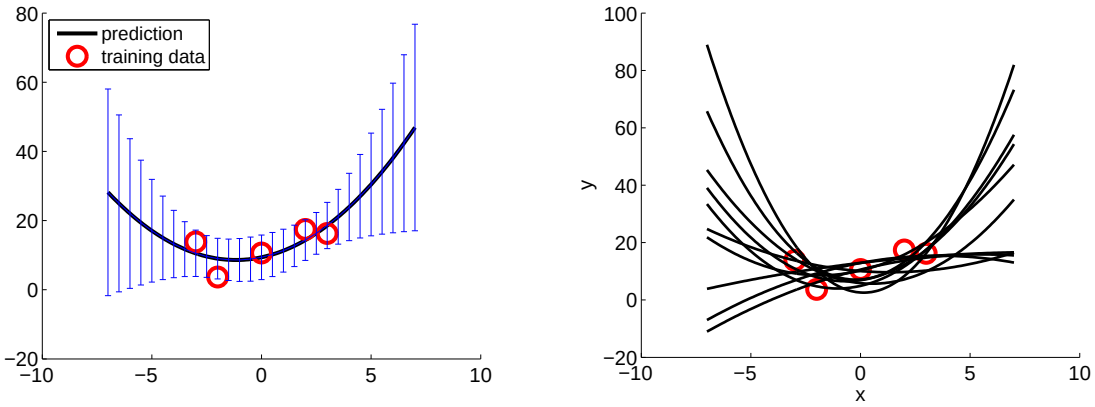


Figure 2.2.: The plots show the posterior predictive density of a Bayesian linear regression model. (a) Mean of the posterior predictive density (black line) with error bars of 2 standard deviations (blue bars). The red dots are training data points. One can observe that the uncertainty of the prediction is higher in regions with few data. (b) The black lines are 10 samples from the posterior predictive. The variance in the samples again reflects the uncertainty of the prediction far away from the training data. Figures created with the `linregPostPredDemo` Skript of the Pmtk3 probabilistic modeling toolkit for Matlab of Kevin Murphy [17].

Our ultimate aim in Bayesian regression is to make predictions $\hat{\boldsymbol{y}}$ for new input data $\hat{\boldsymbol{x}}$. We can get the posterior predictive distribution by combining the definition of $\boldsymbol{y}$ (2.3), the posterior of $\boldsymbol{\beta}$ (2.7) and the posterior of $\boldsymbol{V}$ (2.8).
We first combine the definition $\boldsymbol{y}$ and the posterior of $\boldsymbol{\beta}$ to:

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, D, \boldsymbol{V}) = \mathrm{N}\left(\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\hat{\boldsymbol{x}}, \boldsymbol{V}c^{-1}\right)$$
$$c = \left(1 + \hat{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{S}_{xx}^{-1}\hat{\boldsymbol{x}}\right)^{-1} = 1 - \hat{\boldsymbol{x}}^{\mathrm{T}}\left(\boldsymbol{S}_{xx} + \hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{T}}\right)^{-1}\hat{\boldsymbol{x}}$$

This expression is the predictive posterior distribution for $\hat{\boldsymbol{y}}$, if the noise variance $\boldsymbol{V}$ is known. We now multiply this by the posterior for $\boldsymbol{V}$ and integrate out $\boldsymbol{V}$, in order to get the posterior predictive:

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, D) = \mathrm{T}\left(\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\hat{\boldsymbol{x}}, \left(\boldsymbol{S}_{y|x} + \boldsymbol{S}_0\right)c^{-1}, N + N_0 + 1\right)$$

The result has the form of a Matrix t-distribution. [18]
For chapter 4 we introduce the following notation for the parameters of the inverse Wishart posterior and the matrix-normal posterior:

$$\boldsymbol{L}_k = \boldsymbol{S}_{y|x} + \boldsymbol{S}_0$$
$$\nu_k = N + N_0$$
$$\boldsymbol{M}_k = \boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}$$
$$\boldsymbol{K}_k = \boldsymbol{S}_{xx}$$

We thus can rewrite the Matrix t-distribution:

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, D) = \mathrm{T}\left(\boldsymbol{M}_k\hat{\boldsymbol{x}}, \boldsymbol{L}_k c^{-1}, \nu_k + 1\right) \tag{2.9}$$
$$c = 1 - \hat{\boldsymbol{x}}^{\mathrm{T}}\left(\boldsymbol{K}_k + \hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{T}}\right)^{-1}\hat{\boldsymbol{x}} \tag{2.10}$$

Figure 2.2 shows a simple example of Bayesian linear regression, where both the input and outputs are one-dimensional and the noise variance is assumed to be known. We can observe the advantages of the Bayesian approach in comparison with a maximum likelihood estimate. The predictions have a quantification of uncertainty and the model is not prone to overfitting.

2.3 Graphical Models

Graphical models [19] are visual representations of probabilistic models. They provide a simple way to visualize the structure of the probabilistic model and encode properties like conditional independence.

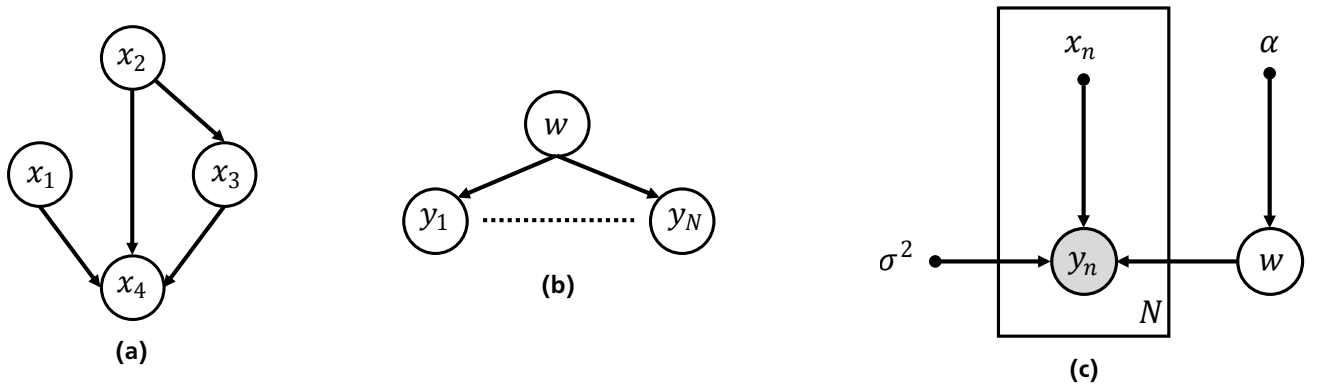


Figure 2.3.: Three examples of graphical models. (a) A simple graphical model that encodes the joint distribution of four random variables. Random variables are denoted as large circles. (b) Graphical model of the joint distribution of random variables of a Bayesian polynomial regression model. $\boldsymbol{w}$ are the regression parameters and y_n are the targets. (c) This graphical model shows the plate notation that is suitable for summarizing i.i.d. random variables. Deterministic parameters are displayed as small solid circles. Observed random variables are shaded in grey. Based on figures 8.2, 8.3 and 8.6 of [7].

We here want to focus on directed graphical models, which are denoted *Bayesian networks*. By the chain rule of probabilities any joint distribution can be written as a product of conditional distributions, one for each of K variables: [7]

$$p(x_1, \ldots, x_K) = p(x_K|x_1, \ldots, x_{K-1}) \ldots p(x_2|x_1)\, p(x_1)$$

This can be displayed as directed graph with K nodes, where each node represents one random variable. Directed links (arrows) which are pointing towards a node correspond to the variables that the distribution of that node is conditioned on.
A Bayesian network does not need to be fully connected as we can see in Figure 2.3a. From this graph we can read off the following joint distribution:

$$p\left(x_1, \ldots, x_4\right) = p(x_1)\, p(x_2)\, p(x_3|x_2)\, p\left(x_4|x_1, x_2, x_3\right)$$

The relationship between a Bayesian network and the corresponding distribution over variables can be stated in general terms for a graph with K nodes as follows:

$$p(\boldsymbol{x}) = \prod_{k=1}^{K} p\left(x_k|\,\mathrm{pa}_k\right)$$

pa_k here denotes the set of parent nodes of the node k and $\boldsymbol{x} = \{x_1, \ldots, x_K\}$.
We use specific example of polynomial regression, which we already have introduced in section 2.1, in order to illustrate graphical models. We define the polynomial coefficients $\boldsymbol{w}$, the input data $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ and the observed targets $\boldsymbol{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$. We want to treat the problem in a Bayesian way and put a Gaussian prior with precision α over the regression coefficients $\boldsymbol{w}$. The noise variance is σ^2.
The joint distribution of the random variables variables of the model, y and $\boldsymbol{w}$, can now be written as follows:

$$p\left(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}, \alpha, \sigma^2\right) = p(\boldsymbol{w}|\alpha) \prod_{n=1}^{N} p\left(y_n|\boldsymbol{w}, x_n, \sigma^2\right)$$

Figure 2.3b represents this joint distribution in a graphical model. Since all targets $y_1, \ldots, y_N$ are random variables and need to be included in the graph, the visualization can get cluttered for more complex models. Due to this reason there is a simplifying *plate* notation, which contains a single representative node y_n that summarizes N nodes of this kind. A further convention is that random variables are denoted by large open circles, whereas deterministic parameters are indicated by small solid circles. In a supervised learning scenario there will typically be some observed variables during training. These are marked by shading the respective nodes. In the regression case the values of the targets are observed during training, whereas the parameters $\boldsymbol{w}$ are examples for *latent* or *hidden* variables, that are not observed. Following these conventions we end up with the graphical model in Figure 2.3c for the regression model. [7]

2.4 Finite Mixture Models

Mixture models [20] are taking linear combinations of basic distributions such as Gaussians, in order to create more complex distributions. These can in turn be used for density estimation and clustering. By using a sufficient number of Gaussians and inferring their means and covariances, almost any continuous density can be approximated accurately. The mixture models we are concerned with in this section are finite in the sense that the number of mixture components K needs to be specified. [7] Mixture models are an instance of *latent variable models*, which apart from observed variables also take hidden variables into account. The formulation of a mixture model for a random variable $\boldsymbol{x}_i$is: [8]

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x}_i|\boldsymbol{\theta}_k) \tag{2.11}$$

The latent variables in this model are the discrete indicator variables $z_i \in \{1,\dots,K\}$, which indicate the corresponding mixture component. A Categorical distribution $p(z_i) = \text{Cat}(\boldsymbol{\pi})$ is used as a discrete prior for the indicator variables. In equation 2.11 we used the following abbreviations:

$$\begin{aligned} \pi_k &= p(z_i = k|\boldsymbol{\pi}) \\ p_k(\boldsymbol{x}_i|\boldsymbol{\theta}_k) &= p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\theta}) \end{aligned}$$

p_k is the k^{th} base distribution, which can be of arbitrary type and is parametrized by parameters $\boldsymbol{\theta}_k$.
Equation 2.11 is a weighted sum of the p_k's . The *mixing weights* have to satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.
The most common example of mixture model is the *Gaussian mixture model* (GMM), where each base distribution is a multivariate Gaussian with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. The model can thus be written as:

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \,\text{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

In clustering applications we first fit the mixture model and then compute $p(z_i = k|\boldsymbol{x}_i, \boldsymbol{\theta})$, which is the posterior probability that point i belongs to cluster k. This posterior probability is referred to as responsibilty of cluster k for point i. It can be computed using Bayes' rule: [8]

$$r_{ik} \triangleq p(z_i = k|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\theta})\, p(z_i = k|\boldsymbol{\theta})}{\sum_{k'=1}^{K} p(\boldsymbol{x}_i|z_i = k', \boldsymbol{\theta})\, p(z_i = k'|\boldsymbol{\theta})}$$

By using responsibilites we do not hard-assign a datapoint to one cluster, but for each of the clusters calculate the posterior probability that a datum belongs to the respective cluster. That is why the approach is called *soft clustering*. Iterating between the steps of fitting the parameters of the components by maximum likelihood and weighting the data with the responsibilites and calculating new responsibilities comprises the expectation maximization algorithm (EM) [21].
It is possible to extend the mixture models by putting a prior H with hyperparameters λ on the parameters $\boldsymbol{\theta}$ of the components and a Dirichlet distribution prior with hyperparameter α on the mixing weights $\boldsymbol{\pi}$, so that:

$$p(\boldsymbol{\pi}|\alpha) = \text{Dir}(\alpha/K)\mathbf{1}_K)$$

$\mathbf{1}_K$ is a unit vector with length K. The generative probabilistic model, which describes, how a datapoint can be sampled from the model, can finally be written as: [8, 22]

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dir}((\alpha/K)\mathbf{1}_K) \\ z_i &\sim \text{Cat}(\boldsymbol{\pi}) \\ \boldsymbol{\theta}_k &\sim H(\lambda) \\ \boldsymbol{x}_i &\sim F\left(\boldsymbol{\theta}_{z_i}\right) \end{aligned}$$

We have here written $p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$ as $\boldsymbol{x}_i \sim F\left(\boldsymbol{\theta}_{z_i}\right)$, where F is the observation distribution. For this generative model we get a graphical model as in Figure 2.4a.

A central problem to mixture models is to choose the appropriate number of components K. In order to do so model selection methods for probabilistic models can be used. If K is not chosen properly, this might result in over- or underfitting of the data as there is a misfit between the complexity of the model and the amount of available data. An alternative way of dealing with the problem is to use a *Dirichlet process* prior on the mixing weights, which automatically infers the number of components from the data. In this case instead of the EM algorithm Bayesian inference methods like Gibbs sampling and variational inference need to be used. [8, 23]
If the mixture components are probability densities for target variables and are conditioned on the inputs, and the mixture coefficients are input-dependent, we get the regression method of mixture of experts (see chapter 3.3).

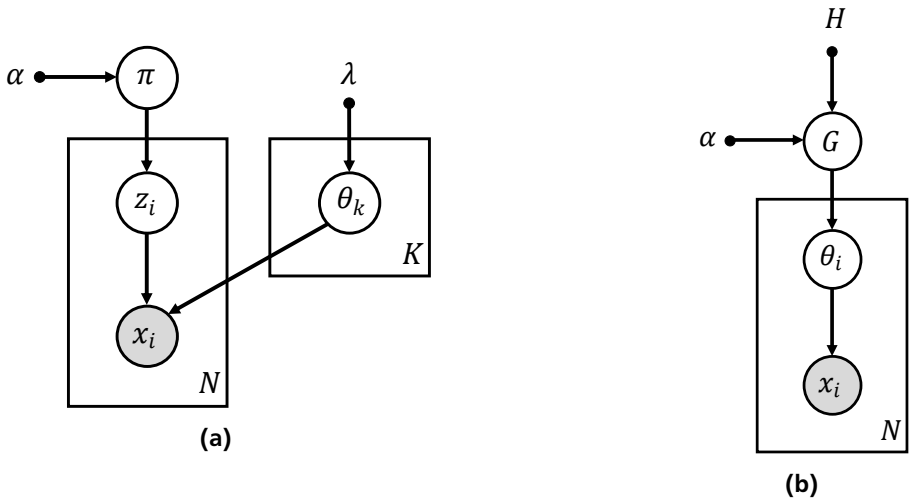


Figure 2.4.: Two different graphical models for finite mixture models. Assuming a Gaussian mixture, the parameters $\boldsymbol{\theta}$ are a mean and a covariance matrix. (a) In the standard representation the component parameters $\boldsymbol{\theta}_k$ are drawn from a prior H with hyperparameters λ, separately for every component k. The latent variables z_i assign N data points x_i to a component and are drawn from a Categorical distribution. The Categorical distribution is parametrized by the mixture weights π. The mixture weights are drawn from a Dirichlet distribution $\mathrm{Dir}((\alpha/K)\mathbf{1}_K)$. (b) In the alternative representation the parameters $\boldsymbol{\theta}_i$ are drawn from a finite, discrete distribution G. G is a weighted sum of delta distributions. α determines the structure of the weight vector π. The delta distributions are located at the values of $\boldsymbol{\theta}$ that are drawn from the prior distribution H. For each datum we have a separate $\boldsymbol{\theta}_i$. In the alternative representation the coincidence of parameters of several data points indicates that these data points belong to the same components. Based on figure 25.2 in [8] and figure 2.9 in [22].

2.5 Dirichlet Process for Infinite Mixture Models

As mentioned before, finite mixture models can be extended to infinite ones by using the *Dirichlet process* (DP) [24, 23, 25] as a prior. In this section, we introduce the Dirichlet process and its' several representations.

2.5.1 From Finite to Infinite Mixture Models

In section 2.4 we have introduced the standard representation of a finite mixture model, that is shown in 2.4a.
There is an alternative representation of finite mixture models, that is suitable for introducing infinite mixtures. In this representation the parameters that are used to generate an observation $\boldsymbol{x}_i$ are sampled from a discrete distribution G of the form: [8]

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta})$$

We can observe that G is a finite mixture of delta distributions $\delta_{\boldsymbol{\theta}_k}$. The delta distribution $\delta_{\boldsymbol{\theta}_k}$ is zero everywhere, except for the value θ_k, where it is one. For each observation $\boldsymbol{x}_i$ a own set of parameters is sampled. As G is a discrete distribution, the parameters of the observations will frequently take the same values. If the same parameters are drawn for two observations, they are associated with the same cluster. The probability that $\boldsymbol{\theta}_i$ is equal to the parameters of a certain cluster k is π_k. This alternative representation of finite mixture models is depicted in 2.4b.
[8] In this model we still always get exactly K clusters. In order to obtain a more flexible model, where a variable number of clusters can be generated, we can use the Dirichlet Process for generating the discrete distribution G:

$$G \sim \mathrm{DP}(\alpha, H)$$

We will explain this in more detail in the following sections.

2.5.2 Definition of the Dirichlet Process

We begin to introduce the *Dirichlet Process* (DP) as a generalization of the Dirichlet distribution by a formal definition. A Dirichlet process is a distribution over probability measures $G : \Theta \to \mathbb{R}^+$, where $G(\theta) \geq 0$ and $\int_\Theta G(\theta) d\theta = 1$. The DP is defined by the requirement that $(G(T_1), \ldots, G(T_K))$ has the a joint Dirichlet distribution

$$\mathrm{Dir}(\alpha H(T_1), \ldots, \alpha H(T_K))$$

for any finite partition $(T_1, \ldots, T_K)$ of Θ.
If this requirement is fulfilled we can write $G \sim \mathrm{DP}(\alpha, H)$, where α is the *concentration parameter* and H is called *base measure*.
Intuitively the Dirichlet process is a distribution over distributions, which means that each draw G from the Dirichlet process is itself a distribution. It is called Dirichlet process as its' finite dimensional marginal distributions are Dirichlet distributed. This is analogous to the Gaussian process, which has Gaussian disitributed finite dimensional marginal distributions. The draws from a DP are discrete distributions. As G cannot be described with a finite number of parameters, the DP is classified as a nonparametric model. [23]
The base distribution H is the mean of the DP, as for any $T \subset \Theta$ we have $E[G(T)] = H(T)$. The concentration parameter α can be interpreted as an inverse variance: $V[G(T)] = H(T)(1-H(T))/(\alpha+1)$. The larger α, the smaller the variance and the DP will concentrate more of its mass around the mean H. For $\alpha \to \infty$ we will have $G(T) \to H(T)$. α is also called strength parameter, as it determines the strength of the prior, when DP is used as a prior in a Bayesian nonparametric model. [23, 8]

2.5.3 Stick-breaking Construction of the Dirichlet Process

The *stick-breaking construction* [26] of the DP can be used for algorithmic realisations of the DP, for example in variational inference or Gibbs sampling algorithms. Using the stick-breaking construction we can derive an infinite sequence of mixture weights π_k of an infinite mixture model from the following probabilistic process: [8]

$$\begin{aligned} \nu_k &\sim \mathrm{Beta}(1, \alpha) \\ \pi_k &= \nu_k \prod_{l=1}^{k-1} (1 - \nu_l) = \nu_k \left(1 - \sum_{l=1}^{k-1} \pi_l\right) \end{aligned}$$

This process is sometimes denoted as:

$$\pi \sim \mathrm{GEM}(\alpha)$$

If the parameter α increases, the size of the π_k components decreases on average. The stick-breaking procedure describes how the ν_k that are drawn from a Beta distribution, are combined to obtain the mixture weights π_k.
We now again use the definition of G from 2.5.1, but set the number of clusters to infinity:

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta})$$

Here we use $\pi \sim \mathrm{GEM}(\alpha)$ and $\boldsymbol{\theta}_k \sim H$. It has been shown that $G \sim \mathrm{DP}(\alpha, H)$ and the stick-breaking construction thus is a representation of the Dirichlet process. [8]
As in the case of finite mixture models, the distribution G, which itself is drawn from a Dirichlet process, is a mixture of delta functions and is discrete. This again means, if the parameters $\boldsymbol{\theta}$ that we draw from G take on the same value for observations $\boldsymbol{x}_i$, these observations belong to the same cluster. [8]
An extension of the basic stick-breaking construction is the logistic stick-breaking process [27]. It generalizes to applications, where the data is ought to be clustered spatially or temporally.

2.5.4 Pólya Urn Sampling Scheme

For sampling algorithms like Gibbs sampling often the *Pólya urn* or *Blackwell-MacQueen* sampling scheme is used [28]. It describes how to draw samples from a DP. If there are N observations drawn from G, that take on $k = 1 \ldots K$ different values $\boldsymbol{\theta}_k$, the predictive distribution of the next draw is given by [8]

$$p(\boldsymbol{\theta}_{N+1} = \boldsymbol{\theta} | \boldsymbol{\theta}_{1:N}, \alpha, H) = \frac{1}{\alpha + N} \left(\alpha H(\boldsymbol{\theta}) + \sum_{k=1}^{K} n_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta}) \right) \tag{2.12}$$

where n_k is the number of previous observations equal to $\boldsymbol{\theta}_k$.
Equation 2.12 is helpful for thinking about the impact of the choice of the prior parameter α in an infinite mixture model on the number of clusters. If α is large, it gets more likely that a new $\boldsymbol{\theta}_{N+1}$ gets drawn from the base distribution H instead of taking on previously seen values. For larger α it is more likely to draw a value that has not seen before. If that is the case, a new cluster k is created. This also underlines why α is denoted as strength parameter, indicating the increasing influence of the base distribution H for larger α.

We can use indicator variables z_i, which define which value of $\boldsymbol{\theta}_k$ to use for the next draw. This is equivalent to defining $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{z_i}$. Using this definition we get for the next draw of an indicator, which specifies which parameters $\boldsymbol{\theta}_k$ to use, the following expression:

$$p\left(z_{N+1} = z | \boldsymbol{z}_{1:N}, \alpha\right) = \frac{1}{\alpha + N}\left(\alpha \mathbb{I}(z = k^*) + \sum_{k=1}^{K} n_k \mathbb{I}(z = k)\right) \tag{2.13}$$

$\mathbb{I}$ is the indicator function and $k^\star$ is a cluster index of a cluster that so far no data points are assigned to.
Equation 2.13 is referred to as *Chinese restaurant process* (CRP), comparing the assignment of cluster variables $\boldsymbol{\theta}_k$ of a new draw from G to the assignment of tables to customers in a Chinese restaurant. The analogy states that the tables are like clusters and the customers are like observations. If a customer enters the restaurant the customer will join an existing table with a probability proportional to the people already being seated at this table N_k. This shows the *rich get richer* property of the Dirichlet process. If tables (clusters) already have a large number of customers (observations) they are more likely to be assigned even more customers. With a probability proportional to α the customer will be seated at a completely new table with index $k^\star$. The probability of being assigned to a new table reduces for the overall number of customers in the restaurant. This is due to the $1/(\alpha + N)$ term. It has been shown that the number of overall clusters grows logarithmically with the size of the dataset and almost surely approaches $\alpha \log(N)$ for $N \to \infty$. In Gibbs sampling algorithms the CRP is used for obtaining the probability of an observation belonging to one of the existing clusters or to a newly created one. [8]

2.5.5 Nonparametric Infinite Mixture Models

After the introduction to the theory of the Dirichlet process and its' representations we can extend the generative model of finite mixtures from section 2.4 to the infinite mixture case [29, 30]. The generative model of nonparametric infinite mixtures is: [8]

$$\begin{aligned}
\boldsymbol{\pi} &\sim \mathrm{GEM}(\alpha) \\
z_i &\sim \mathrm{Cat}(\boldsymbol{\pi}) \\
\boldsymbol{\theta}_k &\sim H(\lambda) \\
\boldsymbol{x}_i &\sim F\left(\boldsymbol{\theta}_{z_i}\right)
\end{aligned}$$

This generative model is depicted in Figure 2.5. In contrast to the finite mixture case G is now a random draw of an unbounded number of parameters $\boldsymbol{\theta}_k$ from the base distribution H, as discussed for the stick-breaking construction. Each data point $\boldsymbol{x}_i$ is generated by sampling an own parameter $\boldsymbol{\theta}_i$. Due to the rich get richer property for increasing amount of data a new draw of an of observation is likely to have the same parameters as one of the $\boldsymbol{\theta}_k$ that were drawn previously. In this way a clustering effect is being achieved. New datapoints will likely be generated close to existing ones. There is a probability proportional to α that a new datapoint will be drawn according to completely new cluster parameters that are drawn the base measure H. If the observations distribution F takes on the form of a Gaussian, we call the model Dirichlet process Gaussian mixture model (DP-GMM) in this thesis. [8]

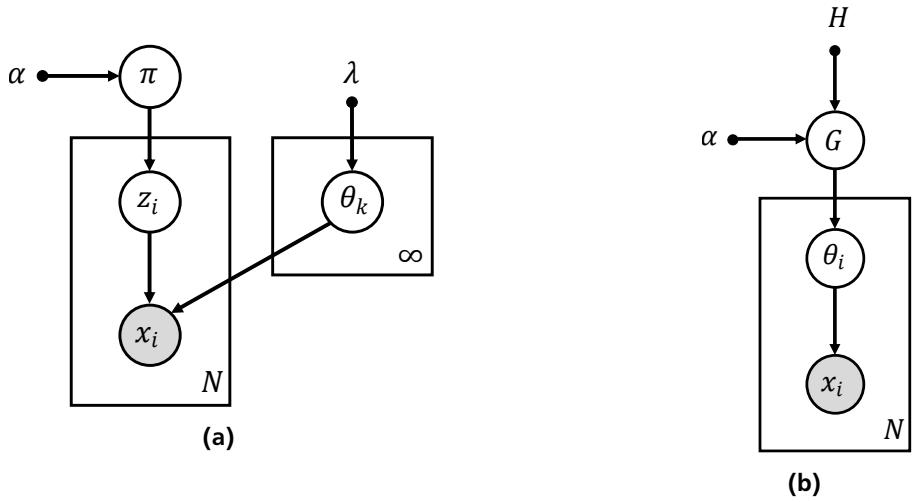


Figure 2.5.: Two different graphical models for infinite mixture models. Assuming a Gaussian mixture, the parameters $\boldsymbol{\theta}$ are a mean and a covariance matrix. (a) The first graphical model depicts the stick-breaking representation and is largely identical to finite mixture models. The only difference is that the mixture weights $\boldsymbol{pi}$ are drawn from a stick-breaking prior that is parametrized by α. As a result of this procedure the number of components is unbounded. (b) In the Pólya Urn representation the parameters $\boldsymbol{\theta}_i$ are drawn from the infinite, discrete distribution G. G is drawn from a Dirichlet process $\mathrm{DP}(\alpha, H)$ with concentration parameter α and base measure H. As in the case of finite mixtures two data points belong to the same cluster, if their parameters take on the same value. Based on figure 25.6 in [8] and figure 2.9 in [22].

3 Regression with Locally Linear Models

In this chapter we review different regression methods with locally linear models. The general idea of these regression approaches is that they use linear sub-models, which specialize in certain regions of the input space. The mechanism of how to choose which models are dominant in which regions of the input space and the learning and prediction procedures differ significantly among the reviewed methods.
We discuss Locally Weighted Regression, Locally Weighted Projection Regression and Mixture of Experts. We then proceed with Dirichlet Process Mixtures of Generalized Linear Models, which is the method we are mainly concerned with in this thesis.

3.1 Memory-Based Locally Weighted Regression

Locally Weighted Regression (LWR), which also has been called local polynomial regression [31, 32], first appeared in the statistical literature [33, 34]. It subsequently also was adopted for control purposes [35, 36]. The regression method estimates the target $\boldsymbol{y}$ at an input $\boldsymbol{x}$ by only considering observations in the neighborhood of $\boldsymbol{x}$. This concept is opposed to global regression, where all the data is used for fitting a global regression curve. Examples for the global function approximation approach are least squares regression, support vector machines, neural networks and Gaussian process regression [37].

We limit our review to locally *linear* weighted regression. This is a generalization of the local fit of a constant as in the Nadaraya-Watson kernel estimator [31, 32]. As some authors argue, linear models are a good trade-off of bias and variance. Locally constant models suffer from bias at the boundaries, whereas higher-order local polynomials lead to more variable estimates. We thus limit the scope of this thesis to the use of locally linear models and do not use higher-order local polynomials or locally fitted constants. [34, 38]

LWR deals with the standard regression problem:

$$y = f(\boldsymbol{x}) + \epsilon$$

$\boldsymbol{x}$ is a n-dimensional input vector, the noise term ϵ has zero mean and for simplicity the output y is kept one-dimensional. The main idea of LWR is to approximate a nonlinear function by piecewise linear models. Key challenges here are to determine the region of validity of the separate linear models and to fit the models within these regions. LWR computes the region of validity through a Gaussian kernel:

$$w_k = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}_k)^T \boldsymbol{D}_k (\boldsymbol{x}-\boldsymbol{c}_k)\right)$$

$\boldsymbol{c}_k$ is the center of the k^{th} linear model and is set to the query point $\boldsymbol{x}_q$ at which a prediction is ought to be made. D_k is a positive semi-definite distance metric, that determines the size and shape of the region of validity of the linear model k.
Algorithm 1 shows how to predict for a query point $\boldsymbol{x}_q$ [39, 37]. It has to be executed once for each query point. After prediction, the regression coefficient $\boldsymbol{\beta}_q$ of the local linear model is discarded. The computational complexity of the algorithm is proportional to $O(np)$, with p denoting the amount of training data and n denoting the input dimensions. β_{n+1} is the $(n+1)^{\text{th}}$ element of the vector β.
The only remaining parameter to determine is the distance metric $\boldsymbol{D}_k$, which can be optimized by leave-one-out cross-validation. The detailed pseudo-algorithm for the leave-one-out cross validation can be found in [39]. To avoid too many parameters $\boldsymbol{D}_k$ can be assumed to be a diagonal matrix in the form of $\boldsymbol{D}_k = h \cdot \text{diag}(n_1, n_2, \cdots, n_n)$, where h is a scale parameter and the n_i normalize the range of the input dimensions. Doing so, one reduces the leave-one-out cross-validation to a one-dimensional search over the parameter h. [39]
The determination of the distance metric is critical for the bias-variance trade-off of the model. In case the kernel is too narrow, it starts fitting noise. If the kernel is too broad the fitted curve will be too smooth. One can either use the same $\boldsymbol{D}_k$ for the entire input space or optimize locally for the query point. [37]

Advantages of LWR are low bias and high accuracy due to the use of local linear models and few parameters that have to be obtained by cross validation.

Input : Query point $\boldsymbol{x}_q$, p training points $\{\boldsymbol{x}_i, y_i\}$ in memory
Output: Prediction $\hat{y}_q$ for $\boldsymbol{x}_q$

1 Compute diagonal weight matrix $\boldsymbol{W}$ with diagonal elements

$$w_{ii} = \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{x}_q\right)^T \boldsymbol{D}_k \left(\boldsymbol{x}_i - \boldsymbol{x}_q\right)\right)$$

2 Build matrix $\boldsymbol{X}$ and vector y such that

$$\boldsymbol{X} = \left(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \ldots, \tilde{\boldsymbol{x}}_p\right)^T \text{ where } \tilde{\boldsymbol{x}}_i = \left[\left(\boldsymbol{x}_i - \boldsymbol{x}_q\right)^T 1\right]^T$$
$$\boldsymbol{y} = \left(y_{1,}, y_2, \ldots, y_p\right)^T$$

3 Compute regression coefficients of locally linear model

$$\boldsymbol{\beta}_q = \left(\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y} \tag{3.1}$$

4 The prediction for $\boldsymbol{x}_q$ is

$$\hat{y}_q = \begin{bmatrix} \boldsymbol{x}_q^T & 1 \end{bmatrix} \boldsymbol{\beta}_q$$

Algorithm 1: Predicting with Locally Weighted Regression

LWR is a so called "lazy" approach as all training data needs to be stored in memory and the processing of the training data is done when prediction is made. If the learning system receives a large, possibly never ending stream of input data, both the memory requirements to store all data as well as the computational cost for prediction become expensive. This is a common observation in online robot applications. A non-memory-based version of the algorithm is desirable, so that each datum can be incrementally included in the learning system. [39]
A further weakness of LWR is the inability to scale well to high-dimensional input. This is due to the covariance matrix inversion in equation 3.1. LWR also becomes computationally expensive as the number of linear model increases. [37]

3.2 Locally Weighted Projection Regression

An extension to a memoryless version of LWR, which is better suited for online applications, is the Receptive Fields Weighted Regression (RFWR) algorithm [40]. It is memoryless in the sense of keeping each model in memory for further prediction and thus eliminating the need to store all training data points. New models are only created in case the activation weight of the existing models does not exceed a certain threshold. Adding the second component of applying dimensionality reduction to the input space one ends up with the Locally Weighted Projection Regression (LWPR) algorithm [14, 41]. LWPR can not only deal with incrementally arriving data but also scales well to high-dimensional inputs as it removes redundant input dimensions. It utilizes several locally weighted linear models to approximate a globally non-linear function. The main ideas for extending LWR to the memoryless version RFWR are as follows [37]:

- New kernels are not created for each new datapoint, but only if no existing kernel in memory covers the training data point with some minimal activation weight.
- A weighted regression is updated with weighted recursive least squares for new training points $\{\boldsymbol{x}, y\}$:

$$\boldsymbol{\beta}_k^{n+1} = \boldsymbol{\beta}_k^n + w\boldsymbol{P}^{n+1}\tilde{\boldsymbol{x}}\left(t - \tilde{\boldsymbol{x}}^T\boldsymbol{\beta}_k^n\right)$$
$$\text{with } \boldsymbol{P}_k^{n+1} = \frac{1}{\lambda}\left(\boldsymbol{P}_k^n - \frac{\boldsymbol{P}_k^n\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T\boldsymbol{P}_k^n}{\frac{\lambda}{w_k} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}_k^n\tilde{\boldsymbol{x}}}\right)$$
$$\text{and } \tilde{\boldsymbol{x}} = \left[\mathrm{x}^T 1\right]^T$$

- The distance metric for each kernel $\boldsymbol{D}_k$ is determined by gradient descent employing leave-one-out cross validation.

- The prediction for a query point is calculated as a weighted average of the predictions of all local models:

$$y_q = \frac{\sum_{k=1}^{K} w_{q,k} \hat{y}_{q,k}}{\sum_{k=1}^{K} w_{q,k}}$$

In above equations $\lambda \in [0, 1]$ determines how much old data in the regression parameters will be forgotten. $\boldsymbol{P}_k$ caches the inverse of the covariance matrix of the input variables.

The second major problem of LWR, namely extending the regression method to high dimensional inputs, can be addressed by local dimensionality reduction techniques [14]. LWPR makes use of Partial Least Squares to eliminate subspaces of the input that do not have a high correlation with the output. Redundant and irrelevant dimensions are identified and removed. Due to this technique LWPR can scale up very well and has been used for learning control problems with over 100 input dimensions. The prediction, the update for one local model and the complete LWPR algorithm can be found in [42]. The complete LWPR algorithm has a complexity of $O(n)$ [37].

3.3 Mixture of Experts

The Mixture of Experts (MOE) [43] model is a modular model for regression based on soft probabilistic splits of the input space. It can be though of as a probabilistic mixture of densities, where the component densities are conditional. This is in contrast to standard mixture models (i.e. Gaussian mixtures), where the densities are unconditional. Additionally, in MOE the mixing coefficients are a function of the inputs. The resulting probabilistic model thus is [7]:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}) p_k(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_k) \tag{3.2}$$

In the context of MOE the mixing coefficients $\pi_k(\boldsymbol{x})$ are called gating functions and the conditional distributions $p_k(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_k)$ are called experts. The experts have parameters $\boldsymbol{\theta}_k$. According to equation 3.2 the prediction of the model is a weighted average of the predictions of the individual components [8]. The gating functions determine, which components of the mixture are dominant in which region of the input. They have to fulfill the following conditions [7]:

$$0 \leqslant \pi_k(\boldsymbol{x}) \leqslant 1 \text{ and } \sum_k \pi_k(\boldsymbol{x}) = 1$$

One common way to represent the mixing coefficients that meets the above criteria is using linear softmax models of the form [7]:

$$\pi_k(\boldsymbol{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \text{ with } a_k = \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x})$$

$\phi(\boldsymbol{x})$ here is a linear transformation of the input and $\boldsymbol{w}_k$ are gating parameters that need to be learnt.
The experts $p_k(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_k)$ are specialized in making predictions in certain regions of the input. Among others the experts can be conditioned Gaussians, Gaussian processes [44], support vector machines or neural networks. As we are dealing with locally linear models, conditioned Gaussians are a natural choice. We show the generative model as a graphical model in Figure 3.1a. [45]

If both the gating and the experts are linear, the model can be fitted efficiently with the EM algorithm [46]. An extension of the MOE is the Hierarchical Mixture of Experts (HME) architecture using multilevel gating functions, where each mixture component is itself a mixture. A Bayesian treatment of HME, which is limited to binary splits on each mixture level and is employing variational inference can be found in [47].

3.4 Dirichlet Process Mixtures of Generalized Linear Models

The regression model that we in the main part of this thesis is called *Dirichlet Process Mixtures of Generalized Linear Models* (DP-GLM) [5]. It is a Bayesian regression method, where multiple localized linear regression models are composing a global model of the joint distribution. The number of regression models is learned from data by using the Dirichlet process as a prior. As the number of clusters is modeled as unbounded and does not need to be specified in advance,

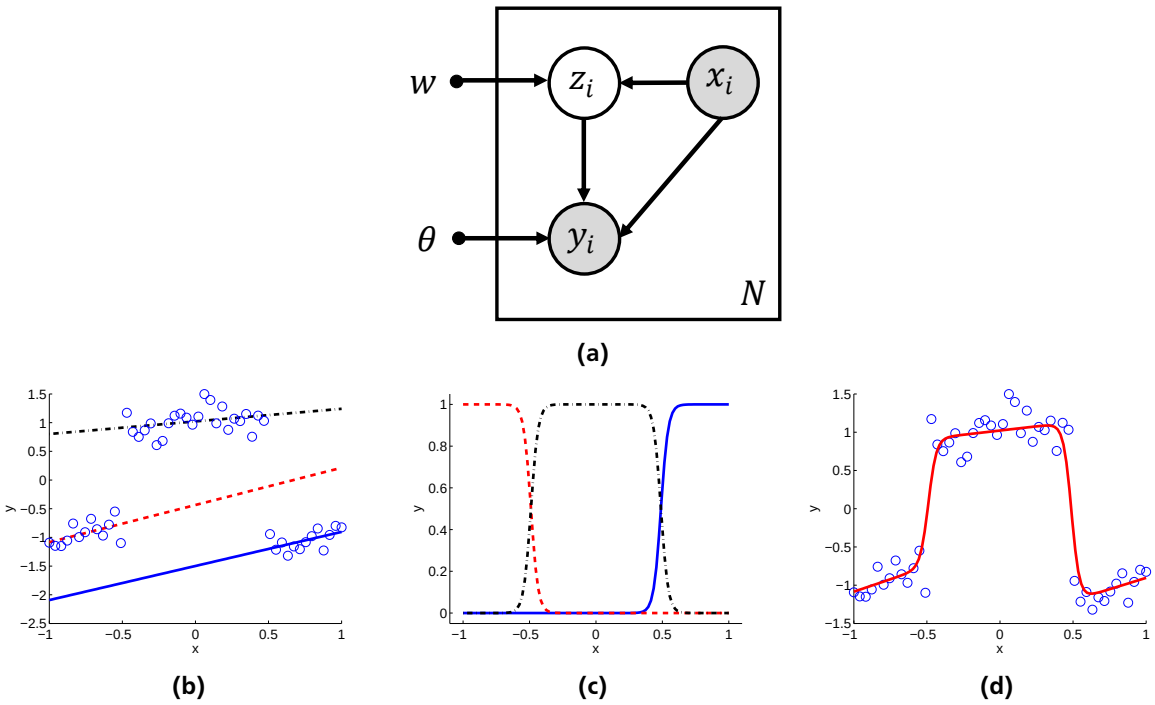


Figure 3.1.: (a) Graphical model for the Mixture of experts model. (b) Three separate experts fitted to some data. (c) Input dependent gating functions for the three experts. (d) Prediction of the Mixture of Experts model as conditionally weighted average of the expert predictions. Figure (a) based on Figure 4 of [45] and on Figure 2 of [48]. Figures (b) - (d) created with the `mixexpDemo` Skript of the Pmtk3 probabilistic modeling toolkit for Matlab of Kevin Murphy [17].

DP-GLM can be categorized as a *nonparametric infinite mixture model*.

We first want to motivate DP-GLM in comparison to a very general regression framework that is called *generalized linear models* (GLM) [49]. We deal with the general regression model, that can be written as: [5]

$$\boldsymbol{y}|\boldsymbol{x} \sim f(m(\boldsymbol{x}))$$

Generalized linear models are a flexible class of models, where the output density of the targets $\boldsymbol{y}$ is in the exponential family and the mean function $m(\boldsymbol{x})$ is a linear combination of the inputs $\boldsymbol{x}$. Wrapped around the mean function, there is a link function f, which in general can be non-linear. In linear regression, the link function is a Gaussian distribution. The GLM formulation entails many methods such as linear regression, logistic regression, and Poisson regression. [8]
The GLM framework makes two assumptions, that limit the flexibility of the regression approach. Firstly, the inputs enter the distribution of the targets through a linear function. A non-linear link function can be applied to the output of the mean function, but it can not depend on the inputs. Secondly, the variance of the targets is not allowed to be input-dependent. DP-GLM can relax both of these assumptions.

The idea of DP-GLM is, to model the mean function $m(\boldsymbol{x})$ by a mixture of simpler response distributions $f_i(m_i(\boldsymbol{x}))$. Each of these simpler distributions is just applicable in a region of the inputs that have similar response patterns. In this thesis, each of the mixture components and thus each $m_i(\boldsymbol{x})$ is a linear regression model. A non-linear mean function $m(\boldsymbol{x})$ emerges if the uncertainty, which local response is active in which regions of the input, is marginalized out. As the variance of the distribution of the targets can vary across the different linear models, we can capture the variance as a function of the inputs. This feature enables the DP-GLM to deal with *heteroscedastic* data. Being able to model input-dependent variance is a strong advantage of DP-GLM in comparison to models like standard Gaussian processes [50] that assume a constant noise across the inputs. Additionally, GPs are mostly limited to smooth functions, as a kernel is used for modeling the covariance matrix. Due to the use of locally linear models, DP-GLM is more flexible in modeling arbitrary response function.
The infinite mixture modeling approach (see chapter 2.5) with a Dirichlet process prior is used to infer both the number and the shape of the locally linear models. This is important to obtain an adaptable regression method, which can construct very simple, but also very complex target distributions, depending on the data. The complexity of the target

distribution arises through the combination of many locally linear models. In this thesis, we also use the Dirichlet distribution prior as a replacement of the Dirichlet process and compare the resulting finite mixture model to the infinite mixture case. [5]

In contrast to other methods that use locally linear models, like LWR, LWPR and MOE, DP-GLM is a Bayesian regression method. It allows us to calculate a posterior predictive distribution for the outputs. We thus have a quantification of the uncertainty of predictions by marginalizing out the parameters. This uncertainty of predictions is naturally included in the model formulation. When employing other state of the art function approximators, like neural networks, extra efforts have to be taken to learn a posterior over all model parameters [51]. This is still an active area of research.

We now state the generative model of DP-GLM using multivariate Gaussians, which is a modeling approach that dates back to [52, 53, 54]. This extends Dirichlet process mixtures (chapter 2.5) for density estimation to the regression case. The generative model describes, how a new input-output pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ can be drawn from the model:

$$\begin{aligned} G &\sim \mathrm{DP}(\alpha, H) \\ \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\beta}_i, \boldsymbol{V}_i &\sim G \\ \boldsymbol{x}_i &\sim \mathrm{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \\ \boldsymbol{y}_i | \boldsymbol{x}_i &\sim \mathrm{N}(\boldsymbol{\beta}_i \boldsymbol{X}_i, \boldsymbol{V}_i) \end{aligned}$$

$\boldsymbol{x}_i$ is an input vector of length m and $\boldsymbol{y}_i$ is a target vector of length d. We model the inputs $\boldsymbol{x}_i$ with a Gaussian distribution and the outputs $\boldsymbol{y}_i$ with a Gaussian that is dependent on the inputs. The output distribution for drawing a target $\boldsymbol{y}_i$ thus has the form of a Bayesian linear regression model (see chapter 2.2).
The steps of the data generating process are as follows. G is a discrete probability measure that is drawn from a Dirichlet process with concentration parameter α and base measure H. The parameters of the input and the output distributions are drawn from G. A new input $\boldsymbol{x}_i$ is drawn from a Gaussian conditioned on the mean $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$. The input enters the regressor matrix $\boldsymbol{X}_i$ of the linear model, which has the regression parameters $\boldsymbol{\beta}_i$ and a covariance matrix $\boldsymbol{V}_i$. The new target variable $\boldsymbol{y}_i$ is drawn from the linear model.
The regression coefficients $\boldsymbol{\beta}_i$ and the regressor matrix $\boldsymbol{X}_i$ have the following shape:

$$\boldsymbol{X}_i = [\, 1 \quad \boldsymbol{x}_i \,]^T \quad \text{and} \quad \boldsymbol{\beta}_i = [\, \boldsymbol{\beta_0} \quad \boldsymbol{\beta}_1 \,]$$

$\boldsymbol{\beta}_1$ is a d by m matrix and represents the slope parameters. $\boldsymbol{\beta}_0$ encodes the intercept and has a length of d.
As we know from infinite mixture models, in this generative model the model parameters are drawn individually for every newly generated data pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$. The clustering effect arises due to the *rich get richer* property of the Dirichlet process. Previously seen parameter values are more likely to be drawn again, so that in some regions of the data the input-output pairs will share the same cluster parameters. The generative model is represented as a graphical model in Figure 3.2.

The DP prior takes the role of clustering the input-output pairs. It acts as a kernel for the inputs. It measures the distance between two data points by the probability that the hidden parameters, and thus the cluster is shared [5]. In training, when both $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are observed, the posterior distribution of the generative model will cluster data points according to nearby inputs that exhibit the same relationship to their input. When testing on new inputs, the posterior predictive distribution can be understood as clustering the test data based on the training data and then predicting the output according to the linear model that is associated with the test input's cluster. We will give further details on the predictive distribution in the following chapter.

The derivation of the predictive distribution differs, depending on which inference method is used for learning the parameters of DP-GLM. The posterior of the DP-GLM is intractable, making it necessary to use Bayesian inference methods. This is due to the complexity of the integration when marginalizing out the uncertainty of the parameters. To nonetheless obtain an approximation of the posterior, we derive and evaluate the Bayesian inference methods of Gibbs sampling and variational inference for DP-GLM. [5]
Examples of data fitted with DP-GLM can be seen in chapter 6.

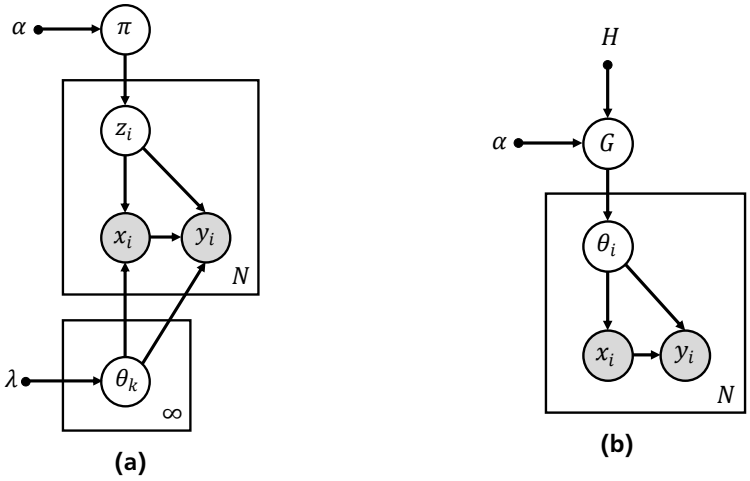


Figure 3.2.: Two different graphical models for DP-GLM. Assuming Gaussian clusters and linear models, the parameters $\boldsymbol{\theta}$ are mean and covariance matrix of the clusters, as well as regression coefficients and covariance matrix of the linear models. (a) The first graphical model displays the stick-breaking representation. The component parameters $\boldsymbol{\theta}_k$ are drawn from a prior H with hyperparameters λ, separately for every component k. The latent variables z_i assign N input-output pairs (x_i, y_i) to a component and are drawn from a Categorical distribution. The Categorical distribution is parametrized by the mixture weights π. The mixture weights π are drawn from a stick-breaking prior with parameter α. As a result of this procedure, the number of components is unbounded in the model formulation. (b) In the Pólya Urn representation the parameters $\boldsymbol{\theta}_i$ are drawn from the infinite, discrete distribution G. G is drawn from a Dirichlet process $\mathrm{DP}(\alpha, H)$ with concentration parameter α and base measure H. Two input-output pairs belong to the same cluster, if their parameters $\boldsymbol{\theta}_i$ coincide. Based on figure 25.2 in [8] and figure 2.9 in [22] and extended to DP-GLM.

4 Variational Inference

In section 3.4 on DP-GLM we have outlined that the posterior of the regression method is intractable. Due to this reason, we need to resort to approximate inference methods. Our method of choice is mean field variational inference (VI).
In this chapter we first elaborate the basic ideas of variational inference [8, 7] and then go on deriving mean field variational inference algorithms. We collect the existing material of VI for the finite mixture of Gaussians [7, 8, 55] and Dirichlet process Gaussian mixture models [56]. We add more detailed derivations for these two models. We then derive a mean field VI algorithm for Dirichlet process mixtures of generalized linear models. In the end we use the obtained approximate posterior distribution to derive a posterior predictive distribution for making predictions with DP-GLM.

4.1 Basics of Variational Inference

Variational inference [57, 58, 6, 59, 60, 61] is a deterministic approximation scheme for Bayesian models. The variational method is in contrast to sampling-based inference methods like the Markov chain Monte Carlo framework, which is stochastic.

In Bayesian models, there are observed data variables, like the training inputs and the training outputs of a regression model. There are also latent variables that are not observed and need to be inferred. These are for example cluster assignment variables $\boldsymbol{z}_n$ for each datum in a mixture model. We additionally have parameters, that we put priors on. The number of latent variables typically grows with the size of the data set, whereas the number of parameters in general stays constant for a bigger data set. To stay in the mixture model example, the parameters could be the component means and covariance matrices. The priors of the parameters have deterministic parameters, which we call hyperparameters.[7]
The problem we are facing is that we want to calculate a posterior distribution $p(\boldsymbol{x} \mid \boldsymbol{D})$ of the unknown random variables $\boldsymbol{x}$ given the observed variables $\boldsymbol{D}$ or want to evaluate expectations with respect to that posterior. In Bayesian models of practical interest, the posterior is often too complex to be evaluated analytically. This is due to the involved integrations over the space of latent variables that is potentially large and high-dimensional. he integrations may not have closed-form analytical solutions or the exact calculation may be computationally too expensive. These intractable integrations arise in the normalization constant of the posterior, whereas evaluating the unnormalized posterior usually is possible. [7, 8]

The main idea of variational inference is to approximate the true posterior $p(\boldsymbol{x} \mid D)$ by a $q(\boldsymbol{x})$ that is from a tractable family of distributions. One can, for example, choose the form of a multivariate Gaussian or assume that $q(\boldsymbol{x})$ has a factored structure. The parameters of the approximate posterior $q(\boldsymbol{x})$, which are called variational parameters, are then chosen in a way that the approximate posterior gets as close to the true posterior as possible. The similarity of the approximate posterior and the true posterior hereby often is measured by using the Kullback-Leibler-divergence (KL) as a cost function. As we will see standard KL-divergence is still not tractable so that the cost function is rewritten as a lower bound of the log-likelihood of the data. This bound is called evidence lower bound (ELBO) or variational lower bound (VLB). By maximizing this cost function, the inference problem is turned into an optimization problem. The outcome is a variational Bayes algorithm with an EM-like structure for the variational parameters of the approximate posterior. It iteratively updates responsibilities and then calculates a new set of the remaining variational parameters using the responsibilities to weight the data.

4.1.1 Cost Functions of Variational Inference

In this section we want to reason, why the variational lower bound is commonly used as a a cost function for VI. The KL-divergence from the true posterior to the approximate posterior is defined as:

$$\mathrm{KL}(p \parallel q) = \sum_{\boldsymbol{x}} p(\boldsymbol{x} \mid D) \log \frac{p(\boldsymbol{x} \mid D)}{q(\boldsymbol{x})}$$

We use the sum for discrete random variables, but the definition is equivalent for continuous random variables, if the sum is replaced by an integral. This definition is called the forwards KL-divergence or *moment projection* (M-projection).

Inspecting the equation we see that the KL gets infinite if $p(\boldsymbol{x} \mid D) = 0$ or $q(\boldsymbol{x}) = 0$. If the forward KL is minimized as a cost function, the parameters of the variational distribution are chosen to avoid the approximate distribution becoming zero. Consequently the variance of $p(\boldsymbol{x})$ is overestimated. If $p(\boldsymbol{x})$ is a multimodal distribution and $q(\boldsymbol{x})$ is restricted to a unimodal family, the resulting mode of the approximate posterior will be in a region of low density of the true posterior. This is not a very accurate approximation. Furthermore the forward KL is hard to evaluate, as we need to take expectations with respect to $p(\boldsymbol{x} \mid D)$. This operation is assumed to be intractable.

An obvious alternative to the forwards KL is the reverse KL-divergence:

$$\mathrm{KL}(q \parallel p) = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x} \mid D)}$$

This definition is also called *information projection* (I-projection). In the context of VI the advantage is that we can choose the family of $q(\boldsymbol{x})$ such that expectations with regard to this distribution are tractable. We observe that the reverse KL is infinite for $p(\boldsymbol{x}) = 0$ and $q(\boldsymbol{x}) > 0$. This means that the KL divergence forces the approximate distribution to be zero, wherever $p(\boldsymbol{x})$ is zero. The result is that the I-projection, assuming a unimodal $q(\boldsymbol{x})$, locks onto one mode of the true posterior. The variance of the true posterior is underestimated. In such a setting the I-projection is a a better choice than the M-projection because we avoid to get an approximate posterior with its' mode in an area of low density of the true posterior. [8, 7]

The remaining problem of this cost function is that the true posterior still appears in the equation. Usually the intractable integrations arise in the normalization constant $p(D)$ of the posterior, since it has to be integrated over the space of all hidden variables. The unnormalized posterior $\tilde{p}(\boldsymbol{x} \mid D) = p(\boldsymbol{x} \mid D)\, p(D)$ usually can be calculated. We define a new objective:

$$\begin{aligned}
J(q) &= \mathrm{KL}(q \parallel \tilde{p}) \\
&= \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x} \mid D)} \\
&= \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x} \mid D)\, p(D)} \\
&= \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x} \mid D)} - \log p(D) \\
&= \mathrm{KL}(q \parallel p) - \log p(D)
\end{aligned}$$

Since $\log p(D)$ is a constant, minimizing $J(q)$ results in minimizing $\mathrm{KL}(q \parallel p)$ and thus enforces the approximate distribution to become close to the true one. As the KL divergence never gets negative, $J(q)$ is an upper bound on the negative log-likelihood:

$$J(q) = \mathrm{KL}(q \parallel p) - \log p(D) \geq -\log p(D)$$

Instead of minimizing this quantity, we can equivalently turn the problem into an maximization problem by reversing the sign:

$$\begin{aligned}
L(q) = -J(q) &= -\mathrm{KL}(q \parallel p) + \log p(D) \leq \log p(D) \\
&= -\sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x} \mid D)} \\
&= \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{\tilde{p}(\boldsymbol{x} \mid D)}{q(\boldsymbol{x})}
\end{aligned}$$

By maximizing $L(q)$ we maximize a lower bound on the log-likelihood of the data. $L(q)$ is thus called evidence lower bound or variational lower bound. $L(q)$ is the final cost function that will be used throughout this chapter.
In principle it is possible to use other measures as a cost function, for example the *alpha divergence* or the *Hellinger distance*. [8]

4.1.2 The Mean Field Method

The mean field method [62] is one of the most common approximations to make the posterior tractable. It is assumed that the posterior is a fully factorized distribution of the form:

$$q(\boldsymbol{x}) = \prod_{i=1}^{D} q_i(\boldsymbol{x}_k)$$

No other assumptions on the functional form of the individual factors q_i have to be made. The functional form will be an automatic result of the optimization. Other approximation schemes assume that the approximate posterior belongs to a certain family of tractable distributions, for example Gaussians. [7]

The optimization problem is to maximize the variational lower bound:

$$\max_{q_1,\dots,q_K} L(q)$$

We optimize over the variational parameters of each marginal distribution q_i. In the following we show the steps to get a general update equation in the case of the mean field assumption, which can be used to derive an EM-like coordinate ascent algorithm for the variational parameters of the approximate posterior.

We can do the optimization for one term q_j at a time. We factor out the term that contain q_j and regard the other terms as constant. Following [8] we get:

$$\begin{aligned}
L(q_j) &= \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{\tilde{p}(\boldsymbol{x} \mid D)}{q(\boldsymbol{x})} \\
&= \sum_{\boldsymbol{x}} \prod_i q_i(\boldsymbol{x}_i) \left[\log \tilde{p}(\boldsymbol{x} \mid D) - \sum_k \log q_k(\boldsymbol{x}_k) \right] \\
&= \sum_{\boldsymbol{x}_j} \sum_{\boldsymbol{x}_{-j}} q_j(\boldsymbol{x}_j) \prod_{i \neq j} q_i(\boldsymbol{x}_i) \left[\log \tilde{p}(\boldsymbol{x} \mid D) - \sum_k \log q_k(\boldsymbol{x}_k) \right] \\
&= \sum_{\boldsymbol{x}_j} q_j(\boldsymbol{x}_j) \sum_{\boldsymbol{x}_{-j}} \prod_{i \neq j} q_i(\boldsymbol{x}_i) \log \tilde{p}(\boldsymbol{x} \mid D) - \sum_{\boldsymbol{x}_j} q_j(\boldsymbol{x}_j) \sum_{\boldsymbol{x}_{-j}} \prod_{i \neq j} q_i(\boldsymbol{x}_i) \left[\sum_{k \neq j} \log q_k(\boldsymbol{x}_k) + q_j(\boldsymbol{x}_j) \right] \\
&= \sum_{\boldsymbol{x}_j} q_j(\boldsymbol{x}_j) \log f_j(\boldsymbol{x}_j) - \sum_{\boldsymbol{x}_j} q_j(\boldsymbol{x}_j) \log q_j(\boldsymbol{x}_j) + \text{ const}
\end{aligned} \tag{4.1}$$

Where:

$$\log f_j(\boldsymbol{x}_j) = \sum_{\boldsymbol{x}_{-j}} \prod_{i \neq j} q_i(\boldsymbol{x}_i) \log \tilde{p}(\boldsymbol{x} \mid D) = \mathbb{E}_{-q_j}[\log \tilde{p}(\boldsymbol{x} \mid D)]$$

The expression $\mathbb{E}_{-q_j}[f(\boldsymbol{x})]$ denotes an expectation of $f(\boldsymbol{x})$ with respect to the q distributions over all variables z_i except for z_j. An example of this for three variables is:

$$\mathbb{E}_{-q_2}[f(\boldsymbol{x})] = \sum_{\boldsymbol{x}_1} \sum_{\boldsymbol{x}_3} q(\boldsymbol{x}_1) q(\boldsymbol{x}_3) f(\boldsymbol{z_1}, \boldsymbol{z_2}, \boldsymbol{z_3})$$

We see that the expectation is only taken with respect to q_1 and q_3. We now keep all $\{q_{i \neq j}\}$ constant and maximize the VLB $L(q_j)$ with respect to the distribution q_j. Inspecting equation 4.1 we see that this is a KL divergence between q_j and f_j. We can write:

$$L(q_j) = -\mathrm{KL}(q_j \parallel f_j) + \text{const}$$

L can be maximized by minimizing $\mathrm{KL}(q_j \parallel f_j)$. This is achieved by setting $q_j = f_j$:

$$q_j(\boldsymbol{x}_j) = \frac{1}{Z_j} \exp\left(\mathbb{E}_{-q_j}[\log \tilde{p}(\boldsymbol{x} \mid D)] \right)$$

The normalization constant Z_j can be ignored, as we know that q_j has to be a normalized probability distribution. The normalization can be done after finding the functional form of q_j.

The mean field update formula to obtain the variational parameters of the factorized approximate posterior distribution thus is:

$$\log q_j\left(\boldsymbol{x}_j\right) = \mathbb{E}_{-q_j}[\log \tilde{p}(\boldsymbol{x} \mid D)] + \text{const} \tag{4.2}$$

The distribution $\tilde{p}(\boldsymbol{x} \mid D)$ can also be written as $p\left(\boldsymbol{x} \mid D\right) p(D)$ or $p(\boldsymbol{x}, D)$. In the following derivations we call this distribution the joint distribution of the hidden variables and the data or unnormalized posterior. It is our *target* distribution, because we need it to plug it into the mean field update equation (4.2) to find the approximate posterior q. [8, 7]

4.1.3 Variational Bayes Expectation Maximization

Models with the two categories of unknowns of parameters $\boldsymbol{\theta}$ and latent variabels $\boldsymbol{z}$ are often fitted with the EM algorithm. In the E-step a posterior $p\left(\boldsymbol{z}_i|\boldsymbol{x}_n, \boldsymbol{\theta}\right)$ over the latent variables is inferred and in the M-step a point estimate of the parameters $\boldsymbol{\theta}$ is calculated.
If we use mean field variational inference to only infer the latent variables (i.e. cluster assignment variables $\boldsymbol{z}$), the method is called *Variational Bayes* (VB). *Variational Bayes EM* (VBEM) is an alternative, where the uncertainty in the parameters is modelled in addition to the uncertainty in the latent variables. The computational cost stays roughly the same as for EM. The mean field factorization then is of the form:

$$p\left(\boldsymbol{\theta}, \boldsymbol{z}_{1:N}|\mathrm{D}\right) \approx q(\boldsymbol{\theta}) q(\mathbf{z}) = q(\boldsymbol{\theta}) \prod_{n=1}^{N} q\left(\boldsymbol{z}_i\right).$$

The first factorization is crucial for obtaining a tractable algorithm and represents the mean field assumption. The second factorization is the general model assumption that the latent variables are i.i.d. conditional on the parameters.
In an VBEM algorithm we alternate between updating $q\left(\boldsymbol{z}_i \mid \mathrm{D}\right)$ (Variational E-step) and updating $q(\boldsymbol{\theta} \mid \mathrm{D})$ (Variational M-step). Standard EM is recovered, when a delta function is used for approximating the posterior of the parameters $q(\boldsymbol{\theta} \mid \mathrm{D}) \approx \delta_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\theta})$. In comparison to standard EM the steps are:

- Variational E-step: Update $q\left(\boldsymbol{z}_i \mid \mathrm{D}\right)$. Instead of plugging in a MAP estimate of $\boldsymbol{\theta}$, we have to average over all possible parameter values.
- Variational M-step: Update $q(\boldsymbol{\theta} \mid \mathrm{D})$. Except of computing a point estimate of the parameters, we have to update the hyperparameters (variational parameters) of the approximate posterior of the parameters $\boldsymbol{\theta}$.

One advantage of VBEM over EM is that we can compute the variational lower bound, which is a lower bound on the marginal likelihood. This can be used for model selection or checking the convergence of the algorithm. Furthermore in VBEM there is no artificial distinction between parameters and latent variables, as both are treated as unknowns. By inferring a posterior over the parameters, we get a quantification of uncertainty of predictions, for example in a regression setting. Similar to EM, a weakness of VBEM is that it can get stuck in bad local optima. To lower this risk, the algorithm should be started multiple times with different initializations. In chapter 5 we introduce Gibbs sampling as a method for initializing VBEM. [8]
In the following we show the steps for deriving VBEM algorithms for GMM, DP-GMM and DP-GLM in detail.

4.2 Variational Bayes EM for Gaussian Mixture Models

In this section, we show the steps for deriving a variational Bayes EM algorithm for Gaussian mixture models. [55, 8, 7] We first define the likelihood and the priors and state the mean field factorization to obtain a variational posterior. After that, we show the steps for deriving the variational E-step and the variational M-step. In the end, we show the form of the variational lower bound.

4.2.1 Choice of Likelihood and Priors

The likelihood function of a multivariate Gaussian mixture model is:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Sigma}) &= \prod_{n=1}^{N}\prod_{k=1}^{K}\mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} \\ &= \prod_{n=1}^{N}\prod_{k=1}^{K}\left[\frac{1}{(2\pi)^{D/2}}|\boldsymbol{\Sigma}_k|^{1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)\right\}\right]^{z_{nk}} \end{aligned} \tag{4.3}$$

For each of $n = 1 \dots N$ observations we have a corresponding latent variable $\boldsymbol{z}_n$, which is a 1-of-K binary vector with elements z_{nk}. The latent variable takes the value $z_{nk} = 1$, if the data point with index n belongs to cluster k and $z_{nk} = 0$ otherwise. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and the precision matrix of a cluster. D is the length of a data vector $\boldsymbol{x}$. K is the upper limit of clusters.

In the further thesis we do not always write out the involved probability distributions. These can be looked up in Appendix C. The normalization constants of the distributions can also be looked up there.
We assume the following factored conjugate prior:

$$\begin{aligned} p(\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) &= p(\boldsymbol{z}\mid\boldsymbol{\pi})\,p(\boldsymbol{\pi})\,p(\boldsymbol{\mu}\mid\boldsymbol{\Sigma})\,p(\boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N}\mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\,\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0,(\lambda_0\boldsymbol{\Sigma}_k)^{-1}\right)\mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0,\nu_0) \\ &= \left[\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}\right]\left[C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1}\right] \\ &\quad \left[\prod_{n=1}^{K}\frac{|\lambda_0\boldsymbol{\Sigma}_k|^{1/2}}{(2\pi)^{D/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{\mu}_k-\boldsymbol{m}_0)^{\mathrm{T}}\lambda_0\boldsymbol{\Sigma}_k(\boldsymbol{\mu}_k-\boldsymbol{m}_0)\right\}B(\boldsymbol{L}_0,\nu_0)|\boldsymbol{\Sigma}_k|^{(\nu_0-D-1)/2}\exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right)\right\}\right] \end{aligned} \tag{4.4}$$

We put a Normal-Wishart prior on the cluster means and cluster precision matrices. This distribution is a composite prior consisting of of a normal distribution and a Wishart distribution. It is used, if both the means and precisions are unknown. The Normal-Wishart prior has the hyperparameters $\boldsymbol{m}_0, \lambda_0, \boldsymbol{L}_0$ and ν_0.

The mixing weights π have a symmetric Dirichlet prior with hyperparameters $\boldsymbol{\alpha_0} = \alpha_0\boldsymbol{I}$. α_0 has a decisive effect on the effective number of clusters that will be used for representing the data. Depending on the choice of α_0, for some of the K clusters, the expected values for the mixing weights will be indistinguishable from their prior values. A smaller α_0 favors solutions, where some of the mixing coefficients are close to zero. We will evaluate the hyperparameter α_0 in detail in chapter 7. [7]

The latent variables $\boldsymbol{z}_n$ are drawn from a Categorical distribution conditioned on the mixing weights. The unknown random variables in summary are $\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The subscript 0 indicates hyperparameters of the prior.

4.2.2 Mean Field Factorization

The exact, but intractable posterior $p(\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\mid\boldsymbol{x})$ is a mixture of K^N distributions corresponding to all possible labelings $\boldsymbol{z}$. The mean field approximation of the posterior concentrates the density around one of these modes: [8]

$$p(\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\mid\boldsymbol{x}) \approx q(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})\,q(\boldsymbol{z}) = q(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})\prod_{n=1}^{N}q(\boldsymbol{z}_i)$$

The first factorization is the mean field approximation and enables a tractable algorithm. The second factorization represents that the latent variables $\boldsymbol{z}$ are i.i.d. conditional on the parameters. Using this factorization, we will derive an approximate posterior of the following form in the next sections:

$$\begin{aligned} q^\star(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= q^\star(\boldsymbol{z})\, q^\star(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \left[\prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{r}_n)\right]\left[\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\lambda_k \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_k, \nu_k)\right] \end{aligned}$$

The superscript $\star$ denotes that the respective probability distribution is the optimal solution of the optimization problem. The subscript k indicates the updated variational parameters of the approximate posterior. We do not need to specify this variational posterior in advance. It will be a result of the free form optimization of the variational distributions using the mean field update equation (4.2).

4.2.3 Target Distribution

As we know from the mean field update equation (4.2), we need the unnormalized log posterior for deriving the form of the factorized distributions of the approximate posterior. We thus call this distribution the target distribution of the optimization procedure. It is as follows:

$$\begin{aligned} \log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(\boldsymbol{z}|\boldsymbol{\pi}) + \log p(\boldsymbol{\pi}) + \log p(\boldsymbol{\mu}|\boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) \\ &= \log \prod_{n=1}^{N} \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} + \log \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi}) + \log \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) \\ &\quad + \log \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\lambda_0 \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0, \nu_0) \end{aligned}$$

We see that the log joint over data and hidden variables is a sum over the log of the data likelihood and the log prior. In the following we use this distribution for deriving the variational E-step and the variational M-step of the VBEM algorithm.

4.2.4 Variational E-step: Updating $q(\boldsymbol{z})$

The form of $q^\star(\boldsymbol{z})$ can be derived by taking the expectation of the unnormalized log posterior with respect to all hidden variables except of $\boldsymbol{z}$. All terms, that do not involve $\boldsymbol{z}$ are constant when updating $q(\boldsymbol{z})$ and can be ignored.

$$\begin{aligned} \log q^\star(\boldsymbol{z}) &= \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}[\log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \text{ const} \\ &= \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[\log \prod_{n=1}^{N} \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}}\right] + \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[\log \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\right] + \text{ const} \\ &\quad + \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[\log \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}}\right] + \text{ const} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log \rho_{nk} + \text{ const} \end{aligned} \tag{4.5}$$

Where we have defined:

$$\log \rho_{nk} = \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}[\log \pi_k] + \frac{1}{2} \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}[\log |\boldsymbol{\Sigma}_k|] - \frac{D}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right]$$

In this term three expectations have to be calculated. Using equation C.16 we have for the first expectation:

$$\log \tilde{\pi}_k = \mathbb{E}[\log \pi_k] = \psi(\alpha_k) - \psi\left(\sum_{k'} \alpha_{k'}\right) \qquad \text{with} \quad \psi(a) = \frac{d}{da} \log \Gamma(a)$$

$\psi(a)$ is the Digamma function.
Using standard expectations of a Wishart distribution (equation C.40) we get for the second expectation:

$$\log \tilde{\Sigma}_k = \mathbb{E}\left[\log|\Sigma_k|\right] = \sum_{j=1}^{D} \psi\left(\frac{\nu_k + 1 - j}{2}\right) + D\log 2 + \log|\Sigma_k|$$

The expectation of the quadratic term can be derived by using the definition of the expectation:

$$\begin{aligned}
\mathbb{E}_{\mu_k,\Sigma_k}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] &= \iint (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\, q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\, d\boldsymbol{\mu}_k d\boldsymbol{\Sigma}_k \\
&= \int \left\{ \int (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\, q^\star(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)\, d\boldsymbol{\mu}_k \right\} q^\star(\boldsymbol{\Sigma}_k)\, d\boldsymbol{\Sigma}_k \\
&= \int \mathbb{E}_{\mu_k}\left[(\boldsymbol{\mu}_k - \boldsymbol{x}_n)^T \boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{x}_n)\right] \cdot q^\star(\boldsymbol{\Sigma}_k)\, d\boldsymbol{\Sigma}_k
\end{aligned}$$

The inner expectation thus is with respect to $\boldsymbol{\mu}_k$ of a Gaussian distribution. We use Equation 380 in [63] for $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{m}, \boldsymbol{V})$ to get:

$$\mathbb{E}_{\mu_k}\left[(\boldsymbol{\mu}_k - \boldsymbol{x}_n)^T \boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{x}_n)\right] = (\boldsymbol{m}_k - \boldsymbol{x}_n)^T \boldsymbol{\Sigma}_k (\boldsymbol{m}_k - \boldsymbol{x}_n) + \mathrm{Tr}\left[\boldsymbol{\Sigma}_k \cdot (\lambda_k \boldsymbol{\Sigma}_k)^{-1}\right]$$

Substituting back yields the following expression for the expectation of the quadratic term:

$$\begin{aligned}
\mathbb{E}_{\mu_k,\Sigma_k}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] &= \int \left[(\boldsymbol{m}_k - \boldsymbol{x}_n)^T \boldsymbol{\Sigma}_k (\boldsymbol{m}_k - \boldsymbol{x}_n) + D\lambda_k^{-1}\right] \cdot q^\star(\boldsymbol{\Sigma}_k)\, d\boldsymbol{\Sigma}_k \\
&= D\lambda_k^{-1} + \mathbb{E}_{\Sigma_k}\left[(\boldsymbol{m}_k - \boldsymbol{x}_n)^T \boldsymbol{\Sigma}_k (\boldsymbol{m}_k - \boldsymbol{x}_n)\right] \\
&= D\lambda_k^{-1} + \mathbb{E}_{\Sigma_k}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k \cdot (\boldsymbol{m}_k - \boldsymbol{x}_n)(\boldsymbol{m}_k - \boldsymbol{x}_n)^T\right]\right\} \\
&= D\lambda_k^{-1} + \mathrm{Tr}\left\{\mathbb{E}_{\Sigma_k}\left[\boldsymbol{\Sigma}_k\right] \cdot (\boldsymbol{m}_k - \boldsymbol{x}_n)(\boldsymbol{m}_k - \boldsymbol{x}_n)^T\right\} \\
&= D\lambda_k^{-1} + \mathrm{Tr}\left\{\nu_k \boldsymbol{L}_k \cdot (\boldsymbol{m}_k - \boldsymbol{x}_n)(\boldsymbol{m}_k - \boldsymbol{x}_n)^T\right\} \\
&= D\lambda_k^{-1} + \nu_k (\boldsymbol{m}_k - \boldsymbol{x}_n)^T \boldsymbol{L}_k (\boldsymbol{m}_k - \boldsymbol{x}_n)
\end{aligned}$$

We here used the knowledge, that $q^\star(\boldsymbol{\Sigma}_k)$ is a Wishart distribution with expectation $\mathbb{E}_{\Sigma_k}(\boldsymbol{V}_k) = \nu_k \boldsymbol{L}_k$ (see equation C.39). This is unknown at this step, but will be confirmed in the derivation for the variational M-step.

We have gathered all necessary expectations and can derive the functional form of for $q^\star(\boldsymbol{z})$. By taking the exponential on both sides of Equation 4.5 we obtain a Categorical distribution:

$$q^\star(\boldsymbol{z}) \propto \prod_{n=1}^{N} \prod_{k=1}^{K} \rho_{nk}^{z_{nk}}$$

This is an unnormalized distribution and we still need to find the normalization constant. For each value of n the quantities z_{nk} are binary and sum to one for all values of k. As a result of this the basis of the exponent z_{nk} has to sum to one for all values of k. We call these values the responsibilities $r_{nk} = (\rho_{nk} / \sum_j \rho_{nj})$. Therefore a normalized distribution is given by:

$$q^\star(\boldsymbol{z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} r_{nk}^{z_{nk}} \qquad \text{with} \quad r_{nk} \propto \tilde{\pi}_k \tilde{\Sigma}_k^{\frac{1}{2}} \exp\left(-\frac{D}{2\lambda_k} - \frac{\nu_k}{2}(\boldsymbol{x}_n - \boldsymbol{m}_k)^T \boldsymbol{L}_k (\boldsymbol{x}_n - \boldsymbol{m}_k)\right) \tag{4.6}$$

$$r_{nk} = \frac{\rho_{nk}}{\sum_j \rho_{nj}}$$

The variational E-step is complete and we go on with the variational M-step.

4.2.5 Variational M-step: Updating $q(\pi, \mu, \Sigma)$

Starting again from the mean field update equation (4.2) we can derive $q(\pi, \mu, \Sigma)$. In the resulting equations we will hold the responsibilities constant, which we known from the variational E-step. We have to take the expectation of the unnormalized log posterior with respect to $\boldsymbol{z}$. All terms, that do not involve π, μ, or Σ are constant when updating these variables and can be ignored. The definitions of the involved densities can be found in appendix C. We get for $q(\pi, \mu, \Sigma)$:

$$\begin{aligned}
\log q^\star(\pi, \mu, \Sigma) &= \mathbb{E}_{q(\boldsymbol{z})}[\log p(\boldsymbol{x}, \boldsymbol{z}, \pi, \mu, \Sigma)] + \text{ const} \\
&= \mathbb{E}_{q(\boldsymbol{z})}\left[\log \prod_{n=1}^{N} \text{Cat}(\boldsymbol{z}_n|\pi)\right] + \mathbb{E}_{q(\boldsymbol{z})}\left[\log \text{Dir}(\pi|\boldsymbol{\alpha}_0)\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log \left\{\prod_{n=1}^{N}\prod_{k=1}^{K} \text{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}}\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log \left\{\prod_{k=1}^{K} \text{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\lambda_0\boldsymbol{\Sigma}_k)^{-1}\right) \text{Wi}\left(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0, \nu_0\right)\right\}\right] + \text{const} \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{z})}[z_{nk}] \log \pi_k + (\alpha_0 - 1)\sum_{k=1}^{K} \log \pi_k \\
&\quad + \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{z})}[z_{nk}]\left(\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k (\boldsymbol{x}_n - \mu_k)\right) \\
&\quad + \sum_{k=1}^{K} \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_0)^T \lambda_0 \boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{m}_0) + \frac{\nu_0 - D - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right) + \text{const}
\end{aligned}$$

We can see that this expression factorizes into terms, which only depend on π and into terms, which only depend on $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. We thus can rewrite:

$$\begin{aligned}
\log q^\star(\pi, \mu, \Sigma) &= \log q^\star(\pi) + \sum_{k=1}^{K} \log q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
q^\star(\pi, \mu, \Sigma) &= q^\star(\pi)\prod_{k=1}^{K} q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)
\end{aligned}$$

Using the fact that $\mathbb{E}_{q(\boldsymbol{z})}[z_{nk}] = r_{nk}$ for the Categorical distribution $q(\boldsymbol{z})$ (see equation C.6), we get the following result for the π term:

$$\log q^\star(\pi) = (\alpha_0 - 1)\sum_k \log \pi_k + \sum_k \sum_i r_{nk} \log \pi_k + \text{const}$$

Exponentiating on both sides, we recognize this as a Dirichlet distribution:

$$\begin{aligned}
q^\star(\pi) &\propto \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1 + \sum_i r_{nk}} \\
&= C(\boldsymbol{\alpha}_k)\prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \\
&= \text{Dir}(\pi|\boldsymbol{\alpha})
\end{aligned} \tag{4.7}$$

The variational parameters of the Dirichlet distribution are:

$$\alpha_k = \alpha_0 + N_k \quad \text{with} \quad N_k = \sum_{n=1}^{N} r_{nk} \tag{4.8}$$

For the $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ terms we have:

$$\begin{aligned}
\log q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \propto \sum_{n=1}^{N} r_{nk}&\left(\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \mu_k)\right) \\
&+ \frac{1}{2}\log|\lambda_0\Sigma_k| - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_0)^T \lambda_0 \boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{m}_0) + \frac{\nu_0 - D - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right)
\end{aligned} \tag{4.9}$$

For the further derivations we assume that the optimal $q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ can be written as a Normal-Wishart distribution:

$$q^\star(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{k=1}^{K} q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \prod_{k=1}^{K} q^\star(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)\, q^\star(\boldsymbol{\Sigma}_k) = \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\lambda_k\boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_k, \nu_k) \tag{4.10}$$
$$= \prod_{n=1}^{K} \frac{|\lambda_k\boldsymbol{\Sigma}_k|^{1/2}}{(2\pi)^{D/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^{\mathrm{T}}\lambda_k\boldsymbol{\Sigma}_k(\boldsymbol{\mu}_k - \boldsymbol{m}_k)\right\} B(\boldsymbol{L}_k, \nu_k)|\boldsymbol{\Sigma}_k|^{(\nu_k - D - 1)/2} \exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{L}_k^{-1}\boldsymbol{\Sigma}_k\right)\right\}$$

Given this functional form, we can do a comparison of coefficients of equation 4.9 and the assumed Normal-Wishart distribution in equation 4.10.

We first want to find $q^\star(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)$. We collect all the terms in equation 4.9, that contain $\boldsymbol{\mu}_k$ and then complete the square with respect to $\boldsymbol{\mu}_k$.

$$-\frac{1}{2}\sum_{n=1}^{N} r_{nk}\,(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\,\boldsymbol{\Sigma}_k\,(\boldsymbol{x}_n - \boldsymbol{\mu}_k) - \frac{1}{2}\,(\boldsymbol{\mu}_k - \boldsymbol{m}_0)^T\,\lambda_0\boldsymbol{\Sigma}_k\,(\boldsymbol{\mu}_k - \boldsymbol{m}_0) = \tag{4.11}$$
$$= -\frac{1}{2}\sum_{n=1}^{N} r_{nk}\left(\boldsymbol{x}_n^T\boldsymbol{\Sigma}_k\boldsymbol{x}_n - \boldsymbol{x}_n^T\boldsymbol{\Sigma}_k\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k\boldsymbol{x}_n + \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k\boldsymbol{\mu}_k\right) - \frac{1}{2}\left[\boldsymbol{\mu}_k^T\lambda_0\Sigma_k\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^T\lambda_0\Sigma_k\boldsymbol{m}_0 - \boldsymbol{m}_0^T\lambda_0\Sigma_k\boldsymbol{\mu}_k + \boldsymbol{m}_0^T\lambda_0\Sigma_k\boldsymbol{m}_0\right]$$

We now take a look at all terms in equation 4.11 that are quadratic in $\boldsymbol{\mu}_k$:

$$\text{(quadratic)} \;= -\frac{1}{2}\sum_{n=1}^{N} r_{nk}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T\lambda_0\boldsymbol{\Sigma}_k\boldsymbol{\mu}_k = -\frac{1}{2}\boldsymbol{\mu}_k^T\,(\lambda_0 + N_k)\,\boldsymbol{\Sigma}_k\boldsymbol{\mu}_k$$

Comparing the quadratic term with the term $-\frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^{\mathrm{T}}\lambda_k\boldsymbol{\Sigma}_k(\boldsymbol{\mu}_k - \boldsymbol{m}_k)$ in equation 4.10 we obtain:

$$\lambda_k = \lambda_0 + N_k$$

In order to determine $\boldsymbol{m}_k$ we only consider the terms in equation 4.11, which are linear in $\boldsymbol{\mu}_k^T$:

$$\text{(linear)} \;= -\frac{1}{2}\sum_{n=1}^{N} r_{nk}\left(-\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k\boldsymbol{x}_n\right) - \frac{1}{2}\left(-\boldsymbol{\mu}_k^T\lambda_0\boldsymbol{\Sigma}_k\boldsymbol{m}_0\right) = \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k\,(N_k\overline{\boldsymbol{x}}_k + \lambda_0\boldsymbol{m}_0)$$

We here used the following definition:

$$\overline{\boldsymbol{x}}_k = \frac{1}{\sum_{n=1}^{N} r_{nk}}\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n$$

Comparing the linear term with the term $-\frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^{\mathrm{T}}\lambda_k\boldsymbol{\Sigma}_k$ in equation 4.10 we obtain:

$$\boldsymbol{m}_k = \frac{1}{\lambda_k}\,(N_k\overline{\boldsymbol{x}}_k + \lambda_0\boldsymbol{m}_0)$$

We have now found the parameters of $q^\star(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k) = \mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\lambda_k\boldsymbol{\Sigma}_k^{-1})\right)$.

For obtaining the parameters $\boldsymbol{L}_k$ and ν_k of the Wishart distribution $q^\star(\boldsymbol{\Sigma}_k)$, we need the following relation:

$$\log q^\star(\boldsymbol{\Sigma}_k) = \log q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) - \log q^\star(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)$$

We additionally need the definition of the trace $\boldsymbol{a}^T\boldsymbol{a} = \mathrm{Tr}(\boldsymbol{a}\boldsymbol{a}^T)$, the cyclic property of the trace $\mathrm{Tr}(\boldsymbol{abcd}) = \mathrm{Tr}(\boldsymbol{bcda}) = \mathrm{Tr}(\boldsymbol{cdab}) = \mathrm{Tr}(\boldsymbol{dabc})$ and the definition of the Cholesky decomposition $\boldsymbol{\Sigma}_k = \boldsymbol{U}\boldsymbol{U}^T$ [63].
Combining these rules and using the substitution $\boldsymbol{y} = (\boldsymbol{\mu}_k - \boldsymbol{m}_k)$, we see that:

$$\boldsymbol{y}^T\boldsymbol{\Sigma}_k\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{y} = \mathrm{Tr}\left(\boldsymbol{U}^T\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{U}\right) = \mathrm{Tr}\left(\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{U}\boldsymbol{U}^T\right)$$
$$= \mathrm{Tr}\left(\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{\Sigma}_k\right)$$

Writing down the terms of $\log q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ in equation 4.9 that contain $\boldsymbol{\Sigma}_k$ and subtracting $\log q^\star(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)$ yields:

$$\begin{aligned}
\log q^\star(\boldsymbol{\Sigma}_k) \propto & \sum_{n=1}^{N} r_{nk}\left(\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \mu_k)\right) \\
& + \frac{1}{2}\log|\lambda_0\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_0)^T \lambda_0\boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{m}_0) \\
& + \frac{\nu_0 - D - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right) \\
& - \frac{1}{2}\log|\lambda_k\boldsymbol{\Sigma}_k| + \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T \lambda_k\boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{m}_k) \\
\propto & \frac{N_k}{2}\log|\boldsymbol{\Sigma}_k| - \frac{N_k}{2}\operatorname{Tr}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \mu_k)^T \boldsymbol{\Sigma}_k\right] \\
& + \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\operatorname{Tr}\left[\lambda_0(\boldsymbol{\mu}_k - \boldsymbol{m}_0)(\boldsymbol{\mu}_k - \boldsymbol{m}_0)^T \boldsymbol{\Sigma}_k\right] \\
& + \frac{\nu_0 - D - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right) \\
& - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \frac{1}{2}\operatorname{Tr}\left[\lambda_k(\boldsymbol{\mu}_k - \boldsymbol{m}_k)(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T \boldsymbol{\Sigma}_k\right] \\
= & \frac{\nu_0 - D - 1 + N_k}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\operatorname{Tr}[\boldsymbol{T}\boldsymbol{\Sigma}_k]
\end{aligned}$$

Here we have defined:

$$\boldsymbol{T} = \lambda_0(\boldsymbol{\mu}_k - \boldsymbol{m}_0)(\boldsymbol{\mu}_k - \boldsymbol{m}_0)^T + \boldsymbol{L}_0^{-1} + \sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T - \lambda_k(\boldsymbol{\mu}_k - \boldsymbol{m}_k)(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T \tag{4.12}$$

By matching coefficients with the term $|\boldsymbol{\Sigma}_k|^{(\nu_k - D - 1)/2}$ in the Wishart distribution in equation 4.10 we get:

$$\nu_k = \nu_0 + N_k$$

By matching coefficients with the term $\exp\left\{-\frac{1}{2}\operatorname{Tr}(\boldsymbol{L}_k^{-1}\boldsymbol{\Sigma}_k)\right\}$ in equation 4.10 we see that:

$$\boldsymbol{L}_k^{-1} = \boldsymbol{T}$$

We have now fully specified $q^\star(\boldsymbol{\Sigma}_k)$ and therefore know all update rules for obtaining the mean field approximation $q^\star(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of the exact posterior. A more compact update equation for $\boldsymbol{L}_k^{-1}$ is:

$$\boldsymbol{L}_k^{-1} = \boldsymbol{L}_0^{-1} + N_k\boldsymbol{S}_k + \frac{\lambda_0 N_k}{\lambda_k + N_k}(\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)(\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)^T$$

We use this definition:

$$\boldsymbol{S}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)^T$$

The details of deriving the simplified update equation can be found in appendix D.1.

4.2.6 Variational Lower Bound on the Marginal Likelihood

For the variational lower bound on the marginal likelihood of the variational mixture of Gaussians we get:

$$\begin{aligned}
\mathrm{L} &= \sum_{\boldsymbol{z}} \iiint q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \log\left\{\frac{p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}\right\} \mathrm{d}\boldsymbol{\pi}\,\mathrm{d}\boldsymbol{\mu}\,\mathrm{d}\boldsymbol{\Sigma} \\
&= \mathbb{E}[\log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] - \mathbb{E}[\log q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] \\
&= \mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{\pi})] + \mathbb{E}[\log p(\boldsymbol{\pi})] + \mathbb{E}[\log p(\boldsymbol{\mu}, \boldsymbol{\Sigma})] \\
&\quad - \mathbb{E}[\log q(\boldsymbol{z})] - \mathbb{E}[\log q(\boldsymbol{\pi})] - \mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})]
\end{aligned}$$

All expectations are with respect to $q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The terms involving expectations over the logs of the q distributions are the negative entropies of these distributions. The results for the individual terms are:

$$\mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] = \frac{1}{2}\sum_{k=1}^{K} N_k \left\{\log \widetilde{\Sigma}_k - D\lambda_k^{-1} - \nu_k \operatorname{Tr}(\boldsymbol{S}_k \boldsymbol{L}_k) - \nu_k (\overline{\boldsymbol{x}}_k - \boldsymbol{m}_k)^{\mathrm{T}} \boldsymbol{L}_k (\overline{\boldsymbol{x}}_k - \boldsymbol{m}_k) - D\log(2\pi)\right\}$$

$$\mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{\pi})] = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \log \widetilde{\pi}_k$$

$$\mathbb{E}[\log p(\boldsymbol{\pi})] = \log C(\boldsymbol{\alpha}_0) + (\alpha_0 - 1)\sum_{k=1}^{K} \log \widetilde{\pi}_k$$

$$\mathbb{E}[\log p(\boldsymbol{\mu}, \boldsymbol{\Sigma})] = \frac{1}{2}\sum_{k=1}^{K}\left\{D\log(\lambda_0/2\pi) + \log \widetilde{\Sigma}_k - \frac{D\lambda_0}{\lambda_k} - \lambda_0 \nu_k (\boldsymbol{m}_k - \boldsymbol{m}_0)^{\mathrm{T}} \boldsymbol{L}_k (\boldsymbol{m}_k - \boldsymbol{m}_0)\right\}$$
$$+ K\log B(\boldsymbol{L}_0, \nu_0) + \frac{(\nu_0 - D - 1)}{2}\sum_{k=1}^{K} \log \widetilde{\Sigma}_k - \frac{1}{2}\sum_{k=1}^{K} \nu_k \operatorname{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{L}_k\right)$$

$$\mathbb{E}[\log q(\boldsymbol{z})] = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \cdot \log r_{nk}$$

$$\mathbb{E}[\log q(\boldsymbol{\pi})] = \log C(\boldsymbol{\alpha}) + (\alpha_k - 1)\sum_{k=1}^{K} \log \widetilde{\pi}_k$$

$$\mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})] = \sum_{k=1}^{K}\left\{\frac{1}{2}\log \widetilde{\Sigma}_k + \frac{D}{2}\log\left(\frac{\lambda_k}{2\pi}\right) - \frac{D}{2} - \mathrm{H}[q(\boldsymbol{\Sigma}_k)]\right\}$$

In appendix D.1 we show the derivations for these expressions one by one. The variational lower bound is very helpful for checking the convergence of the algorithm. Each iteration of the variational E-steps and M-steps has to increase the VLB. If the VLB is not changing anymore, the algorithm is converged. [8, 7]

4.3 Variational Bayes EM for Dirichlet Process Gaussian Mixture Models

In comparison to finite Gaussian mixture models, the variational Bayes EM algorithm and the variational lower bound of the log-likelihood for Dirichlet Process Gaussian Mixture Models have some differences. These differences will be highlighted in the following sections.

4.3.1 Choice of Likelihood and Priors

The likelihood function remains the same as in the case of Gaussian mixture models:

$$p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} \tag{4.13}$$

We assume the following factored conjugate prior:

$$\begin{aligned} p(\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) &= p(\boldsymbol{z}\mid\boldsymbol{\pi})\,p(v)\,p(\boldsymbol{\mu}\mid\boldsymbol{\Sigma})\,p(\boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N}\mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\prod_{k=1}^{\infty}\mathrm{Beta}\left(v_k|\gamma_{0,1},\gamma_{0,2}\right)\prod_{k=1}^{\infty}\mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0,(\lambda_0\boldsymbol{\Sigma}_k)^{-1}\right)\mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0,\nu_0) \\ &= \left[\prod_{n=1}^{N}\prod_{k=1}^{\infty}\left(v_k\prod_{j=1}^{k-1}(1-v_j)\right)^{z_{nk}}\right]\left[\prod_{k=1}^{\infty}\frac{1}{A(\gamma_{0,1},\gamma_{0,2})}v_k^{\gamma_{0,1}-1}(1-v_k)^{\gamma_{0,2}-1}\right] \\ &\quad\left[\prod_{k=1}^{\infty}\frac{|\lambda_0\boldsymbol{\Sigma}_k|^{1/2}}{(2\pi)^{D/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{\mu}_k-\boldsymbol{m}_0)^{\mathrm{T}}\lambda_0\boldsymbol{\Sigma}_k(\boldsymbol{\mu}_k-\boldsymbol{m}_0)\right\}B(\boldsymbol{L}_0,\nu_0)|\boldsymbol{\Sigma}_k|^{(\nu_0-D-1)/2}\exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right)\right\}\right] \end{aligned} \tag{4.14}$$

As in the case of GMM we assume Normal-Wishart priors on the cluster means and precision matrices. The latent cluster assignment variables $\boldsymbol{z}_n$ are drawn from a Categorical distributions conditioned on the mixture weights.

Instead of drawing the mixture weights from a Dirichlet distribution, these are now constructed by the stick-breaking method. The number of clusters is a-priori unbounded and is learned from data. π_k in the Categorical distribution is substituted by the expression $\pi_k(v) = v_k\prod_{j=1}^{k-1}(1-v_j)$. The parameters $v = \{v_i \dots v_k\}$ are independently Beta distributed.

The hyperparameters of the Beta distributions are $\gamma_{0,1}$ and $\gamma_{0,2}$. By definition of the stick-breaking construction we set $\gamma_{0,1} = 1$. The second hyperparameter of the Beta distributions $\gamma_{0,2}$ has an important impact on the number of clusters that are effectively learned for the dataset. In section 2.5 on the Dirichlet Process we call this parameter the concentration parameter and name it α. Inspecting the Pólya Urn Sampling Scheme we observe that for a large concentration parameter it gets more likely that a new set of parameters is drawn from the base distribution H, instead of taking on previously seen values. It is thus more likely that a new cluster is created. The base distribution in DP-GMM is the Normal-Wishart prior.
$\gamma_{0,2}$ takes a comparable role as the parameter α_0 for a Dirichlet prior of the GMM in section 4.2, which also steers the number of clusters that are learned. We will assess the hyperparameter $\gamma_{0,2}$ in detail in the evaluation of DP-GLM in chapter 6.
In summary the unknown random variables are $\boldsymbol{z}$, v, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

4.3.2 Mean Field Factorization

The mean field approximation of the posterior is:

$$p(\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}|\boldsymbol{x}) \approx q(v,\boldsymbol{\mu},\boldsymbol{\Sigma})q(\boldsymbol{z}) = q(v,\boldsymbol{\mu},\boldsymbol{\Sigma})\prod_{n=1}^{N}q(\boldsymbol{z}_n)$$

Using this factorization, we will derive an approximate posterior of the following form in the next sections:

$$\begin{aligned} q^\star(\boldsymbol{z},v,\boldsymbol{\mu},\boldsymbol{\Sigma}) &= q^\star(\boldsymbol{z})\,q^\star(v,\boldsymbol{\mu},\boldsymbol{\Sigma}) \\ &= \left[\prod_{n=1}^{N}\mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{r}_n)\right]\left[\prod_{k=1}^{K-1}\mathrm{Beta}\left(v_k|\gamma_{k,1},\gamma_{k,2}\right)\prod_{k=1}^{K}\mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_k,(\lambda_k\boldsymbol{\Sigma}_k)^{-1}\right)\mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_k,\nu_k)\right] \end{aligned}$$

As before the functional form of the posterior is a result of the optimization and does not need to be specified in advance.

4.3.3 Target Distribution

For variational infinite mixture models the unnormalized log posterior is:

$$\begin{aligned}\log p(\boldsymbol{x}, \boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(\boldsymbol{z}|\boldsymbol{\pi}) + \log p(\upsilon) + \log p(\boldsymbol{\mu}|\boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) \\ &= \log \prod_{n=1}^{N} \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} + \log \prod_{n=1}^{N} \mathrm{Cat}\left(\boldsymbol{z}_n|\boldsymbol{\pi}\right) + \log \prod_{k=1}^{\infty} \mathrm{Beta}\left(\upsilon_k|\gamma_{0,1}, \gamma_{0,2}\right) \\ &\quad + \log \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\lambda_0 \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}\left(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0, \nu_0\right)\end{aligned}$$

We see that as the target distribution includes products will infinite limits. For calculating the variational posterior we will approximate these products.

4.3.4 Variational E-step: Updating $q(\boldsymbol{z})$

The form of $q^\star(\boldsymbol{z})$ can be derived by taking the expectation of the unnormalized log posterior with respect to all hidden variables except of $\boldsymbol{z}$. All terms, that do not involve $\boldsymbol{z}$ are constant when updating $q(\boldsymbol{z})$ and can be ignored.
According to Blei and Jordan (2006) [56] the infinite sums of the stick-breaking prior of the variational posterior can be truncated at an upper limit K. If this upper limit is chosen appropriately high, the variational posterior is a reasonable approximation. It is important to note that the model is still an infinite mixture model, just the variational posterior is truncated.

$$\begin{aligned}\log q^\star(\boldsymbol{z}) &= \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log p(\boldsymbol{x}, \boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \text{ const} \\ &= \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\log \prod_{n=1}^{N} \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}}\right] + \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\log \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\right] + \text{ const} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log \rho_{nk} + \text{ const} \end{aligned} \tag{4.15}$$

We defined here:

$$\begin{aligned}\log \boldsymbol{\rho_{nk}} = &\frac{1}{2}\mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log|\boldsymbol{\Sigma}_k|] - \frac{D}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] \\ &+ \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log \upsilon_k] + \sum_{j=1}^{k-1} \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[(\log(1-\upsilon_j)\right]\end{aligned}$$

All expectations have been stated before, except of the expectation of the terms that involve the Beta distributed υ_k. Refering to [56] we get:

$$\begin{aligned}\mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log \upsilon_i] &= \Psi\left(\gamma_{i,1}\right) - \Psi\left(\gamma_{i,1} + \gamma_{i,2}\right) \qquad \text{with } \psi(x) = \frac{d}{dx}\log(\Gamma(x)) = \frac{\Gamma'(x)}{\Gamma(x)} \\ \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log(1-\upsilon_i)] &= \Psi\left(\gamma_{i,2}\right) - \Psi\left(\gamma_{i,1} + \gamma_{i,2}\right)\end{aligned}$$

The first expectation can be derived by using the fact, that for exponential families the expectation of the log of a random variable is equivalent to the maximum likelihood solution. The second expectation is a result of symmetry properties of the Beta distribution. [64]
By exponentiating equation 4.15 we get a Categorical distribution, just as in the case of finite mixture models:

$$q^\star(\boldsymbol{z}) \propto \prod_{n=1}^{N} \prod_{k=1}^{K} \rho_{nk}^{z_{nk}}$$

The normalized distribution is given by:

$$q^\star(\boldsymbol{z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} r_{nk}^{z_{nk}} \tag{4.16}$$

The responsibilities are:

$$r_{nk} \propto \tilde{\Sigma}_k^{\frac{1}{2}} \exp\left(-\frac{D}{2\lambda_k} - \frac{\nu_k}{2}(\boldsymbol{x}_n - \boldsymbol{m}_k)^T \boldsymbol{L}_k (\boldsymbol{x}_n - \boldsymbol{m}_k) + \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log \upsilon_k] + \sum_{j=1}^{k-1} \mathbb{E}_{q(\upsilon,\boldsymbol{\mu},\boldsymbol{\Sigma})}[\log(1-\upsilon_i)]\right)$$

$$r_{nk} = \frac{\rho_{nk}}{\sum_j \rho_{nj}}$$

4.3.5 Variational M-step: Updating $q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

To derive $q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ we have to take the expectation of the unnormalized log posterior with respect to $\boldsymbol{z}$. All terms, that do not involve v, $\boldsymbol{\mu}$, or $\boldsymbol{\Sigma}$ are absorbed into the constant term. The infinite sums are again truncated at K [56].
We get for $q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$
\begin{aligned}
\log q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \mathbb{E}_{q(\boldsymbol{z})}[\log p(\boldsymbol{x}, \boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \text{ const} \\
&= \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{n=1}^{N} \text{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\right\}\right] + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{k=1}^{\infty} \text{Beta}\left(v_k|\gamma_{0,1}, \gamma_{0,2}\right)\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{n=1}^{N}\prod_{k=1}^{\infty} \text{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}}\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{k=1}^{\infty} \text{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\lambda_0\boldsymbol{\Sigma}_k)^{-1}\right)\text{Wi}\left(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0, \nu_0\right)\right\}\right] + \text{const} \\
&= \prod_{n=1}^{N} \text{E}_{q(z)}\left[\log p\left(z_n|\boldsymbol{\pi}\right)\right] + \sum_{k=1}^{K}(\gamma_{0,1}-1)\log v_k + (\gamma_{0,2}-1)\log(1-v_k) \\
&\quad + \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{z})}[z_{nk}]\left(\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k\right)^T \Sigma_k \left(\boldsymbol{x}_n - \mu_k\right)\right) \\
&\quad + \sum_{k=1}^{K} \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\left(\boldsymbol{\mu}_k - \boldsymbol{m}_0\right)^T \lambda_0\boldsymbol{\Sigma}_k\left(\boldsymbol{\mu}_k - \boldsymbol{m}_0\right) + \frac{\nu_0 - D - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right) + \text{const}
\end{aligned}
$$

In this equation the unknown expectation $\text{E}_{q(z)}\left[\log p\left(z_n|\boldsymbol{\pi}\right)\right]$ remains to be calculated. We use the following result from [56]:

$$
\text{E}_{q(\boldsymbol{z})}\left[\log p\left(\boldsymbol{z}_n|\boldsymbol{\pi}\right)\right] = \text{E}_{q(\boldsymbol{z})}\left[\log \text{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\right] = \sum_{k=1}^{K}\sum_{j=k+1}^{K} r_{nj}\log(1-v_k) + r_{nk}\log v_k
$$

We can see that the whole expression for $\log q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ factorizes into terms, which only depend on v and into terms, which only depend on $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. We can write the approximate posterior as the following factorized distribution:

$$
\begin{aligned}
\log q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log q^\star(v) + \sum_{k=1}^{K} \log q^\star\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \\
q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= q^\star(v)\prod_{k=1}^{K} q^\star\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)
\end{aligned}
$$

Using the fact that $\mathbb{E}_{q(\boldsymbol{z})}[z_{nk}] = r_{nk}$ for the Categorical distribution $q(\boldsymbol{z})$ (see equation C.6), we get as a result for the v term:

$$
\begin{aligned}
\log q^\star(v) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{j=k+1}^{K} r_{nj}\log(1-v_k) + r_{nk}\log v_k \\
&\quad + \sum_{k=1}^{K}\left(\gamma_{0,1}-1\right)\log v_k + \left(\gamma_{0,2}-1\right)\log(1-v_k) \\
&= \sum_{k=1}^{K}\log(1-v_k)\left[\sum_{n=1}^{N}\sum_{j=k+1}^{K} r_{nj} + \gamma_{0,2} - 1\right] + \sum_{k=1}^{K}\log(v_k)\left[\sum_{n=1}^{N} r_{nk} + \gamma_{0,1} - 1\right] + \text{const}
\end{aligned}
$$

Exponentiating on both sides, we recognize $q^\star(v)$ as a product of Beta distributions:

$$
\begin{aligned}
q^\star(v) &\propto \prod_{k=1}^{K} v_k^{\sum_{k=1}^{K} r_{nk} + \gamma_{0,1} - 1}(1-v_k)^{\sum_{n=1}^{N}\sum_{j=k+1}^{K} r_{nj} + \gamma_{0,2} - 1} \\
&= \prod_{k=1}^{K} \frac{1}{A\left(\gamma_{k,1}, \gamma_{k,2}\right)} v_k^{\gamma_{k,1}-1}(1-v_k)^{\gamma_{k,2}-1} \\
&= \prod_{k=1}^{K-1} \text{Beta}(v_k|\gamma_{0,k}, \gamma_{1,k})
\end{aligned}
$$

The variational parameters of the Beta distributions are:

$$\gamma_{k,1} = \sum_{n=1}^{N} r_{nk} + \gamma_{0,1}$$
$$\gamma_{k,2} = \sum_{n=1}^{N} \sum_{j=k+1}^{K} r_{nj} + \gamma_{0,2}$$

We stop the product of the Beta distributions at $K-1$ as the truncated stick-breaking construction defines, that $q(v_k = 1) = 1$ and thus all mixture proportions $\pi(v)$ are equal to zero for $k > K$ [56].
The terms that depend on $\boldsymbol{\mu}$ and on $\boldsymbol{\Sigma}$ are the same as for the variational Gaussian mixture model.

4.3.6 Variational Lower Bound on the Marginal Likelihood

For the variational lower bound on the marginal likelihood of the Dirichlet process mixture of Gaussians we get:

$$\begin{aligned}
\mathrm{L} &= \sum_{\boldsymbol{z}} \iiint q(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \log \left\{ \frac{p(\boldsymbol{x}, \boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{q(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \right\} \mathrm{d}v \, \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\Sigma} \\
&= \mathbb{E}[\log p(\boldsymbol{x}, \boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})] - \mathbb{E}[\log q(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})] \\
&= \mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{\pi})] + \mathbb{E}[\log p(v)] + \mathbb{E}[\log p(\boldsymbol{\mu}, \boldsymbol{\Sigma})] \\
&\quad - \mathbb{E}[\log q(\boldsymbol{z})] - \mathbb{E}[\log q(v)] - \mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})]
\end{aligned}$$

All expectations are with respect to $q(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.
With exception of $\mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{\pi})]$, $\mathbb{E}[\log p(v)]$ and $\mathbb{E}[\log q(v)]$ all expectations are the same as in the case of GMM. We do not derive the remaining terms in detail.

4.4 Variational Bayes EM for Dirichlet Process Mixtures of Generalized Linear Models

In this section, we build upon the results for finite and infinite Gaussian mixture models and extend the variational Bayesian treatment to the regression case. A summary of the results of this section can be found in appendix B.

4.4.1 Choice of Likelihood and Priors

The likelihood function of Dirichlet Process Mixtures of Generalized Linear Models, assuming linear models, is:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{x} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= p(\boldsymbol{y} | \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{V}) p(\boldsymbol{x} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{y}_n | \boldsymbol{\beta}_k \boldsymbol{X}_n, \boldsymbol{V}_k^{-1}\right)^{z_{nk}} \mathrm{N}\left(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\frac{|\boldsymbol{V}_k|^{1/2}}{(2\pi)^{d/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)^{\mathrm{T}} \boldsymbol{V}_k (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n) \right\} \right]^{z_{nk}} \\ &\quad \left[\frac{|\boldsymbol{\Sigma}_k|^{1/2}}{(2\pi)^{m/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right\} \right]^{z_{nk}} \end{aligned} \tag{4.17}$$

m is the length of the input vector $\boldsymbol{x}$ and d is the length of the target vector y. The regression coefficients $\boldsymbol{\beta}$ and the regressor matrix $\boldsymbol{X}$ have the following shape:

$$\boldsymbol{X} = [\, 1 \quad \boldsymbol{x} \,]^T \quad \text{and} \quad \boldsymbol{\beta} = [\, \boldsymbol{\beta}_0 \quad \boldsymbol{\beta}_1 \,]$$

$\boldsymbol{\beta}_1$ is a d by m matrix and represents the slope parameters. $\boldsymbol{\beta}_0$ encodes the intercept and has a length of d.

We assume the following factored conjugate prior:

$$\begin{aligned} p(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= p(\boldsymbol{z} \mid \boldsymbol{\pi}) p(\upsilon) p(\boldsymbol{\mu} \mid \boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) p(\boldsymbol{\beta} \mid \boldsymbol{V}) p(\boldsymbol{V}) \\ &= \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{\pi}) \prod_{k=1}^{\infty} \mathrm{Beta}\left(\upsilon_k | \gamma_{0,1}, \gamma_{0,2}\right) \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_0, (\lambda_0 \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k | \boldsymbol{L}_0, \nu_0) \\ &\quad \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_0, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_0\right) \mathrm{Wi}(\boldsymbol{V}_k | \boldsymbol{P}_0, \eta_0) \\ &= \left[\prod_{n=1}^{N} \prod_{k=1}^{\infty} \left(\upsilon_k \prod_{j=1}^{k-1} (1 - \upsilon_j) \right)^{z_{nk}} \right] \left[\prod_{k=1}^{\infty} \frac{1}{A(\gamma_{0,1}, \gamma_{0,2})} \upsilon_k^{\gamma_{0,1}-1} (1 - \upsilon_k)^{\gamma_{0,2}-1} \right] \\ &\quad \left[\prod_{k=1}^{\infty} \frac{|\lambda_0 \boldsymbol{\Sigma}_k|^{1/2}}{(2\pi)^{m/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{\mu}_k - \boldsymbol{m}_0)^{\mathrm{T}} \lambda_0 \boldsymbol{\Sigma}_k (\boldsymbol{\mu}_k - \boldsymbol{m}_0) \right\} B(\boldsymbol{L}_0, \nu_0) |\boldsymbol{\Sigma}_k|^{(\nu_0 - m - 1)/2} \exp\left\{ -\frac{1}{2} \mathrm{Tr}\left(\boldsymbol{L}_0^{-1} \boldsymbol{\Sigma}_k\right) \right\} \right] \\ &\quad \left[\prod_{k=1}^{\infty} \frac{|\boldsymbol{K}_0|^{d/2} |\boldsymbol{V}_k|^{m/2}}{(2\pi)^{m/2}} \exp\left(-\frac{1}{2} \mathrm{Tr}\left((\boldsymbol{\beta}_k - \boldsymbol{M}_0)^{\mathrm{T}} \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0) \boldsymbol{K}_0 \right) \right) D(\boldsymbol{P}_0, \eta_0) |\boldsymbol{V}_k|^{(\eta_0 - d - 1)/2} \exp\left\{ -\frac{1}{2} \mathrm{Tr}\left(\boldsymbol{P}_0^{-1} \boldsymbol{V}_k\right) \right\} \right] \end{aligned} \tag{4.18}$$

The first four prior densities in equation 4.18 are the same as for the case of infinite mixtures of Gaussians. As discussed in chapter 2.2 on Bayesian linear regression we use a Matrix-Normal-Wishart prior for the regression coefficients $\boldsymbol{\beta}_k$ and the precision matrix $\boldsymbol{V}_k$ of the linear models. The dimension of the parameters of the Matrix-Normal distribution is: $\boldsymbol{M}_0$ is d by $m+1$, $\boldsymbol{V}_k$ is d by d, and $\boldsymbol{K}_0$ is $m+1$ by $m+1$. The input dimension is $m+1$ as the regressor matrix $\boldsymbol{X}$ has a row of ones.

By definition of the stick-breaking process, the first hyperparameter of the Beta distribution $\gamma_{0,1}$ is set to 1. The second hyperparameter $\gamma_{0,2}$ has a crucial impact on the number of clusters that are effectively learned from the data. In the foundations of this thesis (section 2.5) we denote this parameter as α and call it *concentration parameter*. Looking at the Pólya urn sampling scheme, we observe, that for a large concentration parameter it is more likely to draw a new set of parameters from the base distribution H. It is less likely that the parameters take on values from an existing cluster. The base distribution in DP-GLM is the composite distribution of the Normal-Matrix-Wishart prior on the linear regression model and the Normal-Wishart prior on the cluster parameters. We will asses the impact of the hyperparameter $\gamma_{0,2}$ in chapter 6.
In summary the unknown random variables are $\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}$ and $\boldsymbol{V}$.

4.4.2 Mean Field Factorization

We assume the following mean field factorization of the posterior:

$$p(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V} \mid \boldsymbol{x}) \approx q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) q(\boldsymbol{z}) = q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) \prod_{n=1}^{N} q(\boldsymbol{z}_n)$$

Using this factorization, we will derive an approximate posterior of the following form in the next sections:

$$\begin{aligned} q^\star(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= q^\star(\boldsymbol{z}) q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) \\ &= \left[\prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{r}_n)\right] \left[\prod_{k=1}^{K-1} \mathrm{Beta}\left(v_k | \gamma_{k,1}, \gamma_{k,2}\right) \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_k, (\lambda_k \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k | \boldsymbol{L}_k, \nu_k) \right. \\ &\quad \left. \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_k, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_k\right) \mathrm{Wi}(\boldsymbol{V}_k | \boldsymbol{P}_k, \eta_k)\right] \end{aligned}$$

K is the truncation level of the stick-breaking construction.

4.4.3 Target Distribution

For DP-GLM the unnormalized log posterior over the data and hidden variables is:

$$\begin{aligned} \log p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= \log p(\boldsymbol{y}, \boldsymbol{x} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) + \log p(\boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) \\ &= \log p(\boldsymbol{y} | \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{\Sigma}) + \log p(\boldsymbol{x} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(\boldsymbol{z} | \boldsymbol{\pi}) + \log p(v) + \log p(\boldsymbol{\mu} | \boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) \\ &\quad + \log p(\boldsymbol{\beta} | \boldsymbol{V}) p(\boldsymbol{V}) \\ &= \log \prod_{n=1}^{N} \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{y}_n | \boldsymbol{\beta}_k \boldsymbol{X}_n, \boldsymbol{V}_k^{-1}\right)^{z_{nk}} + \log \prod_{n=1}^{N} \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} \\ &\quad + \log \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{\pi}) + \log \prod_{k=1}^{\infty} \mathrm{Beta}\left(v_k | \gamma_{0,1}, \gamma_{0,2}\right) \\ &\quad + \log \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_0, (\lambda_0 \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k | \boldsymbol{L}_0, \nu_0) \\ &\quad + \log \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_0, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_0\right) \mathrm{Wi}(\boldsymbol{V}_k | \boldsymbol{P}_0, \eta_0) \end{aligned}$$

This distribution is plugged into the mean field update equation (4.2) for deriving the form of the approximate posterior.

4.4.4 Variational E-step: Updating $q(\boldsymbol{z})$

The form of $q^\star(\boldsymbol{z})$ can be derived by taking the expectation of the unnormalized log posterior with respect to all hidden variables except of $\boldsymbol{z}$. All terms, that do not involve $\boldsymbol{z}$ are constant when updating $q(\boldsymbol{z})$ and can be ignored.
We again truncate the infinite sums of the variational posterior at K. That does not apply to the true posterior of the model, which still has an unbounded number of components. [56]

$$\begin{aligned} \log q^\star(\boldsymbol{z}) &= \mathbb{E}_{q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}[\log p(\boldsymbol{x}, \boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})] + \text{ const} \\ &= \mathbb{E}_{q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}\left[\log \prod_{n=1}^{N} \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{y}_n | \boldsymbol{\beta}_k \boldsymbol{X}_n, \boldsymbol{V}_k^{-1}\right)^{z_{nk}}\right] + \mathbb{E}_{q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}\left[\log \prod_{n=1}^{N} \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}}\right] \\ &\quad + \mathbb{E}_{q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}\left[\log \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{\pi})\right] + \text{ const} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log \rho_{nk} + \text{ const} \end{aligned} \tag{4.19}$$

We defined here:

$$\begin{aligned}\log \boldsymbol{\rho_{nk}} =& \frac{1}{2}\mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}[\log|V_k|] - \frac{d}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}\left[(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V_k}(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)\right] \\ &+ \frac{1}{2}\mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}[\log|\boldsymbol{\Sigma}_k|] - \frac{m}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\,\boldsymbol{\Sigma}_k\,(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] \\ &+ \mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}[\log \nu_k] + \sum_{j=1}^{k-1}\mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}\left[(\log(1-\nu_j)\right]\end{aligned}$$

The expectations to a great extend are analogous to DP-GMM. We need to calculate one more expectation of a quadratic term. We use the definition of the expectation:

$$\mathbb{E}_{q(\boldsymbol{\beta},V)}\left[(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V_k}(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)\right] = \int \mathbb{E}_{q(\boldsymbol{\beta})}\left[(\boldsymbol{\beta}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^{\mathrm{T}}\boldsymbol{V_k}(\boldsymbol{\beta}_k\boldsymbol{X}_n - \boldsymbol{y}_n)\right]\cdot q^\star(\boldsymbol{V}_k)\,d\boldsymbol{V}_k$$

The inner expectation is with respect to a Matrix-Normal distribution. We use the following two expectations of a Matrix-Normal distribution: [65, 66]

$$\begin{aligned}\mathbb{E}(\boldsymbol{\beta}_k) &= \boldsymbol{M}_k \\ \mathbb{E}(\boldsymbol{\beta}_k\boldsymbol{A}\boldsymbol{\beta}_k^T) &= \boldsymbol{M}_k\boldsymbol{A}\boldsymbol{M}_k^T + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{A}\right\}\boldsymbol{V}_k^{-1}\end{aligned}$$

Using these expressions we get for the inner expectation:

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{\beta})}\left[(\boldsymbol{\beta}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^{\mathrm{T}}\boldsymbol{V_k}(\boldsymbol{\beta}_k\boldsymbol{X}_n - \boldsymbol{y}_n)\right] &= \mathbb{E}_{q(\boldsymbol{\beta})}\left[\mathrm{Tr}\left\{\boldsymbol{V_k}(\boldsymbol{\beta}_k\boldsymbol{X}_n - \boldsymbol{y}_n)(\boldsymbol{\beta}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^{\mathrm{T}}\right\}\right] \\ &= \mathrm{Tr}\left\{\boldsymbol{V}_k\left(\mathbb{E}_{q(\boldsymbol{\beta})}\left[\boldsymbol{\beta}_k\boldsymbol{X}_n\boldsymbol{X}_n^T\boldsymbol{\beta}_k^T\right] - \mathbb{E}_{q(\boldsymbol{\beta})}[\boldsymbol{\beta}_k]\boldsymbol{X}_n\boldsymbol{y}_n^T - \boldsymbol{y}_n\boldsymbol{X}_n^T\mathbb{E}_{q(\boldsymbol{\beta})}\left[\boldsymbol{\beta}_k^T\right] + \boldsymbol{y}_n\boldsymbol{y}_n^T\right)\right\} \\ &= \mathrm{Tr}\left\{\boldsymbol{V}_k\left(\boldsymbol{M}_k\boldsymbol{X}_n\boldsymbol{X}_n^T\boldsymbol{M}_k^T + \left(\mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\}\boldsymbol{V}_k^{-1}\right) - \boldsymbol{M}_k\boldsymbol{X}_n\boldsymbol{y}_n^T - \boldsymbol{y}_n\boldsymbol{X}_n^T\boldsymbol{M}_k^T + \boldsymbol{y}_n\boldsymbol{y}_n^T\right)\right\} \\ &= \mathrm{Tr}\left\{\boldsymbol{V}_k\left((\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T + \left(\mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\}\boldsymbol{V}_k^{-1}\right)\right)\right\} \\ &= (\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T\,\boldsymbol{V}_k\,(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n) + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\}\end{aligned}$$

Substituting back yields:

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{\beta},V)}\left[(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V_k}(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)\right] &= \int\left[(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T\,\boldsymbol{V}_k\,(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n) + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\}\right]\cdot q^\star(\boldsymbol{V}_k)\,d\boldsymbol{V}_k \\ &= \mathbb{E}_{q(V)}\left[(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T\,\boldsymbol{V}_k\,(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)\right] + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} \\ &= \mathrm{Tr}\left\{\mathbb{E}_{q(V)}[\boldsymbol{V}_k](\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T\right\} + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} \\ &= \mathrm{Tr}\left\{\eta_k\boldsymbol{P}_k\,(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T\right\} + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} \\ &= \eta_k\,(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n)^T\,\boldsymbol{P}_k\,(\boldsymbol{M}_k\boldsymbol{X}_n - \boldsymbol{y}_n) + \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\}\end{aligned}$$

We here used the expectation of the Wishart distribution (equation C.39).

As for GMM and DP-GMM we exponentiate equation 4.19 and get a Categorical distribution:

$$q^\star(\boldsymbol{z}) \propto \prod_{n=1}^{N}\prod_{k=1}^{K}\rho_{nk}^{z_{nk}}$$

The normalized distribution is given by:

$$q^\star(\boldsymbol{z}) = \prod_{n=1}^{N}\prod_{k=1}^{K}r_{nk}^{z_{nk}} \tag{4.20}$$

The responsibilities are:

$$\begin{aligned}r_{nk} \propto\ & \tilde{\boldsymbol{V}}_k^{\frac{1}{2}}\tilde{\boldsymbol{\Sigma}}_k^{\frac{1}{2}}\exp\Bigg(-\frac{m}{2\lambda_k} - \frac{\nu_k}{2}(\boldsymbol{x}_n - \boldsymbol{m}_k)^T\,\boldsymbol{L}_k\,(\boldsymbol{x}_n - \boldsymbol{m}_k) \\ & - \frac{1}{2}\mathrm{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} - \frac{\eta_k}{2}(\boldsymbol{y}_n - \boldsymbol{M}_k\boldsymbol{X}_n)^T\,\boldsymbol{P}_k\,(\boldsymbol{y}_n - \boldsymbol{M}_k\boldsymbol{X}_n) \\ & + \mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}[\log\nu_k] + \sum_{j=1}^{k-1}\mathbb{E}_{q(\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta},V)}[\log(1-\nu_i)]\Bigg) \\ r_{nk} =\ & \frac{\rho_{nk}}{\sum_j\rho_{nj}}\end{aligned}$$

$\tilde{\boldsymbol{V}}_k$ is calculated in an analogous way to $\tilde{\boldsymbol{\Sigma}}_k$:

$$\log \tilde{\boldsymbol{V}}_k = \mathbb{E}\left[\log |\boldsymbol{V}_k|\right] = \sum_{j=1}^{d} \psi\left(\frac{\eta_k + 1 - j}{2}\right) + d \log 2 + \log |\boldsymbol{V}_k|$$

The expectations with respect to $q(v)$ of the Beta distribution are the same as for DP-GMM.

4.4.5 Variational M-step: Updating $q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})$

To derive $q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})$ we have to take the expectation of the unnormalized log posterior with respect to $\boldsymbol{z}$. All terms, that do not involve v, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\beta}$ or $\boldsymbol{V}$ are absorbed into the constant term. We truncate the infinite sums of the approximate posterior at K [56].
We derive $q(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})$ as follows:

$$\begin{aligned}
\log q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= \mathbb{E}_{q(\boldsymbol{z})}[\log p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, v, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \text{ const} \\
&= \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{n=1}^{N} \text{Cat}(\boldsymbol{z}_n|\boldsymbol{\pi})\right\}\right] + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{k=1}^{\infty} \text{Beta}\left(v_k|\gamma_{0,1}, \gamma_{0,2}\right)\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{n=1}^{N}\prod_{k=1}^{\infty} \text{N}\left(\boldsymbol{y}_n|\boldsymbol{\beta}_k \boldsymbol{X}_n, \boldsymbol{V}_k^{-1}\right)^{z_{nk}}\right\}\right] + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{n=1}^{N}\prod_{k=1}^{\infty} \text{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}}\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{k=1}^{\infty} \text{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\lambda_0 \boldsymbol{\Sigma}_k)^{-1}\right) \text{Wi}\left(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0, \nu_0\right)\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z})}\left[\log\left\{\prod_{k=1}^{\infty} \text{N}\left(\boldsymbol{\beta}_k|\boldsymbol{M}_0, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_0\right) \text{Wi}\left(\boldsymbol{V}_k|\boldsymbol{P}_0, \eta_0\right)\right\}\right] + \text{const} \\
&= \prod_{n=1}^{N} \text{E}_{q(z)}\left[\log p\left(z_n|\boldsymbol{\pi}\right)\right] + \sum_{k=1}^{K} (\gamma_{0,1} - 1) \log v_k + (\gamma_{0,2} - 1) \log(1 - v_k) \\
&\quad + \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{z})}\left[z_{nk}\right]\left(\frac{1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}(\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)^{\text{T}} \boldsymbol{V}_{\boldsymbol{k}} (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)\right) \\
&\quad + \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{z})}\left[z_{nk}\right]\left(\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k\right)^T \Sigma_k \left(\boldsymbol{x}_n - \mu_k\right)\right) \\
&\quad + \sum_{k=1}^{K} \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\left(\boldsymbol{\mu}_k - \boldsymbol{m}_0\right)^T \lambda_0 \boldsymbol{\Sigma}_k \left(\boldsymbol{\mu}_k - \boldsymbol{m}_0\right) + \frac{\nu_0 - m - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right) \\
&\quad + \sum_{k=1}^{K} \frac{m}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\text{Tr}\left((\boldsymbol{\beta}_k - \boldsymbol{M}_0)^{\text{T}} \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right) + \frac{\eta_0 - d - 1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k\right) + \text{const}
\end{aligned}$$

We see that the expression for $\log q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})$ factorizes in the following way:

$$\log q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) = \log q^\star(v) + \sum_{k=1}^{K} \log q^\star\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) + \sum_{k=1}^{K} \log q^\star\left(\boldsymbol{\beta}_k, \boldsymbol{V}_k\right)$$

$$q^\star(v, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) = q^\star(v) \prod_{k=1}^{K} q^\star\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \prod_{k=1}^{K} q^\star\left(\boldsymbol{\beta}_k, \boldsymbol{V}_k\right)$$

For $q^\star(v)$ the result is the same as for DP-GMM and for $q^\star(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the update equations are the same as for GMM and for DP-GMM.

We therefore only need to collect the $\boldsymbol{\beta}_k$ and $\boldsymbol{V}_k$ terms:

$$\begin{aligned}
\log q^\star(\boldsymbol{\beta}_k, \boldsymbol{V}_k) \propto \sum_{n=1}^{N} r_{nk}\left(\frac{1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}(\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)^{\text{T}} \boldsymbol{V}_{\boldsymbol{k}} (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)\right) \\
+ \frac{m}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\text{Tr}\left((\boldsymbol{\beta}_k - \boldsymbol{M}_0)^{\text{T}} \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right) + \frac{\eta_0 - d - 1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k\right)
\end{aligned} \tag{4.21}$$

We assume that the optimal form of $q^\star(\boldsymbol{\beta}_k, \boldsymbol{V}_k)$ is a Matrix-Normal-Wishart distribution:

$$
\begin{aligned}
q^\star(\boldsymbol{\beta}, \boldsymbol{V}) &= \prod_{k=1}^{K} q^\star(\boldsymbol{\beta}_k, \boldsymbol{V}_k) = \prod_{k=1}^{K} q^\star(\boldsymbol{\beta}_k | \boldsymbol{V}_k) q^\star(\boldsymbol{V}_k) = \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_k, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_k\right) \mathrm{Wi}\left(\boldsymbol{V}_k | \boldsymbol{P}_k, \eta_k\right) \\
&= \prod_{k=1}^{K} \frac{|\boldsymbol{K}_k|^{d/2} |\boldsymbol{V}_k|^{m/2}}{(2\pi)^{m/2}} \exp\left(-\frac{1}{2} \mathrm{Tr}\left((\boldsymbol{\beta}_k - \boldsymbol{M}_k)^{\mathrm{T}} \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k) \boldsymbol{K}_k\right)\right) \\
&\quad D(\boldsymbol{P}_k, \eta_k) |\boldsymbol{V}_k|^{(\eta_k - d - 1)/2} \exp\left\{-\frac{1}{2} \mathrm{Tr}\left(\boldsymbol{P}_k^{-1} \boldsymbol{V}_k\right)\right\}
\end{aligned}
\tag{4.22}
$$

Given this functional form, we can do a comparison of coefficients of equation 4.21 and of the assumed Matrix-Normal-Wishart distribution in equation 4.22.

For finding $q^\star(\boldsymbol{\beta}_k \mid \boldsymbol{V}_k)$ we gather all terms in equation 4.21 that contain $\boldsymbol{\beta}_k$:

$$
\begin{aligned}
&-\frac{1}{2} \sum_{n=1}^{N} r_{nk} (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)^{\mathrm{T}} \boldsymbol{V}_{\boldsymbol{k}} (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n) - \frac{1}{2} \mathrm{Tr}\left((\boldsymbol{\beta}_k - \boldsymbol{M}_0)^{\mathrm{T}} \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0) \boldsymbol{K}_0\right) \\
&= -\frac{1}{2} \sum_{n=1}^{N} r_{nk} \mathrm{Tr}\left(\boldsymbol{V}_{\boldsymbol{k}} (\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)(\boldsymbol{y}_n - \boldsymbol{\beta}_k \boldsymbol{X}_n)^{\mathrm{T}}\right) - \frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0) \boldsymbol{K}_0 (\boldsymbol{\beta}_k - \boldsymbol{M}_0)^{\mathrm{T}}\right) \\
&= -\frac{1}{2} \sum_{n=1}^{N} r_{nk} \mathrm{Tr}\left(\boldsymbol{V}_{\boldsymbol{k}} \left(\boldsymbol{y}_n \boldsymbol{y}_n^T - \boldsymbol{y}_n \boldsymbol{X}_n^T \boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k \boldsymbol{X}_n \boldsymbol{y}_n^T + \boldsymbol{\beta}_k \boldsymbol{X}_n \boldsymbol{X}_n^T \boldsymbol{\beta}_k^T\right)\right) \\
&- \frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}_k \left(\boldsymbol{\beta}_k \boldsymbol{K}_0 \boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k \boldsymbol{K}_0 \boldsymbol{M}_0^T - \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{\beta}_k^T + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T\right)\right)
\end{aligned}
$$

The quadratic term with respect to $\boldsymbol{\beta}_k$ is:

$$
(\text{quadratic}) = -\frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}_k \boldsymbol{\beta}_k \left[\sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{X}_n^T + \boldsymbol{K}_0\right] \boldsymbol{\beta}_k^T\right)
$$

The linear term with respect to $\boldsymbol{\beta}_k$ is:

$$
(\text{linear}) = -\frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}_k \boldsymbol{\beta}_k \left[-\sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{y}_n^T - \boldsymbol{K}_0 \boldsymbol{M}_0^T\right]\right)
$$

We compare the coefficients with the relevant term of the Matrix-Normal distribution in equation 4.22, which is:

$$
-\frac{1}{2} \mathrm{Tr}(\boldsymbol{\beta}_k - \boldsymbol{M}_k)^{\mathrm{T}} \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k) \boldsymbol{K}_k = -\frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k) \boldsymbol{K}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k)^{\mathrm{T}}\right)
$$

We thus get for the variational parameters of the Matrix-Normal distribution:

$$
\begin{aligned}
\boldsymbol{K}_k &= \sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{X}_n^T + \boldsymbol{K}_0 = N_k \boldsymbol{R}_k + N_k \overline{\boldsymbol{X}}_k \overline{\boldsymbol{X}}_k^T + \boldsymbol{K}_0 \\
\boldsymbol{M}_k &= \left[\sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{y}_n^T + \boldsymbol{K}_0 \boldsymbol{M}_0^T\right] \boldsymbol{K}_k^{-1} = \left[N_k \overline{\boldsymbol{X}\boldsymbol{Y}}_k + \boldsymbol{K}_0 \boldsymbol{M}_0^T\right] \boldsymbol{K}_k^{-1}
\end{aligned}
$$

We here used the following definitions and transformations:

$$
\begin{aligned}
\overline{\boldsymbol{X}}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{X}_n \\
\boldsymbol{R}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\boldsymbol{X}_i-\overline{\boldsymbol{X}}_k)(\boldsymbol{X}_i-\overline{\boldsymbol{X}}_k)^T \\
\sum_{n=1}^{N} r_{nk}\boldsymbol{X}_n\boldsymbol{X}_n^T &= \sum_{n=1}^{N} r_{nk}\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k+\overline{\boldsymbol{X}}_k\right)\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k+\overline{\boldsymbol{X}}_k\right)^T \\
&= \sum_{n=1}^{N} r_{nk}\left[\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)^T+\overline{\boldsymbol{X}}_k\overline{\boldsymbol{X}}_k^T+2\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)\overline{\boldsymbol{X}}_k^T\right] \\
&= \sum_{n=1}^{N} r_{nk}\left[\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)^T\right]+\sum_{n=1}^{N} r_{nk}\left[\overline{\boldsymbol{X}}_k\overline{\boldsymbol{X}}_k^T\right]+\sum_{n=1}^{N} r_{nk}\left[2\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)\overline{\boldsymbol{X}}_k^T\right] \\
&= N_k\boldsymbol{R}_k+N_k\overline{\boldsymbol{X}}_k\overline{\boldsymbol{X}}_k^T+2\sum_{n=1}^{N} r_{nk}\left[\left(\boldsymbol{X}_n-\overline{\boldsymbol{X}}_k\right)\overline{\boldsymbol{X}}_k^T\right] \\
&= N_k\boldsymbol{R}_k+N_k\overline{\boldsymbol{X}}_k\overline{\boldsymbol{X}}_k^T+2\left[\left(N_k\overline{\boldsymbol{X}}_k-N_k\overline{\boldsymbol{X}}_k\right)\overline{\boldsymbol{X}}_k^T\right] \\
&= N_k\boldsymbol{R}_k+N_k\overline{\boldsymbol{X}}_k\overline{\boldsymbol{X}}_k^T \\
\overline{\boldsymbol{XY}}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{X}_n\boldsymbol{y}_n^T
\end{aligned}
\tag{4.23}
$$

For obtaining the parameters $\boldsymbol{P}_k$ and η_k of the Wishart distribution $q^\star(\boldsymbol{V}_k)$, we need the following relation:

$$\log q^\star(\boldsymbol{V}_k) = \log q^\star(\boldsymbol{\beta}_k, \boldsymbol{V}_k) - \log q^\star(\boldsymbol{\beta}_k|\boldsymbol{V}_k)$$

For finding $q^\star(\boldsymbol{V}_k)$ we collect all terms in equation 4.21 that contain $\boldsymbol{V}_k$ and subtract the $\log q^\star(\boldsymbol{\beta}_k \mid \boldsymbol{V}_k)$ term. We also use the cyclic property of the trace (equation 16 in [63]) and the definition of N_k:

$$
\begin{aligned}
\log q^\star(\boldsymbol{\beta}_k, \boldsymbol{V}_k) \propto &\sum_{n=1}^{N} r_{nk}\left(\frac{1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}(\boldsymbol{y}_n-\boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V}_k(\boldsymbol{y}_n-\boldsymbol{\beta}_k\boldsymbol{X}_n)\right) \\
&+\frac{m}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\beta}_k-\boldsymbol{M}_0)^{\mathrm{T}}\boldsymbol{V}_k(\boldsymbol{\beta}_k-\boldsymbol{M}_0)\boldsymbol{K}_0\right) \\
&+\frac{\eta_0-d-1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k\right) \\
&-\frac{m}{2}\log|\boldsymbol{V}_k| + \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\beta}_k-\boldsymbol{M}_k)^{\mathrm{T}}\boldsymbol{V}_k(\boldsymbol{\beta}_k-\boldsymbol{M}_k)\boldsymbol{K}_k\right) \\
\propto &\frac{N_k}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\operatorname{Tr}\left(\sum_{n=1}^{N} r_{nk}(\boldsymbol{y}_n-\boldsymbol{\beta}_k\boldsymbol{X}_n)(\boldsymbol{y}_n-\boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V}_k\right) \\
&+\frac{m}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\beta}_k-\boldsymbol{M}_0)\boldsymbol{K}_0(\boldsymbol{\beta}_k-\boldsymbol{M}_0)^{\mathrm{T}}\boldsymbol{V}_k\right) \\
&+\frac{\eta_0-d-1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k\right) \\
&-\frac{m}{2}\log|\boldsymbol{V}_k| + \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\beta}_k-\boldsymbol{M}_k)\boldsymbol{K}_k(\boldsymbol{\beta}_k-\boldsymbol{M}_k)^{\mathrm{T}}\boldsymbol{V}_k\right) \\
= &\frac{\eta_0-d-1+N_k}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\operatorname{Tr}[\boldsymbol{T}\boldsymbol{V}_k]
\end{aligned}
$$

Here we defined $\boldsymbol{T}$ as:

$$\boldsymbol{T} = \sum_{n=1}^{N} r_{nk}(\boldsymbol{y}_n-\boldsymbol{\beta}_k\boldsymbol{X}_n)(\boldsymbol{y}_n-\boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}} + (\boldsymbol{\beta}_k-\boldsymbol{M}_0)\boldsymbol{K}_0(\boldsymbol{\beta}_k-\boldsymbol{M}_0)^{\mathrm{T}} + \boldsymbol{P}_0^{-1} - (\boldsymbol{\beta}_k-\boldsymbol{M}_k)\boldsymbol{K}_k(\boldsymbol{\beta}_k-\boldsymbol{M}_k)^{\mathrm{T}} \tag{4.24}$$

By matching coefficients with the term $|\boldsymbol{V}_k|^{(\eta_k-d-1)/2}$ in the Wishart distribution in equation 4.22 we get:

$$\eta_k = \eta_0 + N_k$$

By matching coefficients with the term $\exp\left\{-\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{P}_k^{-1}\boldsymbol{V}_k\right)\right\}$ in equation 4.22 we get:

$$\boldsymbol{P}_k^{-1} = \boldsymbol{T}$$

We have now derived the equations for updating the variational parameters of the Matrix-Normal-Wishart distributions $q^\star(\boldsymbol{\beta}_k, \boldsymbol{V}_k)$.

The update term for the variational parameter $\boldsymbol{P}_k^{-1}$ can be further simplified. We can get the following result as a more convenient equation for the variational parameter:

$$\begin{aligned}\boldsymbol{P}_k^{-1} &= \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_0^T + \sum_{n=1}^{N} r_{nk}\boldsymbol{y}_n\boldsymbol{y}_n^T - \boldsymbol{M}_k\boldsymbol{K}_k\boldsymbol{M}_k^T \\ &= \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_0^T + N_k\boldsymbol{Q}_k + N_k\overline{\boldsymbol{y}}_k\overline{\boldsymbol{y}}_k^T - \boldsymbol{M}_k\boldsymbol{K}_k\boldsymbol{M}_k^T\end{aligned}$$

We used the definitions:

$$\overline{\boldsymbol{y}}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{y}_n$$

$$\boldsymbol{Q}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\boldsymbol{y}_n - \overline{\boldsymbol{y}}_k)(\boldsymbol{y}_n - \overline{\boldsymbol{y}}_k)^T$$

In appendix D.2 we show in detail how to get to the more convenient update equation for $\boldsymbol{P}_k^{-1}$.
A summary of the likelihood function, the prior, the approximate posterior and the update equations for the variational parameters of the approximate posterior can be found in appendix B.

4.4.6 Variational Lower Bound on the Marginal Likelihood

For the variational lower bound on the marginal likelihood of the Dirichlet process mixture of generalized linear models we get:

$$\begin{aligned}\mathrm{L} &= \sum_{\boldsymbol{z}}\int\int\int\int\int q(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) \log\left\{\frac{p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}{q(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}\right\} \mathrm{d}\upsilon\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\boldsymbol{\Sigma}\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\boldsymbol{V} \\ &= \mathbb{E}[\log p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})] - \mathbb{E}[\log q(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})] \\ &= \mathbb{E}[\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})] + \mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{\pi})] + \mathbb{E}[\log p(\upsilon)] + \mathbb{E}[\log p(\boldsymbol{\mu}, \boldsymbol{\Sigma})] + \mathbb{E}[\log p(\boldsymbol{\beta}, \boldsymbol{V})] \\ &\quad - \mathbb{E}[\log q(\boldsymbol{z})] - \mathbb{E}[\log q(\upsilon)] - \mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})] - \mathbb{E}[\log q(\boldsymbol{\beta}, \boldsymbol{V})]\end{aligned}$$

All expectations are with respect to $q(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})$.
With exception of the expectations $\mathbb{E}[\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})]$, $\mathbb{E}[\log p(\boldsymbol{\beta}, \boldsymbol{V})]$ and $\mathbb{E}[\log q(\boldsymbol{\beta}, \boldsymbol{V})]$ all terms are the same as for the variational treatment of GMM and DP-GMM.

Using results from before, the first term is:

$$\begin{aligned}\mathbb{E}[\log p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})] &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[z_{nk}\left(\log \mathrm{N}\left(\boldsymbol{y}_n|\boldsymbol{\beta}_k\boldsymbol{X}_n, \boldsymbol{V}_k^{-1}\right) + \log \mathrm{N}\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)\right)\right] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[z_{nk}\right]\cdot\mathbb{E}\Bigg[-\frac{m}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \\ &\qquad -\frac{d}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V}_k(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)\Bigg] \\ &= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[z_{nk}\right]\cdot\Big\{-m\log 2\pi + \mathbb{E}\left[\log|\boldsymbol{\Sigma}_k|\right] - \mathbb{E}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] \\ &\qquad -d\log 2\pi + \mathbb{E}\left[\log|\boldsymbol{V}_k|\right] - \mathbb{E}\left[(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)^{\mathrm{T}}\boldsymbol{V}_k(\boldsymbol{y}_n - \boldsymbol{\beta}_k\boldsymbol{X}_n)\right]\Big\} \\ &= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\cdot\Big\{-m\log 2\pi + \log\widetilde{\boldsymbol{\Sigma}}_k - m\lambda_k^{-1} - \upsilon_k\cdot(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_k)\boldsymbol{L}_k(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_k)^T - \upsilon_k\operatorname{Tr}[\boldsymbol{S}_k\boldsymbol{L}_k] \\ &\qquad -d\log 2\pi + \log\widetilde{\boldsymbol{V}}_k - \operatorname{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} - \eta_k(\boldsymbol{y}_n - \boldsymbol{M}_k\boldsymbol{X}_n)^T\boldsymbol{P}_k(\boldsymbol{y}_n - \boldsymbol{M}_k\boldsymbol{X}_n)\Big\} \\ &= \frac{1}{2}\sum_{k=1}^{K}N_k\cdot\Big\{-m\log 2\pi + \log\widetilde{\boldsymbol{\Sigma}}_k - m\lambda_k^{-1} - \upsilon_k\cdot(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_k)\boldsymbol{L}_k(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_k)^T - \upsilon_k\operatorname{Tr}[\boldsymbol{S}_k\boldsymbol{L}_k] \\ &\qquad -d\log 2\pi + \log\widetilde{\boldsymbol{V}}_k - \operatorname{Tr}\left\{\boldsymbol{K}_k^{-1}\boldsymbol{X}_n\boldsymbol{X}_n^T\right\} - \eta_k(\boldsymbol{y}_n - \boldsymbol{M}_k\boldsymbol{X}_n)^T\boldsymbol{P}_k(\boldsymbol{y}_n - \boldsymbol{M}_k\boldsymbol{X}_n)\Big\}\end{aligned}$$

The second term is:

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{\beta}, \boldsymbol{V})] &= \sum_{k=1}^{K} \mathbb{E}\left[\log \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_0, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_0\right) + \log \mathrm{Wi}\left(\boldsymbol{V}_k | \boldsymbol{P}_0, \eta_0\right)\right] \\
&= \sum_{k=1}^{K} \mathbb{E}\Bigg[-\frac{m}{2}\log 2\pi + \frac{d}{2}\log|\boldsymbol{K}_0| + \frac{m}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\mathrm{Tr}\left[(\boldsymbol{\beta}_k - \boldsymbol{M}_0)^T \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right] \\
&\quad + \log D(\boldsymbol{P}_0, \eta_0) + \frac{\eta_0 - d - 1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\mathrm{Tr}(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k)\Bigg] \\
&= \frac{Km}{2}\log 2\pi + \frac{Kd}{2}\log|\boldsymbol{K}_0| + \frac{m}{2}\sum_{k=1}^{K}\log\widetilde{V}_k - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left[\mathrm{Tr}\left[(\boldsymbol{\beta}_k - \boldsymbol{M}_0)^T \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right]\right] \\
&\quad + K\log D(\boldsymbol{P}_0, \eta_0) + \frac{\eta_0 - d - 1}{2}\sum_{k=1}^{K}\log\widetilde{V}_k - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left[\mathrm{Tr}(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k)\right]
\end{aligned}$$

In this term there are two expectations left that we need to derive:

$$\mathbb{E}\left[\mathrm{Tr}(\boldsymbol{P}_0^{-1}\boldsymbol{V}_k)\right] = \mathrm{Tr}(\boldsymbol{P}_0^{-1}\mathbb{E}\left[\boldsymbol{V}_k\right]) = \eta_k \mathrm{Tr}(\boldsymbol{P}_0^{-1}\boldsymbol{P}_k)$$

And:

$$\begin{aligned}
\mathbb{E}\left[\mathrm{Tr}\left[(\boldsymbol{\beta}_k - \boldsymbol{M}_0)^T \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right]\right] &= \mathbb{E}\left[\mathrm{Tr}\left[\boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_0)\boldsymbol{K}_0(\boldsymbol{\beta}_k - \boldsymbol{M}_0)^T\right]\right] \\
&= \mathbb{E}\left[\mathrm{Tr}\left[\boldsymbol{V}_k\left(\boldsymbol{\beta}_k\boldsymbol{K}_0\boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k\boldsymbol{K}_0\boldsymbol{M}_0^T - \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{\beta}_k^T + \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_0^T\right)\right]\right] \\
&= \mathbb{E}_{q(V)}\left[\mathrm{Tr}\left[\boldsymbol{V}_k\left(\mathbb{E}_{q(\beta)}\left[\boldsymbol{\beta}_k\boldsymbol{K}_0\boldsymbol{\beta}_k^T\right] - \mathbb{E}_{q(\beta)}\left[\boldsymbol{\beta}_k\right]\boldsymbol{K}_0\boldsymbol{M}_0^T - \boldsymbol{M}_0\boldsymbol{K}_0\mathbb{E}_{q(\beta)}\left[\boldsymbol{\beta}_k^T\right] + \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_0^T\right)\right]\right] \\
&= \mathbb{E}_{q(V)}\left[\mathrm{Tr}\left[\boldsymbol{V}_k\left(\boldsymbol{M}_k\boldsymbol{K}_0\boldsymbol{M}_k^T + \mathrm{Tr}\left[\boldsymbol{K}_k^{-1}\boldsymbol{K}_0\right]\boldsymbol{V}_k^{-1} - \boldsymbol{M}_k\boldsymbol{K}_0\boldsymbol{M}_0^T - \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_k^T + \boldsymbol{M}_0\boldsymbol{K}_0\boldsymbol{M}_0^T\right)\right]\right] \\
&= \mathrm{Tr}\left[\mathbb{E}_{q(V)}\left[\boldsymbol{V}_k\right](\boldsymbol{M}_k - \boldsymbol{M}_0)\boldsymbol{K}_0(\boldsymbol{M}_k - \boldsymbol{M}_0)^T\right] + \mathrm{Tr}\left[\boldsymbol{K}_k^{-1}\boldsymbol{K}_0\right] \\
&= \eta_k \mathrm{Tr}\left[\boldsymbol{P}_k(\boldsymbol{M}_k - \boldsymbol{M}_0)\boldsymbol{K}_0(\boldsymbol{M}_k - \boldsymbol{M}_0)^T\right] + \mathrm{Tr}\left[\boldsymbol{K}_k^{-1}\boldsymbol{K}_0\right] \\
&= \eta_k \mathrm{Tr}\left[(\boldsymbol{M}_k - \boldsymbol{M}_0)^T\boldsymbol{P}_k(\boldsymbol{M}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right] + \mathrm{Tr}\left[\boldsymbol{K}_k^{-1}\boldsymbol{K}_0\right]
\end{aligned}$$

In summary the second term is:

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{\beta}, \boldsymbol{V})] &= \frac{Km}{2}\log 2\pi + \frac{Kd}{2}\log|\boldsymbol{K}_0| + \frac{m}{2}\sum_{k=1}^{K}\log\widetilde{V}_k - \frac{1}{2}\sum_{k=1}^{K}\eta_k \mathrm{Tr}\left[(\boldsymbol{M}_k - \boldsymbol{M}_0)^T\boldsymbol{P}_k(\boldsymbol{M}_k - \boldsymbol{M}_0)\boldsymbol{K}_0\right] + \mathrm{Tr}\left[\boldsymbol{K}_k^{-1}\boldsymbol{K}_0\right] \\
&\quad + K\log D(\boldsymbol{P}_0, \eta_0) + \frac{\eta_0 - d - 1}{2}\sum_{k=1}^{K}\log\widetilde{V}_k - \frac{1}{2}\sum_{k=1}^{K}\eta_k \mathrm{Tr}(\boldsymbol{P}_0^{-1}\boldsymbol{P}_k)
\end{aligned}$$

The third term is:

$$\begin{aligned}
\mathbb{E}[\log q(\boldsymbol{\beta}, \boldsymbol{V})] &= \sum_{k=1}^{K} \mathbb{E}\left[\log \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_k, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_k\right) + \log \mathrm{Wi}\left(\boldsymbol{V}_k | \boldsymbol{P}_k, \eta_k\right)\right] \\
&= \sum_{k=1}^{K} \mathbb{E}\Bigg[-\frac{m}{2}\log 2\pi + \frac{d}{2}\log|\boldsymbol{K}_k| + \frac{m}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\mathrm{Tr}\left[(\boldsymbol{\beta}_k - \boldsymbol{M}_k)^T \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k)\boldsymbol{K}_k\right] \\
&\quad + \log D(\boldsymbol{P}_k, \eta_k) + \frac{\eta_k - d - 1}{2}\log|\boldsymbol{V}_k| - \frac{1}{2}\mathrm{Tr}(\boldsymbol{P}_k^{-1}\boldsymbol{V}_k)\Bigg] \\
&= \frac{Km}{2}\log 2\pi + \frac{Kd}{2}\log|\boldsymbol{K}_k| + \frac{m}{2}\sum_{k=1}^{K}\log\widetilde{V}_k - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left[\mathrm{Tr}\left[(\boldsymbol{\beta}_k - \boldsymbol{M}_k)^T \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k)\boldsymbol{K}_k\right]\right] \\
&\quad + K\log D(\boldsymbol{P}_k, \eta_k) + \frac{\eta_k - d - 1}{2}\sum_{k=1}^{K}\log\widetilde{V}_k - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left[\mathrm{Tr}(\boldsymbol{P}_k^{-1}\boldsymbol{V}_k)\right]
\end{aligned}$$

In this term there are two expectations left that we need to derive:

$$\mathbb{E}\left[\mathrm{Tr}(\boldsymbol{P}_k^{-1}\boldsymbol{V}_k)\right] = \mathrm{Tr}(\boldsymbol{P}_k^{-1}\mathbb{E}\left[\boldsymbol{V}_k\right]) = \eta_k \mathrm{Tr}(\boldsymbol{P}_k^{-1}\boldsymbol{P}_k) = \eta_k d$$

And:

$$\begin{aligned}
\mathbb{E}\left[\mathrm{Tr}\left[(\boldsymbol{\beta}_k - \boldsymbol{M}_k)^T \boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k)\boldsymbol{K}_k\right]\right] &= \mathbb{E}\left[\mathrm{Tr}\left[\boldsymbol{V}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k)\boldsymbol{K}_k (\boldsymbol{\beta}_k - \boldsymbol{M}_k)^T\right]\right] \\
&= \mathbb{E}\left[\mathrm{Tr}\left[\boldsymbol{V}_k \left(\boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{M}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T + \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T\right)\right]\right] \\
&= \mathbb{E}_{q(\boldsymbol{V})}\left[\mathrm{Tr}\left[\boldsymbol{V}_k \left(\mathbb{E}_{q(\boldsymbol{\beta})}\left[\boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T\right] - \mathbb{E}_{q(\beta)}\left[\boldsymbol{\beta}_k\right] \boldsymbol{K}_k \boldsymbol{M}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \mathbb{E}_{q(\beta)}\left[\boldsymbol{\beta}_k^T\right] + \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T\right)\right]\right] \\
&= \mathbb{E}_{q(\boldsymbol{V})}\left[\mathrm{Tr}\left[\boldsymbol{V}_k \left(\boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T + \mathrm{Tr}\left[\boldsymbol{K}_k^{-1} \boldsymbol{K}_k\right] \boldsymbol{V}_k^{-1} - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T + \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T\right)\right]\right] \\
&= m
\end{aligned}$$

In summary the third term is:

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{\beta}, \boldsymbol{V})] = \frac{Km}{2} \log 2\pi + \frac{Kd}{2} \log |\boldsymbol{K}_k| + \frac{m}{2} \sum_{k=1}^{K} \log \widetilde{V}_k - \frac{Km}{2} \\
+ K \log D(\boldsymbol{P}_k, \eta_k) + \frac{\eta_k - d - 1}{2} \sum_{k=1}^{K} \log \widetilde{V}_k - \frac{1}{2} \sum_{k=1}^{K} \eta_k d
\end{aligned}$$

A summary of the algorithm for the variational Bayes EM algorithm for DP-GLM can be found in appendix B. We now have the tools for obtaining an approximate posterior. In a regression setting usually the prediction on new inputs is the aim. Therefore we need to use the approximate posterior to calculate a predictive distribution. This is the focus of the next section.

4.5 Posterior Predictive Distribution of Dirichlet Process Mixtures of Generalized Linear Models

For making predictions with DP-GLM we need to derive the posterior predictive distribution. We will make use of the approximate posterior here.
One can obtain the predictive distribution by marginalizing the likelihood of the prediction $\hat{\boldsymbol{y}}$ given the test input $\hat{\boldsymbol{x}}$ and the parameters and the likelihood of the test input $\hat{\boldsymbol{x}}$ over the posterior of the parameters given the training data $\boldsymbol{x}, \boldsymbol{y}$. Associated with $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$ is a latent cluster assignment variable $\hat{\boldsymbol{z}}$. $\hat{\boldsymbol{X}}$ is defined as $\hat{\boldsymbol{X}} = [\,1 \quad \hat{\boldsymbol{x}}\,]^T$. We follow the derivation of the posterior predictive distribution of a variational mixture of Gaussians in [7] and extend it to DP-GLM.
The posterior predictive distribution is as follows:

$$\begin{aligned} p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) &= \sum_{\hat{\boldsymbol{z}}} \int\int\int\int\int p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \boldsymbol{\beta}, \boldsymbol{V})\, p(\hat{\boldsymbol{x}} \mid \hat{\boldsymbol{z}}, \boldsymbol{\mu}, \boldsymbol{\Sigma})\, p(\hat{\boldsymbol{z}} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V} \mid \boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{\pi}\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\boldsymbol{\Sigma}\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\boldsymbol{V} \\ &= \sum_{\hat{\boldsymbol{z}}} \int\int\int\int\int \prod_{k=1}^{K} \mathrm{N}\left(\hat{\boldsymbol{y}} \mid \boldsymbol{\beta}_k \hat{\boldsymbol{X}}, \boldsymbol{V}_k^{-1}\right)^{\hat{\boldsymbol{z}}_k} \mathrm{N}\left(\hat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{\hat{\boldsymbol{z}}_k} \prod_{k=1}^{K} \boldsymbol{\pi}_k^{\hat{\boldsymbol{z}}_k}\, p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V} \mid \boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{\pi}\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\boldsymbol{\Sigma}\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\boldsymbol{V} \end{aligned}$$

We perform the summation over $\hat{\boldsymbol{z}}$ to get:

$$p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) = \sum_{k=1}^{K} \int\int\int\int\int \boldsymbol{\pi}_k\, \mathrm{N}\left(\hat{\boldsymbol{y}} \mid \boldsymbol{\beta}_k \hat{\boldsymbol{X}}, \boldsymbol{V}_k^{-1}\right) \mathrm{N}\left(\hat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right) p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V} \mid \boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{\pi}\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\boldsymbol{\Sigma}\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\boldsymbol{V} \tag{4.25}$$

As the true posterior $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V} \mid \boldsymbol{x}, \boldsymbol{y})$ is not tractable we replace it by the approximate posterior. The approximate posterior factorizes to:

$$q(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) = \prod_{j=1}^{K-1} q(\nu_j) \prod_{j=1}^{K} q\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) \prod_{j=1}^{K} q\left(\boldsymbol{\beta}_j, \boldsymbol{V}_j\right)$$

By additionally using the stick-breaking definition of the mixture weights $\pi_k(\nu) = \nu_k \prod_{j=1}^{k-1}\left(1-\nu_j\right)$ we get:

$$\begin{aligned} p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) \approx \sum_{k=1}^{K} \int\int\int\int\int \nu_k \prod_{j=1}^{k-1}\left(1-\nu_j\right) \mathrm{N}\left(\hat{\boldsymbol{y}} \mid \boldsymbol{\beta}_k \hat{\boldsymbol{X}}, \boldsymbol{V}_k^{-1}\right) \mathrm{N}\left(\hat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right) \prod_{j=1}^{K-1} q(\nu_j) \prod_{j=1}^{K} q\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) \\ \prod_{j=1}^{K} q\left(\boldsymbol{\beta}_j, \boldsymbol{V}_j\right)\, \mathrm{d}\nu\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\boldsymbol{\Sigma}\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\boldsymbol{V} \end{aligned}$$

For all indices $j \neq k$ the integration with respect to $\boldsymbol{\mu}_j$,$\boldsymbol{\Sigma}_j$, $\boldsymbol{\beta}_k$ and $\boldsymbol{V}$ will sum to 1:

$$\begin{aligned} p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) \approx \sum_{k=1}^{K} \int\int\int\int\int \nu_k \prod_{j=1}^{k-1}\left(1-\nu_j\right) \prod_{j=1}^{K-1} q(\nu_j)\, \mathrm{N}\left(\hat{\boldsymbol{y}} \mid \boldsymbol{\beta}_k \hat{\boldsymbol{X}}, \boldsymbol{V}_k^{-1}\right) \mathrm{N}\left(\hat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right) \\ q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\, q(\boldsymbol{\beta}_k, \boldsymbol{V}_k)\, \mathrm{d}\nu\, \mathrm{d}\boldsymbol{\mu}_k\, \mathrm{d}\boldsymbol{\Sigma}_k\, \mathrm{d}\boldsymbol{\beta}_k\, \mathrm{d}\boldsymbol{V}_k \end{aligned} \tag{4.26}$$

We see that we need to calculate the expectation of the term $\nu_k \prod_{j=1}^{k-1}\left(1-\nu_j\right)$ under the Beta distributions $\prod_{j=1}^{K-1} q\left(\nu_j \mid \gamma_{1,j}, \gamma_{2,j}\right)$. The expectation of a Beta distribution is $\mathbb{E}(\nu_j) = \gamma_{1,j}/(\gamma_{1,j}+\gamma_{2,j})$ (see B.7 in [7]).
Using this fact and writing out the approximate posterior we get:

$$\begin{aligned} p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) &\approx \sum_{k=1}^{K} \frac{\gamma_{1,k}}{\gamma_{1,k}+\gamma_{2,k}} \prod_{j=1}^{K-1}\left(1 - \frac{\gamma_{1,j}}{\gamma_{1,j}+\gamma_{2,j}}\right) \iint \mathrm{N}\left(\hat{\boldsymbol{y}} \mid \boldsymbol{\beta}\hat{\boldsymbol{X}}, \boldsymbol{V}_k^{-1}\right) \mathrm{N}\left(\boldsymbol{\beta}_k \mid \boldsymbol{M}_k, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_k\right) \mathrm{Wi}(\boldsymbol{V}_k \mid \boldsymbol{L}_k, \nu_k) \\ &\quad \iint \mathrm{N}\left(\hat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right) \mathrm{N}\left(\boldsymbol{\mu}_k \mid \boldsymbol{m}_k, (\lambda_k \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k \mid \boldsymbol{P}_k, \eta_k)\, \mathrm{d}\boldsymbol{\mu}_k\, \mathrm{d}\boldsymbol{\Sigma}_k\, \mathrm{d}\boldsymbol{\beta}_k\, \mathrm{d}\boldsymbol{V}_k \\ &\propto \sum_{k=1}^{K} \frac{\gamma_{1,k}}{\gamma_{1,k}+\gamma_{2,k}} \prod_{j=1}^{K-1}\left(1 - \frac{\gamma_{1,j}}{\gamma_{1,j}+\gamma_{2,j}}\right) \cdot p_k(\hat{\boldsymbol{x}}) \cdot \mathrm{T}\left(\boldsymbol{M}_k \hat{\boldsymbol{x}}, \boldsymbol{L}_k^{-1} c^{-1}, \nu_k + 1\right) \end{aligned}$$

In the last step we inspected the integral over the likelihood of $\hat{\boldsymbol{y}}$ times the Matrix-Normal-Wishart posterior of the regression parameters $\boldsymbol{\beta}_k$ and the precision matrix $\boldsymbol{V}_k$ and recognized it as the predictive distribution of a Bayesian

linear regression model. The predictive distribution of a Bayesian linear regression model is a matrix T-distribution (see equation 2.10). In contrast to equation 2.10 we use the inverse of L_k. This is due to the reason that we use the Wishart distribution for our derivations.
The constant c is:

$$c = 1 - \hat{\boldsymbol{x}}^T \left(\boldsymbol{K}_k + \hat{\boldsymbol{x}} \hat{\boldsymbol{x}}^T \right)^{-1} \hat{\boldsymbol{x}}$$

We also made use of the fact that the integral over the likelihood of a given input data point $\hat{\boldsymbol{x}}$ times the Normal-Wishart posterior of the cluster means $\boldsymbol{\mu}_k$ and the precision matrix $\boldsymbol{\Sigma}_k$ is the evidence $p_k(\hat{\boldsymbol{x}})$ of the input $\hat{\boldsymbol{x}}$ for cluster k. The evidence is also called marginal likelihood. According to equations (233) and (234) in [67] the evidence can be calculated as:

$$\begin{aligned} p_k(\hat{\boldsymbol{x}}) &= \frac{Z_{\hat{x}}}{Z_k} \frac{1}{(2\pi)^{nm/2}} \\ &= \frac{1}{\pi^{nm/2}} \frac{\Gamma_m(\eta_{\hat{x}}/2)}{\Gamma_m(\eta_k/2)} \frac{|P_k|^{\eta_k/2}}{|P_{\hat{x}}|^{\eta_{\hat{x}}/2}} \left(\frac{\lambda_k}{\lambda_{\hat{x}}} \right)^{m/2} \end{aligned}$$

The parameters with index $\hat{x}$ indicate the parameters of the Normal-Wishart posterior, that have been updated by the new input $\hat{\boldsymbol{x}}$. This means that we take the posterior parameters with index k as a new prior and apply the mean field update equations with input data $\hat{\boldsymbol{x}}$ to obtain new posterior parameters with index $\hat{x}$. As we only use one data point for calculating the evidence we can set $n = 1$.
The first line shows an alternative way of calculating the evidence as a ratio of the normalization constants of the Normal-Wishart posterior Z_k and the updated Normal-Wishart posterior $Z_{\hat{x}}$.

To obtain a proper probability distribution, the approximate posterior predictive distribution still needs to be normalized. We use the following abbreviation:

$$\epsilon_k = \frac{\gamma_{1,k}}{\gamma_{1,k} + \gamma_{2,k}} \prod_{j=1}^{K-1} \left(1 - \frac{\gamma_{1,j}}{\gamma_{1,j} + \gamma_{2,j}} \right)$$

We get as a final result:

$$p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) = \frac{1}{\sum_{k=1}^{K} \epsilon_k \cdot p_k(\hat{\boldsymbol{x}})} \sum_{k=1}^{K} \epsilon_k \cdot p_k(\hat{\boldsymbol{x}}) \cdot \mathrm{T}\left(\boldsymbol{M}_k \hat{\boldsymbol{x}}, \boldsymbol{L}_k^{-1} c^{-1}, \nu_k + 1 \right)$$

The posterior predictive distribution of DP-GLM is a mixture of matrix T-distributions. It can be interpreted as follows. The prediction for $\hat{\boldsymbol{y}}$ is a weighted sum of the predictions of the overall K linear models that have been used for fitting the training data. The prediction of $\hat{\boldsymbol{y}}_k$ for a cluster k is drawn from a matrix T-distribution with mean $\boldsymbol{M}_k \hat{\boldsymbol{x}}$. The weights are represented by the term $\epsilon_k \cdot p_k(\hat{\boldsymbol{x}})$. ϵ_k is large, if many data points were assigned to the cluster k in training. The evidence $p_k(\hat{\boldsymbol{x}})$ of is large, if it is probable that the new input $\hat{\boldsymbol{x}}$ was generated by cluster k.

Replacing the Stick-Breaking Prior with a Dirichlet Prior

In chapter 6 we want to compare the infinite mixture approach of DP-GLM to the finite mixture case, that uses a Dirichlet distribution instead of the stick-breaking construction as a prior. In equation 4.25 we do not replace π_k by the stick-breaking construction, but we use a Dirichlet distribution as an approximate posterior of the mixture weights. The $k = 1, \ldots, K$ mixture weights π_k are distributed according to a Dirichlet distribution $q(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^{\mathrm{T}}$.
The approximate posterior now factorizes to:

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) = q(\boldsymbol{\pi}) \prod_{j=1}^{K} q\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j \right) \prod_{j=1}^{K} q\left(\boldsymbol{\beta}_j, \boldsymbol{V}_j \right)$$

After some analogous intermediate steps we get the following result, which is similar to equation 4.26:

$$p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) \approx \sum_{k=1}^{K} \int\int\int\int\int \pi_k q(\boldsymbol{\pi}) \mathrm{N}\left(\hat{\boldsymbol{y}} \mid \boldsymbol{\beta}_k \hat{\boldsymbol{X}}, \boldsymbol{V}_k^{-1} \right) \mathrm{N}\left(\hat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1} \right) q\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right) q\left(\boldsymbol{\beta}_k, \boldsymbol{V}_k \right) \mathrm{d}\boldsymbol{\pi}\, \mathrm{d}\boldsymbol{\mu}_k\, \mathrm{d}\boldsymbol{\Sigma}_k\, \mathrm{d}\boldsymbol{\beta}_k\, \mathrm{d}\boldsymbol{V}_k$$

We see that we need to calculate the expectation of π_k under the Dirichlet distribution $p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$. This expectation is $\mathbb{E}(\pi_k) = \alpha_k / \sum_{j=1}^{K} \alpha_j$ (see B.17 in [7]).
The intermediate steps are then again the same as in the case of using the DP. We get for the approximate predictive posterior for the DP-GLM, where the DP is replaced with a Dirichlet distribution as a prior:

$$p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) \propto \sum_{k=1}^{K} \frac{\alpha_k}{\sum_{j=1}^{K} \alpha_j} \cdot p_k(\hat{\boldsymbol{x}}) \cdot \mathrm{T}\left(\boldsymbol{M}_k \hat{\boldsymbol{x}}, \boldsymbol{L}_k^{-1} c^{-1}, \upsilon_k + 1\right)$$

In the normalized form this is:

$$p(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{x}) = \frac{1}{\sum_{k=1}^{K} \alpha_k \cdot p_k(\hat{\boldsymbol{x}})} \sum_{k=1}^{K} \alpha_k \cdot p_k(\hat{\boldsymbol{x}}) \cdot \mathrm{T}\left(\boldsymbol{M}_k \hat{\boldsymbol{x}}, \boldsymbol{L}_k^{-1} c^{-1}, \upsilon_k + 1\right)$$

The interpretation is analogous to the predictive distribution, where the DP is used as a prior.

5 Gibbs Sampling

This chapter is about the Bayesian inference method of Gibbs sampling for obtaining samples from the true posterior of DP-GLM. In contrast to variational inference, which is a deterministic approach, Gibbs sampling is a stochastic sampling method. We first give a general overview of the method in the context of the Markov chain Monte Carlo framework and then show how to apply Gibbs sampling to DP-GLM.
In the experiments in chapter 6 we use Gibbs sampling as an initialization for the variational Bayes EM algorithm. This combines the strengths of both algorithms. The mean field VI algorithm can get stuck in bad local optima. An initialization with Gibbs sampling can overcome this weakness. On the other hand, the convergence of the Gibbs sampling algorithm is hard to assess. It thus is sensible to combine Gibbs sampling with mean field VI. The variational lower bound of VI can be used for judging the convergence. The VLB increases with every iteration of the variational Bayes EM algorithm and thus provides a criterion for stopping the algorithm. The samples that are obtained by Gibbs sampling can be from different modes of the posterior, which lowers the risk of variational Bayes EM to repeatedly getting stuck in bad local optima.

5.1 Gibbs Sampling as a Markov Chain Monte Carlo Method

Monte Carlo techniques are approximate inference methods, that are based on numerical sampling. The problem that we want to address is to find the expectation of a function $f(\boldsymbol{x})$ with respect to a probability distribution $p(\boldsymbol{x})$. In the case of continuous variables $\boldsymbol{x}$ the expectation is: [7]

$$\mathbb{E}[f] = \int f(\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

If the expectation is too complex to be evaluated analytically, one can make use of sampling methods. First a set of samples $\mathbf{x}^{(l)}$ with $l = 1, \ldots, L$ have to be obtained. These samples are drawn independently from $p(\mathbf{x})$. The expectation can then be approximated by a finite sum:

$$\widehat{f} = \frac{1}{L} \sum_{l=1}^{L} f\left(\mathbf{x}^{(l)}\right)$$

If the samples are drawn from true distribution $p(\boldsymbol{x})$, the estimator $\hat{f}$ has the correct mean. The variance of the estimator is:

$$\mathrm{var}[\widehat{f}] = \frac{1}{L} \mathbb{E}\left[(f - \mathbb{E}[f])^2\right]$$

The accuracy of the estimator does not depend on the dimensionality of $\boldsymbol{x}$ and in principle high accuracy can be achieved for a relatively small amount of samples. A frequent problem in that subsequent samples are not drawn independently. In these cases many samples might need to be taken, to estimate the mean with sufficient accuracy.

We now turn to the framework of *Markov chain Monte Carlo* (MCMC). In MCMC samples are drawn from a proposal distribution $q\left(\mathbf{x} \,|\, \mathbf{x}^{(\tau)}\right)$. The prosal distribution depends on the current state $\mathbf{x}^{(\tau)}$, so that a record of the current state has to be maintained. The proposal distribution needs to be sufficiently simple to draw samples from it. The sequence of samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ forms a *Markov chain*. A first order Markov chain is defined as a series of random variables, such that the following conditional independence property holds: [7]

$$p\left(\mathbf{x}^{(m+1)} | \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right) = p\left(\mathbf{x}^{(m+1)} | \mathbf{x}^{(m)}\right)$$

This means that each state is only dependent on the directly preceding state.

At each cycle of an MCMC algorithm, a candidate sample $\boldsymbol{x}^{\star}$ is drawn from the proposal distribution. The sample is accepted or rejected, depending uppon an appropriate criterion. In the case of Gibbs sampling the samples are always accepted.

Gibbs sampling [68] is a special instance of a MCMC algorithm called *Metropolis-Hastings algorithm* [69]. In Gibbs sampling we have a joint distribution $p(\mathbf{x}) = p(x_1, \ldots, x_M)$ that we want to sample from. In the case of DP-GLM this is the joint distribution of the hidden variables. First an initial state of the Markov chain has to be chosen, i.e. by an intialization of the random variables. Then in each iteration of the algorithm the value of one of the variables needs to be replaced by a value drawn from the distribution of that variable conditioned on the values of the remaining variables. This procedure is repeated by cycling through the conditional distributions in some particular order. We use the example of a joint distribution of three random variables $p(x_1, x_2, x_3)$. The Gibbs sampling procedure is displayed in algortihm 2. [7]
After a *burn-in* period this procedure produces samples from the true joint distribution $p(\boldsymbol{x})$. The advantage of this ap-

Input: Initial state of the Markov chain: x_1, x_2, x_3

1 **for** τ in T **do**
2 Sample $x_1^{(\tau+1)} \sim p\left(x_1 | x_2^{(\tau)}, x_3^{(\tau)}\right)$;
3 Sample $x_2^{(\tau+1)} \sim p\left(x_2 | x_1^{(\tau)}, x_3^{(\tau)}\right)$;
4 Sample $x_3^{(\tau+1)} \sim p\left(x3 | x_1^{(\tau)}, x_2^{(\tau)}\right)$;

Algorithm 2: Gibbs sampling algorithm for obtaining samples from a joint distribution of three random variables.

proach is, that no samples need to be drawn from $p(\boldsymbol{x})$ directly, as that might be infeasible. Sampling from the conditional distributions is sufficient to converge to samples from $p(\boldsymbol{x})$. We refer to [7] for an explanation why the algorithm creates samples from the required distribution. Successive samples from the Markov chain are highly correlated, so that the sequence must be sub-sampled to generate nearly independent samples.

5.2 Gibbs Sampling for Dirichlet Process Mixtures of Generalized Linear Models

We begin our analysis of the Gibbs sampling algorithm by showing the connection of the finite and infinite mixture models to the Chinese restaurant process representation of the DP as described in [30, 11, 10].
In a finite mixture model one can write the conditional prior distribution of an indicator variable, that assigns a datum to a component, as follows: [10, 30]

$$p(z_i = k | \mathbf{z}_{-i}, \alpha_0) = \frac{n_{k,-i} + \alpha_0/K}{N + \alpha_0 - 1} \tag{5.1}$$

$\mathbf{z}_{-i}$ are the indicator variables of the other data points in cluster k excluding data point x_i. α_0 is the hyperparameter of the Dirichlet prior. K is the number of clusters and N is the number of data points. $n_{k,-i}$ is the number of data points in cluster k exluding x_i. For the derivation of the formula see [11, 30].

If we let K go to infinity, the conditional prior distributions of the indicator variables reach the following limits:

$$\begin{aligned} P(z_i = k | \mathbf{z}_{-i}, \alpha_0) &= \frac{n_{k,-i}}{N + \alpha_0 - 1} \\ P\left(z_i \neq z_j \text{ for all } j \neq i | \mathbf{z}_{-i}, \alpha_0\right) &= \frac{\alpha_0}{\alpha_0 + N - 1} \end{aligned} \tag{5.2}$$

As the ordering of the assignment variables z_i does not matter in infinite mixture models, we can think of z_i being the last one in the order. Equation 5.2 then resembles the results of the Chinese restaurant process (equation 2.13). We need these conditional probabilities for the Gibbs sampling algorithm of DP-GLM. We develop the algorithm by starting with GMM and DP-GMM.

In a Gibbs sampling algorithm for finite mixture models [8, 11], the data points $\mathbf{x} = \{x_i\}_{i=1}^n$ are observed and $\mathbf{z} = \{z_i\}_{i=1}^n$ are latent cluster indicator variables. The Gibbs sampling algorithm draws from the conditional distributions of the random variables given the other random variables. It iteratively draws from the distributions of the

- Cluster indicators $\mathbf{z} = \{z_i\}_{i=1}^n$,
- and the cluster parameters $\{\theta_k\}_{k=1}^K$.

The mixture weights can be integrated out and do not have to be updated. For implementing the Gibbs sampling algorithm, the conditional distributions of the remaining variables have to be derived. The hyperparameters of the Dirichlet distribution α_0 and of the hyperparameters λ of the observational distribution F are assumed as known. The conditional posterior for each indicator variables then is: [10]

$$p\left(z_i = k|\mathbf{z}_{-i}, \mathbf{x}, \{\theta_k\}_{k=1}^K, \alpha_0, \lambda\right) = \frac{n_{k,-i} + \alpha_0/K}{N + \alpha_0 - 1} F\left(x_i|\theta_k\right)$$

F here is the distribution of the observations. In the case of a GMM these are Gaussians with cluster means and cluster covariances.

The conditional posterior of the parameters of the k^{th} cluster, θ_k, only depends on the observations of that cluster $\boldsymbol{x}_k$: [10]

$$p\left(\theta_k|\boldsymbol{\theta}_{-k}, , \mathbf{z}, \mathbf{x}, \alpha_0, \lambda\right) \propto G_0\left(\theta_k|\lambda\right) \mathrm{L}(\mathbf{x}_k|\theta_k)$$

G_0 is the prior on the cluster parameters, which is usually chosen to be conjugate. The conjugate prior to a Normal distribution with unknown variance in a univariate setting is the Normal-inverse-Gamma distribution. L is the likelihood of the data $\boldsymbol{x}_k$ that is assigned to cluster k. The stated conditional distributions are sufficient for implementing a Gibbs sampling algorithm for finite mixture models as in algorithm 2 of [10].

The Gibbs sampling algorithm for GMM can be generalized to DP-GMM by taking the limit of infinity for the number of clusters K [10, 30, 9]. By doing so, the conditional prior of z_i changes from equation 5.1 to equation 5.2. If z_i is assigned to one of the existing clusters K, the conditioned posterior of z_i thus becomes:

$$p\left(z_i = k|\mathbf{z}_{-i}, \mathbf{x}, \{\theta_k\}_{k=1}^K, \alpha_0, \lambda\right) = \frac{n_k, -i}{N + \alpha_0 - 1} F\left(x_i|\theta_k\right)$$

If z_i is assigned to a new cluster index, that is denoted as $K+1$, the conditional posterior for this case is:

$$p\left(z_i = K+1|\mathbf{z}_{-i}, \mathbf{x}, \alpha_0, \lambda\right) = \frac{\alpha_0}{N + \alpha_0 - 1} \int F\left(x_i|\theta\right) G_0(\theta|\lambda) d\theta$$

If z_i is assigned to a news cluster index we need to sample cluster parameters for the newly created cluster. The new cluster parameters θ_{K+1} can be drawn from the posterior distribution based on the prior G_0 and the likelihood of the single observation x_i. This posterior is:

$$H\left(\theta_{K+1}|x_i\right) = \frac{G_0\left(\theta_{K+1}\right) F\left(x_i|\theta_{K+1}\right)}{\int_\theta G_0(\theta) F\left(x_i|\theta\right)}$$

Furthermore the number of clusters K must be increased by one.
With these results it is possible to implement a Gibbs sampling algorithm for Dirichlet process Gaussian mixture models as algorithm 4 of [10].

If we want to extend infinite mixtures to the regression case, we need to consider the set of parameters of the GLM, which we assume to be linear models. The Gibbs sampling algorithm iteratively samples from the conditional distributions of the

- Cluster indicators $\mathbf{z} = \{z_i\}_{i=1}^n$
- cluster parameters $\left\{\theta_{x,k}\right\}_{k=1}^K$,
- and the GLM parameters $\left\{\theta_{y,k}\right\}_{k=1}^K$.

y_i are the targets that are associated with the inputs x_i. We also have a composite observational distribution F and a composite prior G_0 of the form:

$$F = F_x\left(x_i|\theta_x\right) F_y\left(y_i|x_i, \theta_y\right)$$
$$G_0 = G_x\left(\theta_x\right) G_y\left(\theta_y\right)$$

We now have all ingredients for the Gibbs sampling algorithm of DP-GLM which is shown in algorithm 3. $\boldsymbol{x}_k$ indicates all input data that is assigned to a cluster and $\boldsymbol{y}_k$ are all targets that are assigned to a cluster. L denotes the likelihood.

In our implementation that is used for the evaluation in chapter 6 we have used a conjugate Normal-inverse-Wishart prior on the cluster means and covariance matrices. For the linear models, we have used a Matrix-Normal-inverse-Wishart prior on the mean of the regression coefficients and the covariance matrices.
Other conjugate priors are possible. For one-dimensional targets, one can choose a Normal-inverse-Gamma prior for the clusters and a multivariate Normal-inverse-Gamma for the linear models. In this case, the detailed derivations of the conditional posteriors are shown in appendix D.3.

Input : Data pairs $D = \{x_i, y_i\}_{n=1}^{N}$, initial state $\left\{\theta_{x,k}\right\}_{k=1}^{K}$, $\left\{\theta_{y,k}\right\}_{k=1}^{K}$ and $\{z_i\}_{n=1}^{N}$, prior hyperparameters, initial number of clusters K, number of iterations T, convergence criterion

for *t in number of iterations T* **do**

  for *i in number of data pairs N* **do**

    Remove data pair $\{x_i, y_i\}$ from cluster z_i;

    if *cluster* z_i *is empty* **then**

      Remove cluster z_i and its' parameters θ_{x,z_i} and θ_{y,z_i};

      Decrease K by 1;

    Sample a new z_i from the cond. posterior of cluster assignments:

$$p\left(z_i = k, k \leq K\right) \propto \frac{n_{k,-i}}{N + \alpha_0 - 1} F_x\left(x_i|\theta_{x,k}^{(t-1)}\right) F_y\left(y_i|x_i, \theta_{y,k}^{(t-1)}\right) \qquad n_{k,-i} = \sum_{j \neq i} \delta\left(z_j - k\right)$$

$$p\left(z_i = K+1\right) \propto \frac{\alpha_0}{N + \alpha_0 - 1} \int_{\theta_x} \int_{\theta_y} F_x\left(x_i|\theta_x\right) F_y\left(y_i|x_i, \theta_y\right) G_x\left(\theta_x\right) G_y\left(\theta_y\right) d\theta_x d\theta_y$$

    if $z_i = K + 1$ **then**

      Sample $\theta_{x,K+1}$ and $\theta_{y,K+1}$ for new cluster z_i from the cond. posteriors of cluster and GLM parameters:

$$H\left(\theta_{x,K+1}|x_i\right) = \frac{G_x\left(\theta_{x,K+1}\right) F_x\left(x_i|\theta_{x,K+1}\right)}{\int_{\theta_x} G_x(\theta_x) F_x\left(x_i|\theta_x\right)} \qquad H\left(\theta_{y,K+1}|y_{K+1}\right) = \frac{G_y\left(\theta_{y,K+1}\right) F_y\left(y_i|x_i, \theta_{y,K+1}\right)}{\int_{\theta_y} G_y(\theta_y) F_y\left(y_i|x_i, \theta_y\right)}$$

      Increase K by 1;

  for *k in number of clusters K* **do**

    Resample θ_k for all clusters k from the cond. posteriors of cluster and GLM parameters:

$$H\left(\theta_{x,k}|\boldsymbol{x}_k^{(t)}\right) = \frac{G_x\left(\theta_{x,k}\right) \mathrm{L}_x\left(\boldsymbol{x}_k^{(t)}|\theta_{x,k}^{(t-1)}\right)}{\int_{\theta_x} G_x(\theta_x) \mathrm{L}_x\left(\boldsymbol{x}_k^{(t)}|\theta_x\right)} \qquad H\left(\theta_{y,k}|\boldsymbol{y}_k^{(t)}\right) = \frac{G_y\left(\theta_{y,k}\right) \mathrm{L}_y\left(\boldsymbol{y}_k^{(t)}|\boldsymbol{x}_k^{(t)}, \theta_{y,k}^{(t-1)}\right)}{\int_{\theta_y} G_y(\theta_y) \mathrm{L}_y\left(\boldsymbol{y}_k^{(t)}|\boldsymbol{x}_k^{(t)}, \theta_y\right)}$$

  if *convergence criterion is met* **then**

    Record state $\{z_1, \ldots, z_N\}$, $\left\{\theta_{x,1}, \ldots, \theta_{x,K}\right\}$ and $\left\{\theta_{y,1}, \ldots, \theta_{y,K}\right\}$;

Algorithm 3: Gibbs Sampling for DP-GLM.

6 Experiments

In this chapter, we use a Python implementation of DP-GLM for predicting on test datasets. We apply the Bayesian inference methods of variational inference and Gibbs sampling to DP-GLM for learning the posterior. We evaluate the method on several different datasets. We use simulated sine data, heteroscedastic cosmic microwave background data [12, 5], simulated forward kinematics data from a robot arm with one joint, simulated data from a robot arm with three joints, as well as inverse kinematics data from a real SARCOS robot arm [13, 15, 70].

6.1 Experimental Setup

For learning the parameters of DP-GLM we use Gibbs sampling (chapter 5) in combination with a variational Bayes EM algorithm (section 4.4). The Gibbs sampling algorithm generates samples from different modes of the posterior. The variational Bayes EM algorithm locks onto the respective mode. We start the algorithm by running 1000 iterations of the Gibbs sampling algorithm. The VBEM algorithm then is initialized with the posterior parameters of the last iteration of Gibbs sampling and subsequently improves the initialization by iterating the variational E- and M-steps. The variational lower bound is used as a convergence criterion. If the VLB does not change by a threshold of 0.01, we regard the algorithm as converged. We set an upper limit of 500 for the iterations of the VBEM algorithm. We use the learned approximate posterior to calculate the posterior predictive distribution. We finally use the mean of the posterior predictive distribution to make predictions on test data.

We use the following datasets to evaluate the predictive accuracy of DP-GLM:

- **Simulated sine data** (*Sine*): This dataset has one input dimension and one output dimension.
- **Cosmic microwave background data** (*CMB*) [12, 5]: This dataset is characterized by heteroscedastic data. The level of noise in the data is high and input-dependent. The CMB dataset has one input dimension and one output dimension.
- **Simulated forward kinematics data of a robot arm with one joint** (*Fk_1_joint*): The Fk_1_joint dataset is a forward kinematics dataset of a robot arm with one joint. The input variable is the joint angle and the output variables are the x- and y- coordinates of the endeffector.
- **Simulated forward kinematics data of a robot arm with three joints** (*Fk_3_joint*): The input variables are three joint angles. The target variables are the x- and y- coordinates of the endeffector.
- **Inverse dynamics dataset of a SARCOS robot arm** (*Sarcos*) [13, 70, 15]: This is a high-dimensional dataset of a real SARCOS robot arm. There are 21 input variables and 7 targets. The inputs are the position, velocity, and acceleration of the 7 joints. The targets are the motor torques of the joints.

Further details on the datasets are given in the following sections.

6.1.1 Explained Variance Score as a Metric for the Prediction Accuracy

For every dataset, we evaluate the prediction accuracy and the number of models that are used for the predictions. The prediction accuracy is assessed using the `sklearn.metrics.explained_variance_score` metric of the `sklearn` package. We call this metric the *explained variance score*. The best explained variance score is 1.0 and lower values are worse. The score can also be negative, as the prediction for test data can be arbitrarily bad. It is important to note that the explained variance score is dependent on the dataset. If there is a lot of noise in the data, the score is comparatively low. A fitted regression curve is not able to explain a large share of the variance for very noisy datasets.
The score is calculated as follows:

$$\text{explained\_variance}(y, \hat{y}) = 1 - \frac{\text{Var}\{y - \hat{y}\}}{\text{Var}\{y\}}$$

y is the ground truth for the target and $\hat{y}$ is the prediction. We use the argument `multioutput='variance_weighted'`. If there is more than one output dimension, the individual score of each dimension is weighted by the variance of the corresponding dimension. If the target dimensions are of different scale, the score puts more importance on explaining dimensions with higher variance. [71]

6.1.2 Three Types of Experiments

For every dataset we evaluate the predictive accuracy and the number of learned models on test data. We do this by using different settings of the data and the prior hyperparameters. The following experimental settings are used:

- **Training sample size:** For every dataset we use five different settings of the training sample size to evaluate the predictive accuracy and the number of linear models. The average number of training samples can vary for the datasets. For these experiments, the stick-breaking prior is used. The hyperparameter $\gamma_{0,2}$ of the Beta distribution is held constant at $\gamma_{0,2} = 10$.
- **Hyperparameter of the stick-breaking prior:** We then fix the largest sample size from the training sample size evaluation to assess the influence of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior on the prediction accuracy and the number of linear models. In these experiments, the truncation level K of the stick-breaking prior, which is an upper bound on the number of components of the approximate posterior, is set to 100.
- **Hyperparameter of the Dirichlet prior:** To asses the stick-breaking prior we evaluate the Dirichlet prior as a finite mixture alternative (see section 4.2). We fix the largest sample size from the training sample size evaluation and use different values α_0 for the symmetric Dirichlet prior. We then evaluate the prediction accuracy and the number of linear models. We set the number of components to 100. In the experiments, in general, many components will not be represented in the data as their mixing weights are indistinguishable from their prior values. α_0 can be regarded as the effective prior counts for a component [7, 8].

6.1.3 Choice of Hyperparameters

For creating the violin plots in the following sections, each experiment is run for 100 times. In each run for some hyperparameters of the prior new values are drawn from a uniform distribution. The details of the initialization of the priors can be found in appendix E.
For making the derivation of the VBEM algorithm more compact we assume the Wishart distribution as a prior in this thesis. For the implementation the inverse Wishart distribution is used, which is closely related to the Wishart distribution. A draw from a Wishart distribution is a precision matrix, whereas a draw from an inverse Wishart is a covariance matrix. To convert the draws of the two distributions from a precision matrix to a covariance matrix and vice versa, a matrix inversion operation suffices. In our implementation, we use the inverse Wishart as a prior on the covariance matrices of the clusters and as a prior on the covariances matrices of the linear models. The form of the composite conjugate prior for DP-GLM, for the case of inverse Wishart distributions, can also be found in appendix E.

6.1.4 Violin Plots

We visualize the prediction accuracy and the number of linear models for 100 runs of each setting of the experiments. Violin Plots combine the ideas of boxplots and kernel density estimation. They show the extreme values, as well as the whole distribution of the dataset in between the extreme values. The distribution of the dataset is estimated with kernel density estimation. The black bar within the violin plots shows the lower quartile (0.25-quantile) and the upper quartile (0.75-quantile). This means that 50% of the data lies within the black bar. The white dot in the violin plot is the median value. If no black bar is visible, the lower quartile and the upper quartile are identical to the median value and more than 50% of the data is equal to the median value. [72]

6.2 Sine Dataset

The first dataset that we evaluate is the sine dataset. The training dataset is displayed in Figure 6.1a in black. In figure 6.1a we additionally see the training data fitted into clusters, with the prediction given one draw of the approximate posterior after the convergence of the algorithm (red). The model learns an appropriate number of models, as well as a

Table 6.1.: Median of the explained variance score for different choices of α_0 of the Dirichlet prior. The best results for each dataset are marked in bold font.

	α_0 of Dirichlet prior - Median						
	0.01	0.1	1	5	10	50	100
Sine	0.121	0.949	0.953	**0.957**	0.954	0.874	0.863
CMB	0.127	**0.170**	0.152	0.086	0.079	0.073	0.072
Fk_1_joint	0.981	0.991	0.997	0.998	**0.999**	**0.999**	**0.999**
Fk_3_joints	0.635	0.918	0.925	0.927	0.944	**0.970**	**0.970**
Sarcos	0.930	0.941	0.943	0.944	0.944	0.944	**0.947**

Table 6.2.: Median of the explained variance score for different choices of $\gamma_{0,2}$ of the Stick-breaking prior. The best results for each dataset are marked in bold font.

	$\gamma_{0,2}$ of the Stick-breaking prior - Median						
	0.1	1	10	50	100	500	1000
Sine	0.002	0.282	0.952	0.952	**0.953**	**0.953**	**0.953**
CMB	0.011	0.127	0.153	**0.166**	0.153	0.152	0.152
Fk_1_joint	0.318	0.981	0.992	0.996	**0.997**	**0.997**	**0.997**
Fk_3_joints	0.069	0.687	0.922	0.933	0.924	**0.927**	0.923
Sarcos	0.929	0.931	0.942	**0.943**	**0.943**	**0.943**	0.942

linear relationship between the training input and outputs within the clusters. We can observe that sometimes additional linear models are learned at the peaks of the sine curve, where relatively many training samples are concentrated. In Figure 6.1b we see a test data set of one-fifth of the training data (black). The red regression line represents the mean of the predictive posterior distribution for the training samples. The algorithm successfully produces a smooth and accurate prediction for test data. The green lines represent the mean of the predictive posterior distribution plus and minus two standard deviations. The variance of a mixture, such as a mixture of regression models, is calculated according to the law of total variance: [73]

$$\mathrm{Var}(X) = E[\mathrm{Var}(X|\Theta)] + \mathrm{Var}[E(X|\Theta)]$$

The variance of a mixture thus is a mixture of the variances plus an additional variability that is caused by component uncertainty. We take the square root of the variance to get the standard deviation of the mixture at each training data point.

Figure 6.2 shows a detailed statistical evaluation of the sine dataset using violin plots. Additionally, the median of the explained variance score for the stick-breaking prior and the Dirichlet prior are summarized in the tables 6.1 and 6.2 for all datasets. The tables A.1 and A.2 in appendix A additionally sum up the results using the mean as an alternative metric.

In figure 6.2 each row summarizes the results of one of the three types of experiments that we have conducted. The first row element (left) hereby always shows the explained variance score on the y-axis, whereas the second row element (right) displays the corresponding number of linear models on the y-axis. The first row in figure 6.2 contains the violin plots for different training sample sizes on the x-axis. The second row evaluates the type of experiment, where we have fixed the largest size of the training data set from the evaluation in the first row. We assess different values of the $\gamma_{0,2}$

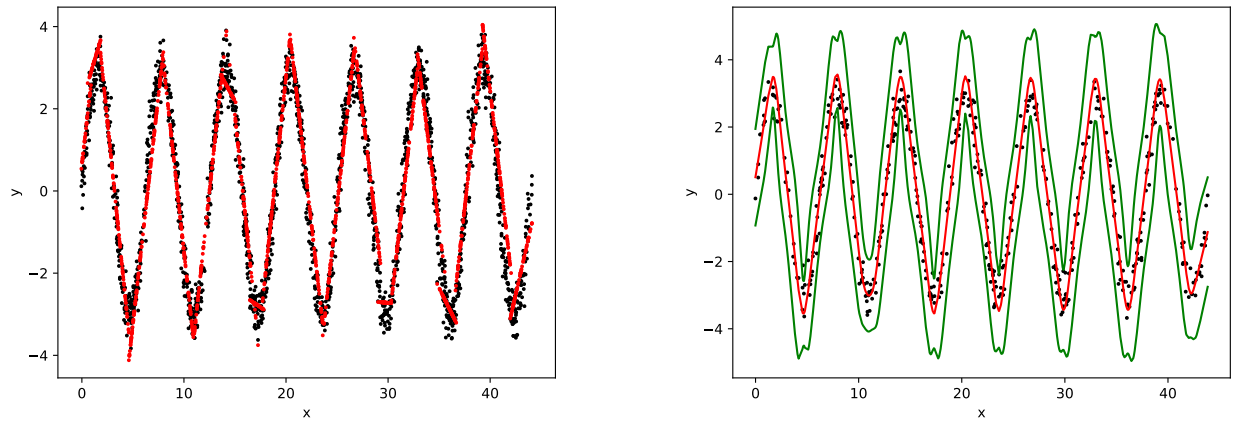


Figure 6.1.: Prediction of DP-GLM on training data and test data for the sine data set. (a) We see the training data (black dots) fitted into clusters. The red dots represent the prediction for training data according to one draw from the approximate posterior distribution. (b) The test data is displayed as black dots. The smooth regression estimate (red line) represents the mean of the posterior predictive distributions for the test data. The green lines are the mean of the posterior predictive distribution plus / minus two standard deviations.

hyperparameter of the Stick-breaking prior (x-axis). In contrast to the second row, the third row in figure 6.2 replaces the stick-breaking prior by a Dirichlet prior. We analyze various values of the α_0 parameter (x-axis) and its' impact on the prediction accuracy and the number of linear models (y-axis). Each of the violin plots in figure 6.2 has been created by executing 100 experiments. Since some hyperparameters of the prior are drawn from uniform distributions, the initialization can vary for each run of an experiment. Details on the initialization can be found in appendix E.

Evaluation of the Training Sample Size

The first row in Figure 6.2 shows the influence of the training sample size on the explained variance score and the number of linear models. We use a stick-breaking prior with a hyperparameter of $\gamma_{0,2} = 10$. In figure 6.2a we observe that the accuracy increases for the number of samples in the training set. Starting from 750 data points relatively high explained variance scores are reached. For 1500 samples we have a median of the explained variance score of 0.95. The median is denoted by the white dot in the violin plots. The black bar indicates the range from the 0.25-quantile to the 0.75-quantile. A sample size of only 250 or 500 data points is not sufficient to learn the sinus function from figure 6.1. For 250 data points, some experiments even have negative scores for the explained variance scores. In 6.2a we observe that also for larger sample sizes there are outliers with low explained variance score. We attribute this to the fact, that each run uses uniformly initialized prior hyperparameters. The outliers possibly represent bad initializations of the prior. Additionally, the Gibbs sampler, which initializes the VBEM algorithm, is of stochastic nature. it generates samples from modes of the posterior, which are a bad initialization for VBEM.
In figure 6.2b we observe that the number of linear models roughly correlates with the results for the explained variance score. For small training sample size, few linear models are learned, that cannot represent the sine function accurately. If 750 or more training samples are used, the median of the number linear models stays within the range of 16 to 17 models.

Evaluation of the Stick-breaking Prior and the Dirichlet Prior

The second row in figure 6.2 shows the explained variance score and the number of linear models for a varying hyperparameter $\gamma_{0,2}$ of the stick-breaking prior. The size of the dataset is fixed to 1500 samples. In figure 6.2c we see that values of 0.1 and 1 for $\gamma_{0,2}$ are too small. Higher values in the range of 10 to 1000 consistently produce high accuracy scores. The highest median score of 0.953 is achieved by a setting of the hyperparameter of 100, 500 or 1000. 6.2d shows that the number of linear models increases with the parameter $\gamma_{0,2}$. At a certain threshold of $\gamma_{0,2}$, the number of linear models does not increase anymore with a higher value of $\gamma_{0,2}$. In the range of $\gamma_{0,2} = 10$ to $\gamma_{0,2} = 1000$ the median number of models stays in a range from 17 to 19.
The last row in figure 6.2 shows our evaluation of the hyperparameter α_0 of the Dirichlet prior that replaces the stick-breaking prior. In comparison to the stick-breaking prior it is noteworthy, that a higher α_0 consistently raises the number of linear models (see 6.2b). This is different for the stick-breaking prior, where the median value does not exceed 20

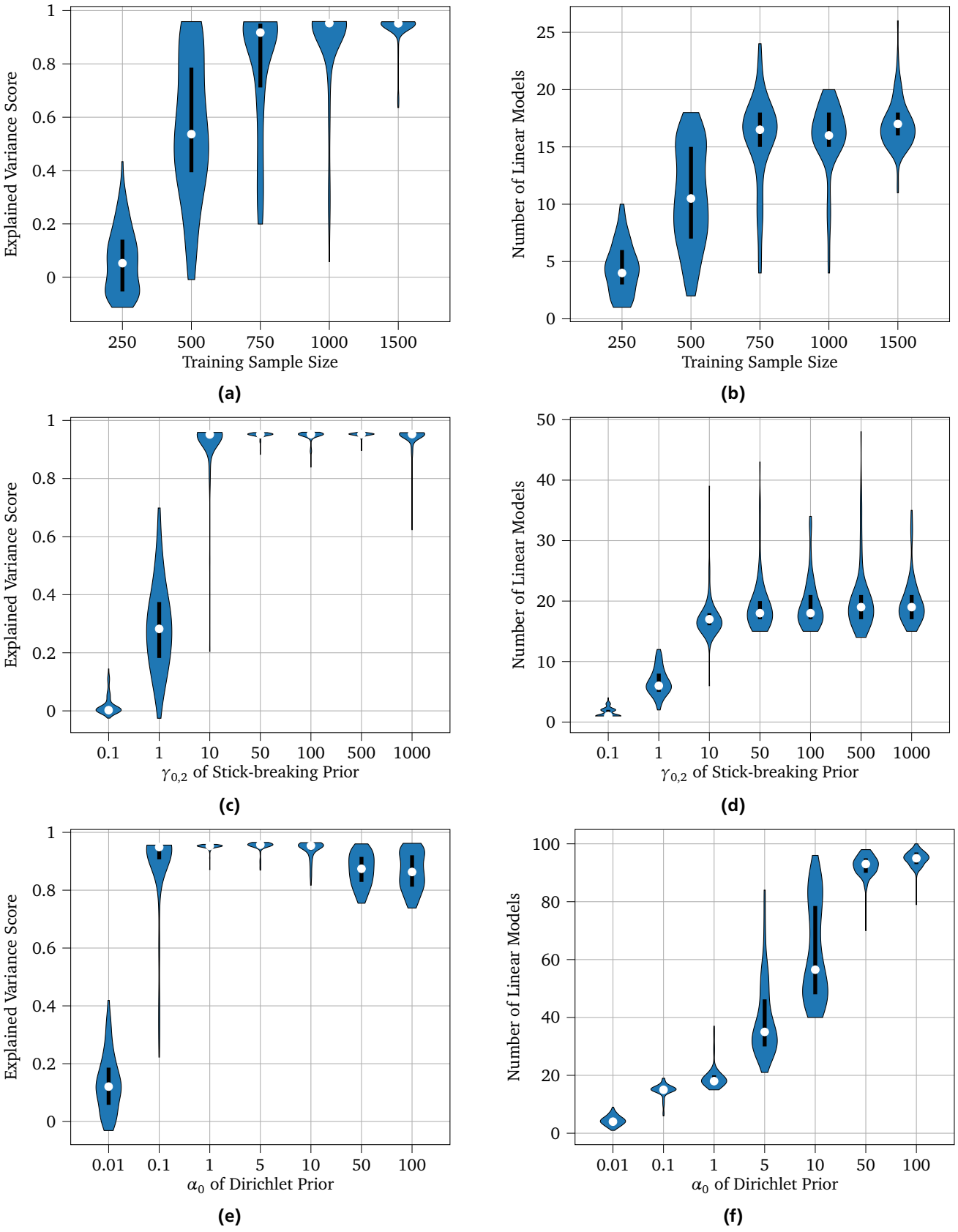
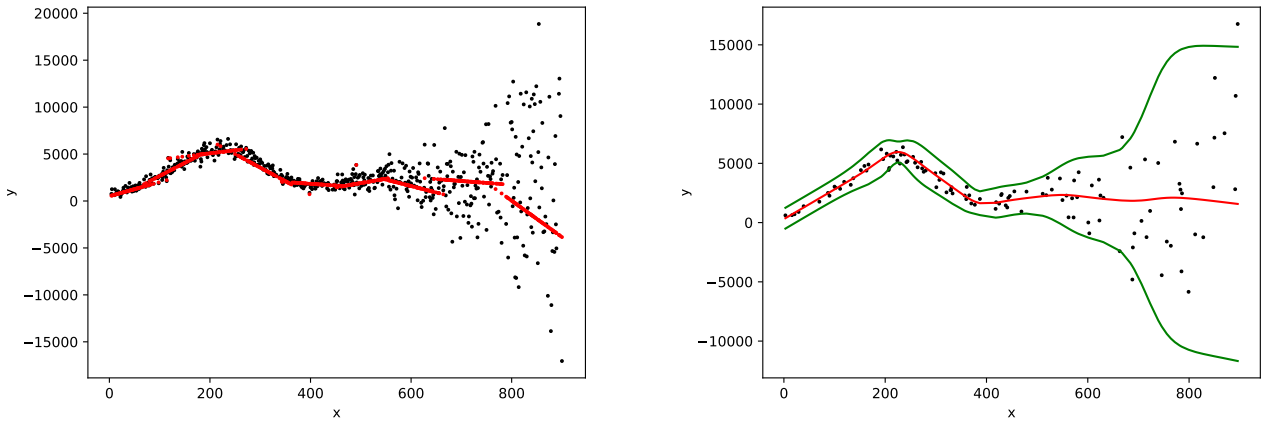


Figure 6.2.: Sine dataset: Statistical analysis of DP-GLM with simulated sine data using violin plots. **(a) and (b)**: Impact of the training sample size on the explained variance score (a) and the number of linear models (b). **(c) and (d)**: Influence of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior on the explained variance score (c) and the number of linear models (d). **(e) and (f)**: Effect of the α_0 hyperparameter of the Dirichlet prior on the explained variance score (e) and the number of linear models (f).



(a) (b)

Figure 6.3.: Prediction of DP-GLM on training data and test data for the CMB data set. (a) The training data (black dots) is fitted into clusters. The red dots represent the prediction for training data according to one draw from the approximate posterior distribution. (b) We make predictions on test data (black dots). The smooth regression estimate (red line) represents the mean of the posterior predictive distributions for the test data. The green lines are the mean of the posterior predictive distribution plus / minus two standard deviations.

models. For $\alpha_0 = 100$ we have a median result of 95 models. This high number of models is not necessary for approximating the suggested sine function and correlates with a worse performance in the accuracy score. For $\alpha_0 = 100$ we have a score of 0.863. The highest median score is 0.957 for $\alpha_0 = 5$. This is slightly higher than the highest median score of the stick-breaking prior.

We conclude that for a proper choice of the training sample size and the prior hyperparameters, DP-GLM achieves a high prediction accuracy on the simulated sine dataset. The stick-breaking prior is more robust to the choice of the hyperparameters, as there is no decline in the median score for relatively high values of $\gamma_{0,2}$.

6.3 Cosmic Microwave Background Dataset

The second dataset we are evaluating is the cosmic microwave background (CMB) dataset [12, 5]. The training data is shown in Figure 6.3a. In this figure, we also see the prediction for one draw of the posterior (red). We observe that the data is fitted to an overall of nine clusters. For each cluster, a linear model is learned. The approximate posterior is used to calculate the predictive distribution to make predictions on test data. This is depicted in figure 6.3b. The red line shows the prediction using the mean of the predictive distribution. The green lines represent the mean plus and minus two standard deviations. For calculating the standard deviation we use the law of total variance. In figure 6.3b we can observe that DP-GLM can deal with input-dependent variance and adjusts the uncertainty of the predictions accordingly. The uncertainty of the predictions is higher on the right side of x-axis the plot, where the input data has the highest intensity of noise. In figure 6.4 the statistical evaluation of the CMB dataset is summarized. The plot has the same structure as the one for the sine dataset. We have three rows, that focus on the training sample size, and the hyperparameters of the stick-breaking prior and the Dirichlet prior.

Evaluation of the Training Sample Size

In Figure 6.4a we see the effect of the training sample size on the explained variance score. An observation is that the explained variance score is overall much lower than for the sine data set. This can be explained by the fact, that the CMB dataset is very noisy. For noisy datasets, a fitted curve can not make very accurate predictions. A second finding is that, in contrast to the sine dataset, there is a less linear relationship of training sample size and explained variance score. For 100 training samples, a score of 0.573 is reached. The sample sizes 200, 300, 400 and 600 lie in a range of medians close to zero up to a score of 0.169 for 600 samples. We can explain this fact by the heteroscedasticity of the CMB dataset. If there are only 100 training data points, it is less likely that these training data points include some of the outliers of the dataset. These outliers with a large variance in comparison to the rest of the dataset can be seen in figure 6.4 for $x = 800$ and larger values on the x-axis. If the outliers are not included in the training samples, it becomes easier to fit the data with a small number of linear models.

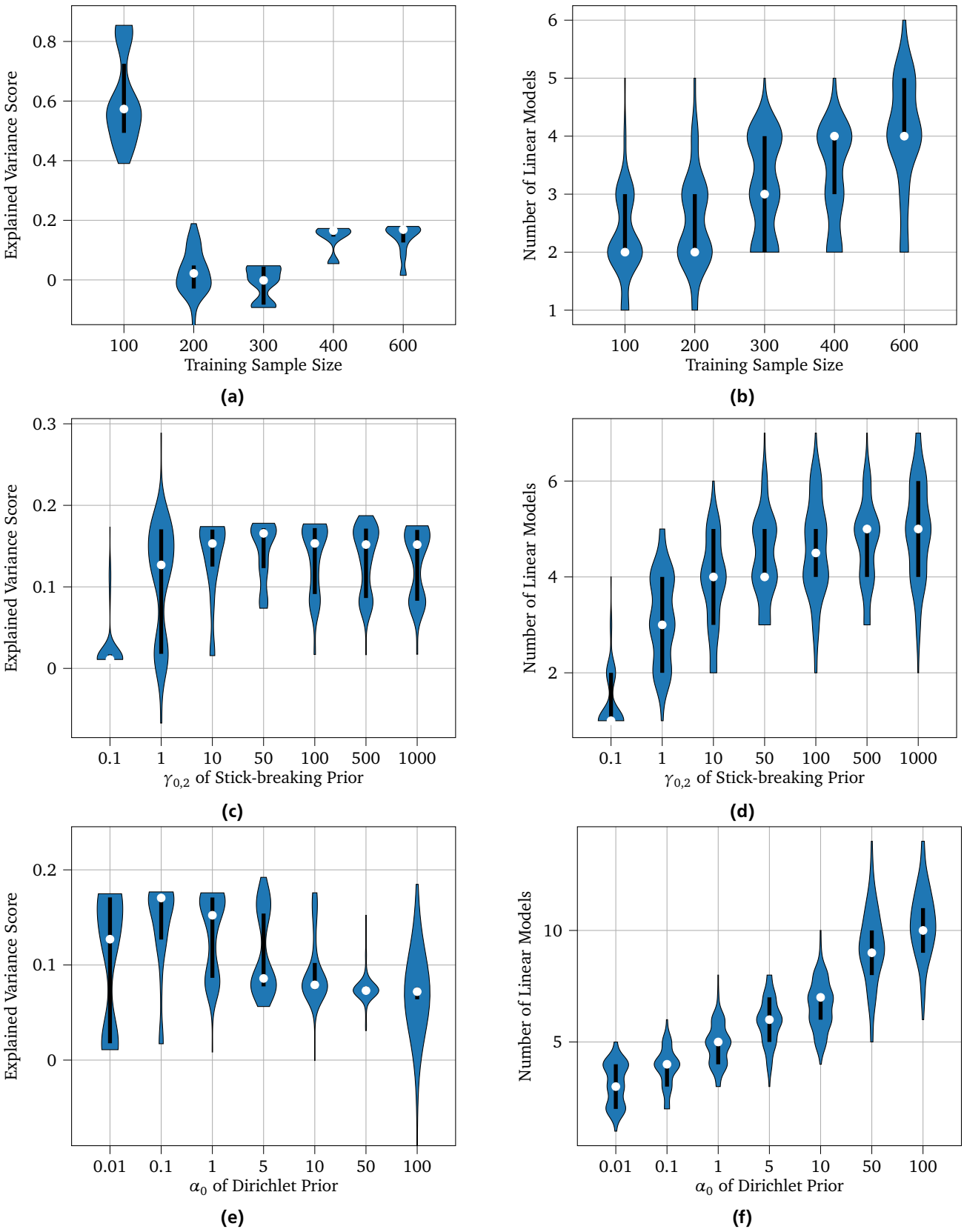


Figure 6.4.: CMB dataset: Statistical analysis of DP-GLM with cosmic microwave background data using violin plots. **(a) and (b)**: Impact of the training sample size on the explained variance score (a) and the number of linear models (b). **(c) and (d)**: Influence of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior on the explained variance score (c) and the number of linear models (d). **(e) and (f)**: Effect of the α_0 hyperparameter of the Dirichlet prior on the explained variance score (e) and the number of linear models (f).

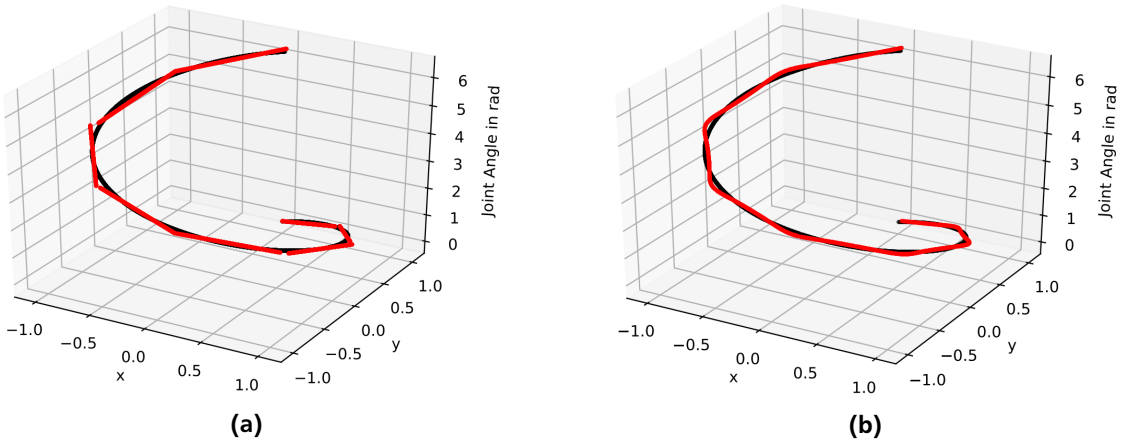


Figure 6.5.: Prediction of DP-GLM on training data and test data for forward kinematics data set with one joint (Fk_1_joint). (a) We see the training data (black) fitted into clusters. The red dots represent the prediction for training data according to one draw from the approximate posterior distribution. (b) The test data is displayed as black dots. The smooth regression estimate (red line) represents the mean of the posterior predictive distributions for the test data.

The medians of the number of linear models lie within a small range of 2 to 4 models. We again see the tendency of an increasing number of models for larger sample sizes.

Evaluation of the Stick-breaking Prior and the Dirichlet Prior

In 6.4c we can confirm the finding of the sine dataset that the accuracy score is not very sensitive to the setting of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior, as long as the parameter is not chosen too small. For a value of 0.1 and 1 the regression performs relatively bad. The best median score of 0.166 is achieved for $\gamma_{0,2} = 50$. The number of linear models in 6.4d ranges from only one model to a median value of 5 models. As in the case of the sine dataset for high values of $\gamma_{0,2}$ the number of modelsdoes not change substantially.
In figures 6.4e and 6.4f we see that there is a correlation between high values of α_0, an increasing number of linear models, and a decreasing explained variance score. The stick-breaking prior indeed seems to be less sensitive to the hyperparameters than the Dirichlet prior. On the other hand, the overall best median score of 0.170 is achieved by using a Dirichlet prior with $\alpha = 0.1$. This is a slightly higher value than a score of 0.166 for the stick-breaking prior with $\gamma_{0,2} = 50$.

6.4 Forward Kinematics Dataset with One Joint

The third dataset is a forward kinematics dataset of a robot arm with one joint. An equivalent analogy for the data is the kinematics of a pendulum. The data is simulated according to the following equations:

$$x = l_1 \cdot \cos(\theta)$$
$$y = l_1 \cdot \sin(\theta)$$

In these equations, θ is the joint angle and x and y are the x- and y-coordinates of the endeffector of the robot arm. l_1 is the length of the arm, which is set to 1. For the data generation, we uniformly draw an angle from the interval $[0, 2\pi)$ and calculate the respective endeffector coordinates x and y.
We thus have a regression problem with one input, θ, and two outputs, x, and y. The z dimension of the endeffector is ignored, which means that we have the kinematic of a robot arm within a plane and the y-coordinate of the endeffector is kept constant. Apart from the fact that there are two output variables, there is another feature of the Fk_1_joint dataset that differentiates it from the sine and the CMB datasets. There is no noise added in the equations above, whereas there is noise in the previously evaluated datasets.
Figure 6.5 shows a three-dimensional plot of the output variables x and y, as well as the target variable of the joint angle. The black training data in figure 6.5a has no noise added. In figure 6.5a we see a draw from the training data, where 8 linear models are in use. The plot on the right hand (6.5b) side shows the corresponding smooth prediction on test data.

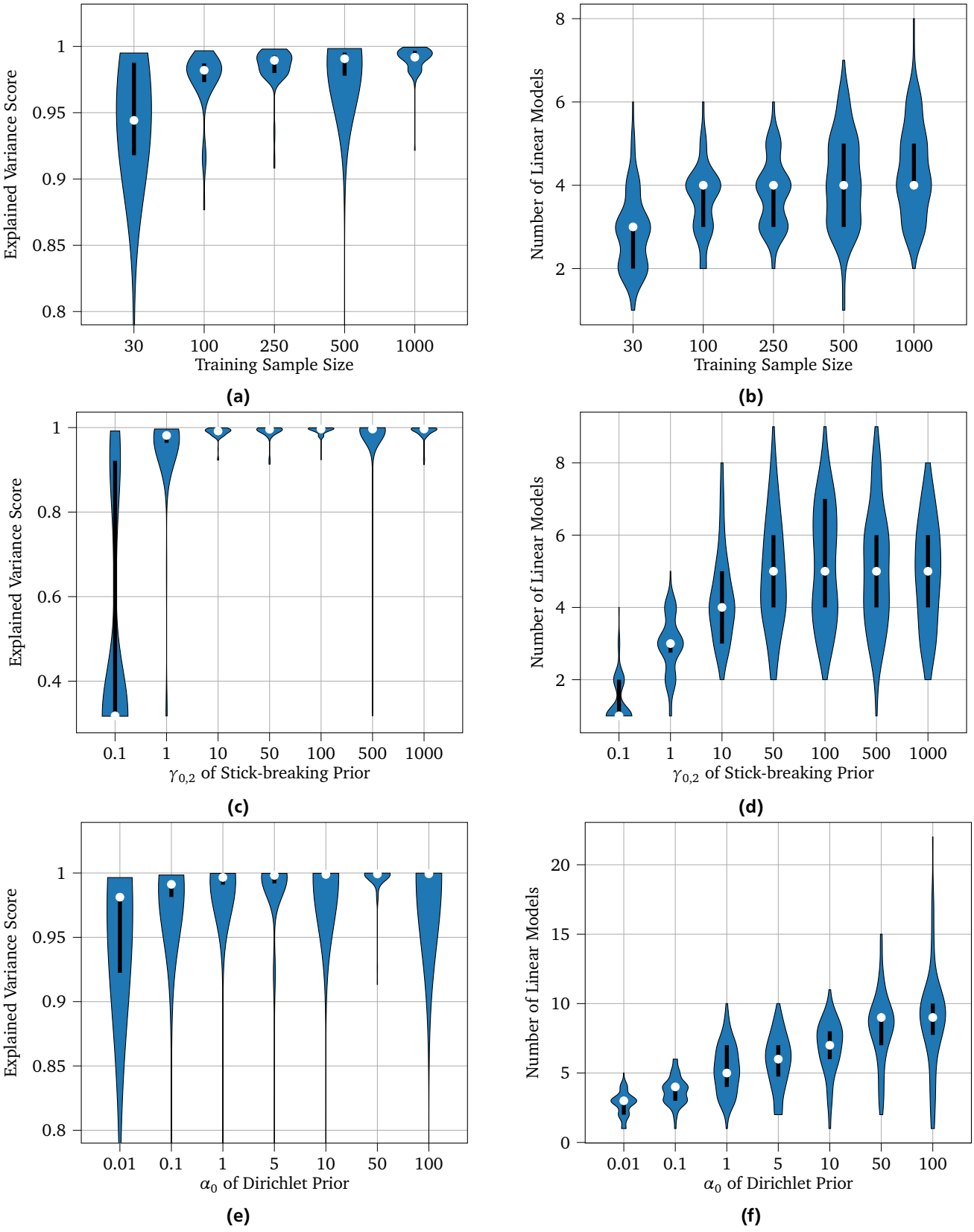


Figure 6.6.: Fk_1_joint dataset: Statistical analysis of DP-GLM with forward kinematics data of a robot arm with one joint using violin plots. **(a) and (b)**: Impact of the training sample size on the explained variance score (a) and the number of linear models (b). **(c) and (d)**: Influence of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior on the explained variance score (c) and the number of linear models (d). **(e) and (f)**: Effect of the α_0 hyperparameter of the Dirichlet prior on the explained variance score (e) and the number of linear models (f).

Evaluation of the Training Sample Size

As there is no noise added to the dataset, we can achieve very high explained variance scores. Figure 6.6a and 6.6b show the influence of the size of the training set on the prediction accuracy and the number of linear models. For a sample size of 1000 we have an accuracy score of 0.992. It is worth noting that an accuracy score of 0.944 can be achieved with only 30 training data points. As the violin plots indicate, there are some outliers of experiments with a low accuracy score. We attribute this to the stochastic initialization of the hyperparameters. The median number of linear models ranges from 3 to 4 models.

Evaluation of the Stick-breaking Prior and the Dirichlet Prior

Figure 6.6a and 6.6b confirm our results from the other datasets. The stick-breaking prior is not sensitive to the hyperparameter $\gamma_{0,2}$, in case the value of the hyperparameter is not chosen too low. For a $\gamma_{0,2}$ of 0.1, 1 and 10 a median of 1, 3 and 4 linear models is used. For higher $\gamma_{0,2}$ the median stays constant at 5 linear models. The highest median explained variance score of 0.997 is achieved by a value of the hyperparameter $\gamma_{0,2}$ of 10, 100 or 1000.
We have the overall highest median explained variance score of 0.999 for the Dirichlet prior with a α_0 of 10, 50 or 100.

6.5 Forward Kinematics Dataset with Three Joints

The forth data set is similar to the third one. We have a simulated forward kinematics datasets of a robot arm with three joints. The dataset is generated using the following kinematic equations:

$$x = l_1 \cdot \cos(\theta_1) + l_2 \cdot \cos(\theta_1 + \theta_2) + l_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = l_1 \cdot \sin(\theta_1) + l_2 \cdot \sin(\theta_1 + \theta_2) + l_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$$

There are three joint angles θ_1, θ_2 and θ_3, as well as the endeffector position that is described by x and y. The lengths l_1, l_2 and l_3 are set to 1. As in the case of the dataset Fk_1_joint, the dataset Fk_3_joint does not have noise added to the kinematic equations. In contrast to the Fk_1_joint dataset, we now have three input dimensions and two output dimensions. For the data generation we sample the angles from a uniform distribution on the range of $[0, 2\pi)$ and then calculate the endeffector position according to the kinematic equations.

Evaluation of the Training Sample Size

Regarding the training sample size we see a pattern in the figures 6.7a and 6.7b that a larger sample size results in higher accuracy and more linear models. The median of the explained variance score for a sample size of 2500 is 0.921. Depending on the sample size the number of models ranges from 3 to 13. If we compare this dataset to Fk_1_joint we notice that the explained variance score is a bit lower and the number of used linear models is higher. This makes sense as Fk_3_joint has a total of three input angles.

Evaluation of the Stick-breaking Prior and the Dirichlet Prior

The figures 6.7c and 6.7d validate our findings for the other datasets. The stick-breaking prior is not sensitive to $\gamma_{0,2}$, if the value is chosen high enough. For values of 10, 50, 100, 500 or 1000 a median number of 13 to 14 models is used. A median explained variance score of 0.927 is the highest in the experiments, where a stick-breaking prior is assumed. This score is obtained at $\gamma_{0,2} = 500$. The Dirichlet prior accomplishes a median accuracy of 0.970 if α_0 is set to a value of 50 or 100. In contrast to the stick-breaking prior, the number of models continuously rises for higher α_0. Overall the range of the median of used models lies between 3 and 10 models.

6.6 Sarcos Inverse Dynamics Dataset

The last dataset is an inverse dynamics dataset of a real SARCOS robot arm [13, 70, 15]. A robot is governed by the following state dynamics: [13]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{u}$$

Here the variables $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are joint angles, velocities and accelerations. $\boldsymbol{u}$ denotes the torque that is applied to the joints. $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix of the robot, $\boldsymbol{C}(\boldsymbol{q}.\dot{\boldsymbol{q}})$ represents the Coriolis and centripetal forces and $\boldsymbol{G}(\boldsymbol{q})$ represents the gravity forces. $\epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ are non-linearities of the robot that are not part of the rigid body dynamics. If the aim is to learn the torques $\boldsymbol{u}$, while the above inertia, Coriolis and gravity matrices are unknown, one can learn the mapping $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \rightarrow \boldsymbol{u}$ from data. Such a mapping can then be used for model-based control.

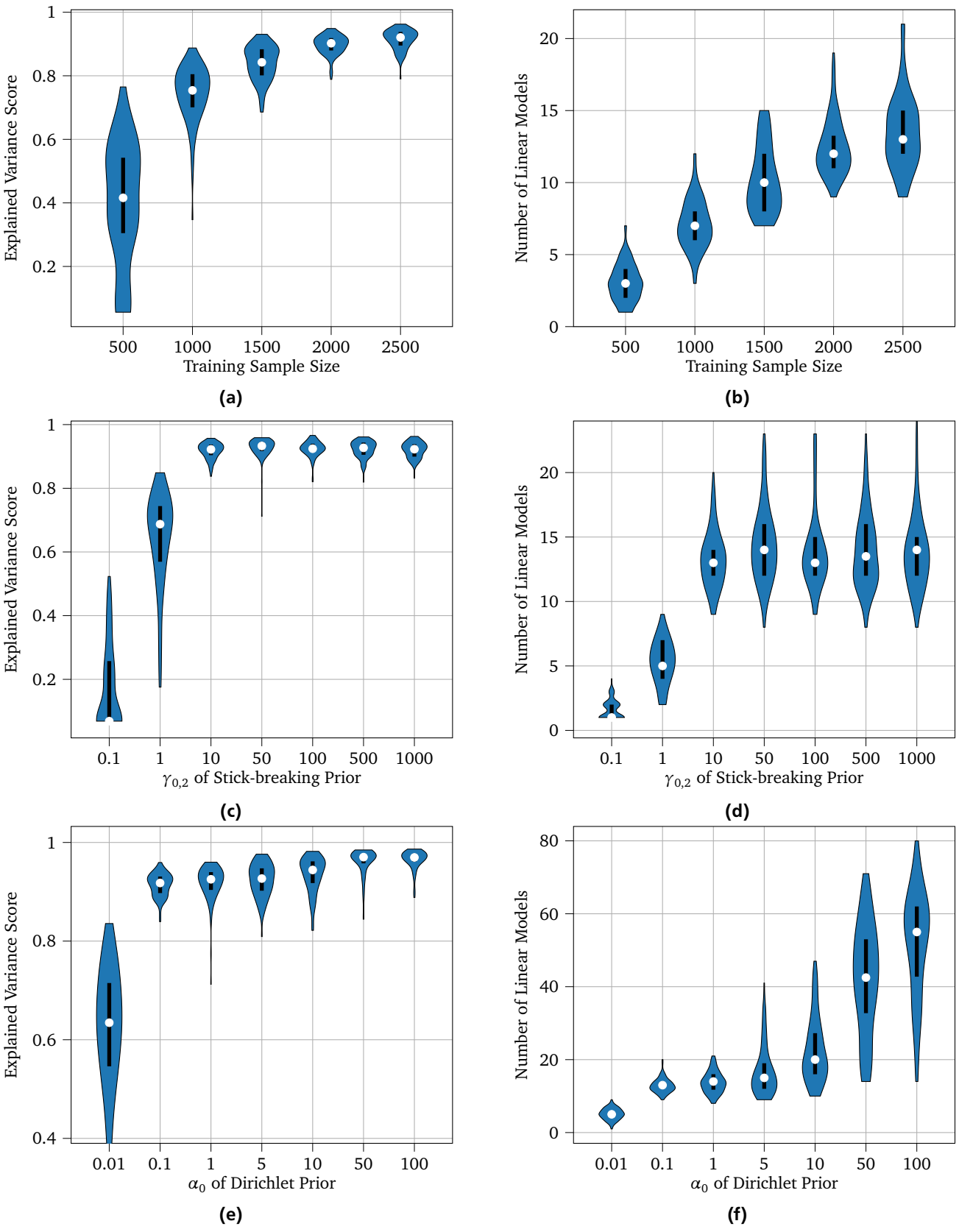


Figure 6.7.: Fk_3_joints dataset: Statistical analysis of DP-GLM with forward kinematics data of a robot arm with three joints using violin plots. **(a) and (b)**: Impact of the training sample size on the explained variance score (a) and the number of linear models (b). **(c) and (d)**: Influence of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior on the explained variance score (c) and the number of linear models (d). **(e) and (f)**: Effect of the α_0 hyperparameter of the Dirichlet prior on the explained variance score (e) and the number of linear models (f).

We want to learn the mapping of the joint angles, velocities and accelerations of 7 joints to the torques of these joints. We thus have 3 input variables per joint and a total of 21 input dimensions. The 7 output dimensions represent the torques of the joints. The data is sampled from a real SARCOS robot arm. As the data is taken from a real robot, the mapping that is learned does not only include the rigid-body dynamics, but also the nonlinearities that arise through friction and other factors. [13, 70, 15]

Evaluation of the Training Sample Size

In figure 6.8a) and 6.8b) the impact of the training sample size on the explained variance score and on the number of linear models is shown. For all sample sizes, the number of linear models has a low median value. The median ranges from one model for 500 data points to three models for 2500 training samples. DP-GLM is nevertheless able to achieve accuracy scores of over 0.9 if the sample size is increased to 2000 or 2500.

Evaluation of the Stick-breaking Prior and the Dirichlet Prior

For the Sarcos dataset, we observe in the figures 6.8c to f that neither a variation in the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior, nor a variation in the hyperparameter α_0 of the Dirichlet prior have a large impact on the number of used models. The median value for most settings stays constant at three linear models. The predictive accuracy only suffers, if the values of the hyperparameters are too low.
The overall highest median score of 0.947 is achieved with a Dirichlet prior and a hyperparameter setting of $\alpha_0 = 100$.

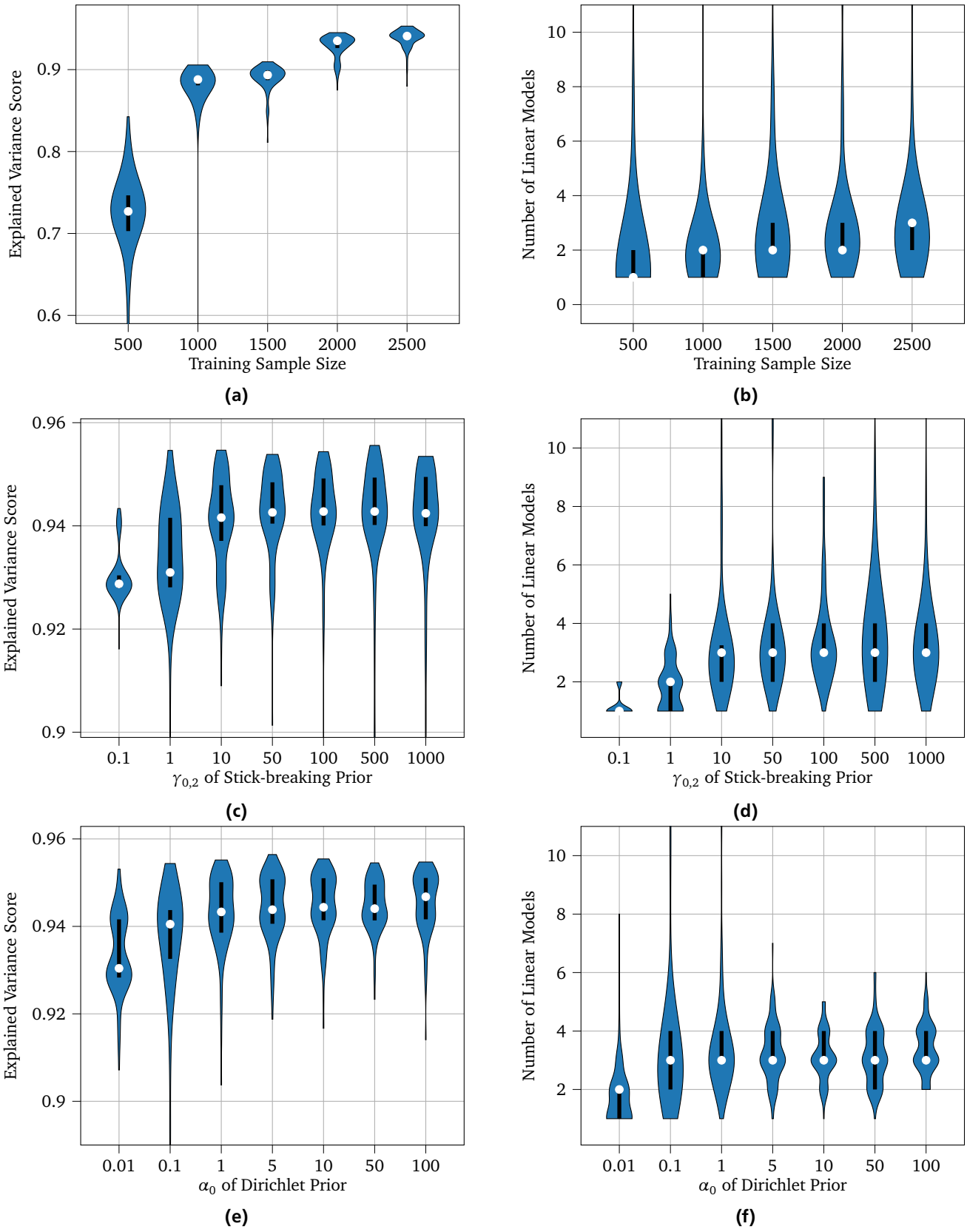


Figure 6.8.: Sarcos dataset: Statistical analysis of DP-GLM with inverse dynamics data of a SARCOS robot arm using violin plots. **(a) and (b)**: Impact of the training sample size on the explained variance score (a) and the number of linear models (b). **(c) and (d)**: Influence of the hyperparameter $\gamma_{0,2}$ of the stick-breaking prior on the explained variance score (c) and the number of linear models (d). **(e) and (f)**: Effect of the α_0 hyperparameter of the Dirichlet prior on the explained variance score (e) and the number of linear models (f).

7 Discussion and Outlook

In this thesis we have covered Bayesian Inference for regression using nonparametric infinite mixtures. After reviewing some general foundations from the field of machine learning, we gave a short introduction to different methods for regression with locally linear models. The focus was on the Bayesian regression method of Dirichlet process mixtures of generalized linear models. The idea of this regression method is to model a complex response distribution by means of simpler linear models. We listed some advantages of the method, such as a quantification of the uncertainty of predictions and the applicability to heteroscedastic data.

Learning the posterior distribution of DP-GLM is hard as intractable integrations over the hidden variables are involved. The main contributions of this thesis are the implementation of a Gibbs sampling algorithm and the derivation and implementation of a mean field variational inference algorithm. In the context of GMM, DP-GMM, and DP-GLM we also call the VI algorithm variational Bayes EM.
We adjusted an existing Gibbs sampling algorithm for infinite Gaussian mixture models [9, 10, 11] to make it applicable to DP-GLM. A similar, collapsed Gibbs sampling method has been suggested in the original DP-GLM paper [5]. The starting point for our variational Bayes EM algorithm was the variational treatment of a finite mixture of unconditional Gaussians in literature [7, 8]. We extended this basis to the infinite mixture case by using the stick-breaking construction of the Dirichlet process as a prior, as described in [56]. The last step was to extend the variational treatment to the regression case. We applied the mean field assumption to make the posterior tractable. We derived and implemented iterative mean field update equations, that have a similar structure like the expectation maximization algorithm. To our knowledge mean field variational inference has not been applied to DP-GLM before.

Using the approximate posterior we derived and implemented the posterior predictive distribution to make predictions for test data. We have evaluated our method on various data sets. The main criterion was the prediction accuracy of DP-GLM. The explained variance score was used as a metric for measuring accuracy. We evaluated the stick-breaking prior and a Dirichlet prior as a finite mixture alternative with different settings of the hyperparameters. We also varied the training sample size. The results suggest that the stick-breaking prior is less sensitive to the choice of hyperparameters than the Dirichlet prior. We evaluated DP-GLM on a broad range of data sets. We used simulated sine data, a heteroscedastic dataset of the cosmic microwave background, simulated forward kinematics data of robot arms and inverse dynamics data of a real SARCOS robot arm. The results are consistent throughout the evaluated datasets. The prediction accuracy is the highest for datasets with little noise, such as the kinematics datasets, where we achieve an explained variance score of up to 0.999.

Several topics can be addressed in future research.
In this thesis, we focused on locally linear models. The justification for this approach is, that linear models constitute a good balance between bias and variance of the global regression estimate [34]. Nevertheless, it would be important to do an empirical comparison of our approach of locally linear models and higher-order local polynomials. Hereby the local polynomials still need to be linear in the parameters to be consistent with the GLM setting but can use more complex features. The aim of such an empirical assessment could be to determine an optimal polynomial degree for local regression.

Bayesian inference, in particular variational inference, has recently been successfully applied to robotics applications [74, 75]. A potential application of DP-GLM in combination with Bayesian inference is to learn the mapping of the joint positions, velocities, and accelerations to the torques of the joint motors of a robot. This is known as learning the inverse dynamics. We have demonstrated this on a high dimensional dataset with the SARCOS dataset and achieved an explained variance score of up to 0.947 with a training sample size of 2500. It can be expected that a larger amount of training data points would further increase the number of used linear models and thus accuracy score.
In future research, the learned mapping of joint positions, velocities, and accelerations to joint torques can be used for determining feedforward motor commands. In this way, the regression method can be applied for control. A next step of evaluating the predictive accuracy of DP-GLM is, to use it for model-based control and to compare the result to other state-of-the-art function approximators. In this context, an interesting research question is to apply DP-GLM as an alternative function approximator in a reinforcement learning setting. [13]

A further interesting perspective for future research is that alternative inference methods could be applied to DP-GLM. In some cases, the mean field assumption of VI might be too inaccurate for approximating the true posterior. The mean field approximation in combination with a cost function that is derived from the information projection, leads to a focus on one of the modes of the posterior. In other cases, the iterative gradient ascent algorithm might get stuck in poor local optima. Variational autoencoders [76] are an alternative way of learning latent variable models that can circumvent these weaknesses. They are based on variational inference and add several further ideas. Firstly, variational autoencoders use black-box VI methods to maximize the evidence lower bound by gradient ascent, rather than using an EM-like approach. Hereby a low-variance gradient estimator is utilized, which is based on a reparametrization trick [77, 78]. Lastly, the approximate posterior is parametrized by a neural net, which is called the encoder. The joint distribution of data and latent variables is parametrized by a neural net that is named decoder.

Another research stream for future investigation is to further look into the paradigm of locally linear models by examining spatially and temporally correlated data. This can be done by using hidden Markov models and state space models instead of the regression approach. Many non-linear dynamical phenomena can be described by a system that switches a set of linear dynamical systems, such as in Fox et al. (2009) [79]. The authors propose a nonparametric Bayesian way of learning to switch linear dynamical systems and successfully apply this interesting approach to diverse dynamical systems like dancing honeybees and stock markets. The extension to recurrent switching linear dynamical systems makes the approach much more interpretable [80].
Regarding the Bayesian nonparametric area of research, the logistic stick-breaking process [27] is a tool that can extend the stick-breaking construction of the Dirichlet process to temporally or spatially correlated data.

Bibliography

[1] M. Laris, "Waymo launches nations first commercial self-driving taxi service in arizona." https://www.washingtonpost.com/local/trafficandcommuting/waymo-launches-nations-first-commercial-self-driving-taxi-service-in-arizona/2018/12/04/8a8cd58a-f7ba-11e8-8c9a-860ce2a8148f_story.html, Dec. 2018. Accessed: 2019-11-30.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (USA), pp. 1097–1105, Curran Associates Inc., 2012.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[5] L. A. Hannah, D. M. Blei, and W. B. Powell, "Dirichlet process mixtures of generalized linear models," *J. Mach. Learn. Res.*, vol. 12, pp. 1923–1953, July 2011.

[6] M. Beal, "Variational algorithms for approximate bayesian inference," *PhD thesis*, 01 2003.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 ed., 2007.

[8] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[9] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[10] X. Yu, "Gibbs sampling methods for dirichlet process mixture model: Technical details." https://yuxiaodong.files.wordpress.com/2009/09/technical-details-in-gibbs-sampling-for-dp-mixture-model.pdf, Sep 2009. Technical report. Accessed: 2019-11-30.

[11] X. Yu, "Derivation of gibbs sampling for finite gaussian mixture model." https://yuxiaodong.files.wordpress.com/2009/09/derivation_of_gibbs_sampling_gmm1.pdf, Sep 2009. Technical report. Accessed: 2019-11-30.

[12] C. L. Bennett, M. Halpern, G. Hinshaw, N. Jarosik, A. Kogut, M. Limon, S. S. Meyer, L. Page, D. N. Spergel, G. S. Tucker, and et al., "Firstyear wilkinson microwave anisotropy probe (wmap) observations: Preliminary maps and basic results," *The Astrophysical Journal Supplement Series*, vol. 148, p. 127, Sep 2003.

[13] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, pp. 2015–2034, Nov. 2009.

[14] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Computation*, vol. 17, pp. 2602–2634, Dec 2005.

[15] J. Peter, "Inverse dynamics data from sarcos." https://www.ias.informatik.tu-darmstadt.de/Miscellaneous/Miscellaneous. Accessed: 2019-11-30.

[16] Z. Ghahramani, "Bayesian non-parametrics and the probabilistic approach to modelling," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 371, p. 20110553, 02 2013.

[17] K. Murphy, "Pmtk3 - probabilistic modeling toolkit for matlab/octave." https://github.com/probml/pmtk3, Nov. 2019. Accessed: 2019-11-30.

[18] T. Minka, "Bayesian linear regression." https://tminka.github.io/papers/linear.html, 2000. Technical report. Accessed: 2019-11-30.

[19] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[20] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. New York: Marcel Dekker, 1988.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[22] E. Sudderth, *Graphical models for visual object recognition and tracking*. PhD thesis, Massachusetts Institute of Technology, 09 2006.

[23] Y. W. Teh, "Dirichlet processes," in *Encyclopedia of Machine Learning*, Springer, 2010.

[24] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[25] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[26] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica Sinica*, vol. 4, no. 2, pp. 639–650, 1994.

[27] L. Ren, L. Du, L. Carin, and D. Dunson, "Logistic stick-breaking process," *Journal of Machine Learning Research*, vol. 12, pp. 203–239, 01 2011.

[28] D. Blackwell and J. B. MacQueen, "Ferguson distributions via polya urn schemes," *Ann. Statist.*, vol. 1, pp. 353–355, 03 1973.

[29] C. E. Antoniak, "Mixtures of dirichlet processes with applications to bayesian nonparametric problems," *Ann. Statist.*, vol. 2, pp. 1152–1174, 11 1974.

[30] C. E. Rasmussen, "The infinite gaussian mixture model," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, (Cambridge, MA, USA), pp. 554–560, MIT Press, 1999.

[31] L. Wasserman, *All of Nonparametric Statistics (Springer Texts in Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[32] L. Györfi, M. Kohler, A. Krzyak, and H. Walk, *A Distribution-Free Theory of Non-Parametric Regression*. Springer New York, 01 2002.

[33] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979.

[34] T. Hastie and C. Loader, "Local regression: Automatic kernel carpentry," *Statistical Science*, vol. 8, no. 2, pp. 120–129, 1993.

[35] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artif. Intell. Rev.*, vol. 11, pp. 11–73, Feb. 1997.

[36] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, pp. 75–113, Feb 1997.

[37] J.-A. Ting, S. Vijayakumar, and S. Schaal, "Locally weighted regression for control," in *Encyclopedia of Machine Learning* (C. Sammut and G. I. Webb, eds.), (Boston, MA), pp. 613–624, Springer US, 2010.

[38] J. Fan and I. Gijbels, "Variable bandwidth and local linear regression smoothers," *Ann. Statist.*, vol. 20, pp. 2008–2036, 12 1992.

[39] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, pp. 49–60, 2002.

[40] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 10, no. 8, pp. 2047–2084, 1998.

[41] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space," *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, vol. Vol. 1, 05 2000.

[42] S. Vijayakumar, A. D'souza, T. Shibata, J. Conradt, and S. Schaal, "Statistical learning for humanoid robots," *Autonomous Robots*, vol. 12, pp. 55–69, Jan 2002.

[43] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixture of local experts," *Neural Computation*, vol. 3, pp. 78–88, 02 1991.

[44] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," in *Advances in Neural Information Processing Systems 14* (T. G. Dietterich, S. Becker, and Z. Ghahramani, eds.), pp. 881–888, MIT Press, 2002.

[45] S. Yuksel, J. Wilson, and P. Gader, "Twenty years of mixture of experts," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, pp. 1177–1193, 08 2012.

[46] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[47] C. M. Bishop and M. Svenson, "Bayesian hierarchical mixtures of experts," in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, (San Francisco, CA, USA), pp. 57–64, Morgan Kaufmann Publishers Inc., 2003.

[48] I. Gormley and S. Frühwirth-Schnatter, "Mixtures of experts models," in *Handbook of Mixture Analysis* (S. Frühwirth-Schnatter, G. Celeux, and C. P. Robert, eds.), Chapman and Hall/CRC, 01 2019.

[49] P. McCullagh and J. Nelder, *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series, Chapman & Hall, 1989.

[50] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[51] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1613–1622, PMLR, 07–09 Jul 2015.

[52] M. West, P. M, and M. Escobar, "Hierarchical priors and mixture models, with application in regression and density estimation," in *Aspects of Uncertainty: A Tribute to DV Lindley*, pp. 363–386, 03 1994.

[53] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 577–588, 1995.

[54] P. Mueller, A. Erkanli, and M. West, "Bayesian curve fitting using multivariate normal mixtures," *Biometrika*, vol. 83, pp. 67–79, 03 1996.

[55] S. Markus and B. C. M., "Pattern recognition and machine learning - solutions to the exercises: Web-edition." `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf`, Sept. 2009. Accessed: 2019-11-30.

[56] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Anal.*, vol. 1, pp. 121–143, 03 2006.

[57] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, Nov 1999.

[58] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, pp. 25–37, Jan 2000.

[59] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, p. 859877, Feb 2017.

[60] T. S. Jaakkola, "Tutorial on variational approximation methods," in *In Advanced Mean Field Methods: Theory and Practice*, pp. 129–159, MIT Press, 2000.

[61] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, pp. 1–305, Jan. 2008.

[62] M. Opper and D. Saad, *Advanced Mean Field Methods: Theory and Practice*. MIT press, 2001.

[63] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012. Version 20121115.

[64] T. Minka, "Estimating a dirichlet distribution." `https://tminka.github.io/papers/dirichlet/minka-dirichlet.pdf`, 2000. Technical report. Accessed: 2019-11-30.

[65] D. V. Rosen, "Moments for matrix normal variables," *Statistics*, vol. 19, no. 4, pp. 575–583, 1988.

[66] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. Chapman and Hall/CRC, Oct. 1999.

[67] K. P. Murphy, "Bayesian linear regression." `https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf`, 2007. Technical report. Accessed: 2019-11-30.

[68] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[69] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[70] S. Schaal, C. Atkeson, and S. Vijayakumar, "Real-time robot learning with locally weighted statistical learning," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, 07 2000.

[71] scikit-learn developers (BSD License), "Scikit - 3.3. metrics and scoring: quantifying the quality of predictions." `https://scikit-learn.org/stable/modules/model_evaluation.html#explained-variance-score`. Accessed: 2019-11-30.

[72] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.

[73] W. G. E. Klugman S.A., Panjer H. H., *Loss Models, From Data to Decisions*. Wiley-Interscience, 2004.

[74] E. Pignat and S. Calinon, "Bayesian gaussian mixture model for robotic policy imitation," *IEEE Robotics and Automation Letters*, vol. 4, p. 44524458, Oct 2019.

[75] E. Pignat, T. Lembono, and S. Calinon, "Variational inference with mixture model approximation: Robotic applications," 2019.

[76] D. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations*, 12 2013.

[77] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *International Conference on Learning Representations*, 2017.

[78] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *Iinternational Conference on Learning Representations*, 2017.

[79] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Nonparametric bayesian learning of switching linear dynamical systems," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 457–464, Curran Associates, Inc., 2009.

[80] S. Linderman, M. Johnson, A. Miller, R. Adams, D. Blei, and L. Paninski, "Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 914–922, PMLR, 20–22 Apr 2017.

A Results

Table A.1.: Mean of the explained variance score for different choices of α_0 of the Dirichlet prior. The best results for each dataset are marked in bold font.

	α_0 of Dirichlet prior - Mean						
	0.01	0.1	1	5	10	50	100
Sine	0.131	0.887	0.951	**0.952**	0.940	0.869	0.864
CMB	0.109	**0.142**	0.134	0.112	0.095	0.076	0.062
Fk_1_joint	0.908	0.971	0.986	0.987	0.983	0.996	**0.970**
Fk_3_joints	0.615	0.914	0.918	0.924	0.934	0.961	**0.965**
Sarcos	0.934	0.937	0.943	0.944	0.945	0.945	**0.946**

Table A.2.: Mean of the explained variance score for different choices of $\gamma_{0,2}$ of the Stick-breaking prior. The best results for each dataset are marked in bold font.

	$\gamma_{0,2}$ of the Stick-breaking prior - Mean						
	0.1	1	10	50	100	500	1000
Sine	0.018	0.289	0.930	0.950	0.948	**0.951**	0.945
CMB	0.023	0.110	0.135	**0.143**	0.136	0.135	0.130
Fk_1_joint	0.509	0.942	0.988	0.990	**0.993**	0.986	0.991
Fk_3_joints	0.173	0.639	0.917	**0.928**	0.924	0.922	0.920
Sarcos	0.930	0.933	0.941	**0.943**	**0.943**	0.942	0.942

B Summary of the VBEM Algorithm for DP-GLM

Summary of the Mean Field Update Equations

Likelihood function:
In the general case of multivariate inputs and multivariate outputs the following likelihood function can be used for DP-GLM:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{V})\, p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{y}_n | \boldsymbol{\beta}_k \boldsymbol{X}_n, \boldsymbol{V}_k^{-1}\right)^{z_{nk}} \mathrm{N}\left(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}\right)^{z_{nk}} \end{aligned}$$

Conjugate prior:
We assume the following conjugate prior:

$$\begin{aligned} p(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= p(\boldsymbol{z} \mid \boldsymbol{\pi})\, p(\upsilon)\, p(\boldsymbol{\mu} \mid \boldsymbol{\Sigma})\, p(\boldsymbol{\Sigma})\, p(\boldsymbol{\beta} \mid \boldsymbol{V})\, p(\boldsymbol{V}) \\ &= \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{\pi}) \prod_{k=1}^{\infty} \mathrm{Beta}\left(\upsilon_k | \gamma_{0,1}, \gamma_{0,2}\right) \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_0, (\lambda_0 \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k | \boldsymbol{L}_0, \nu_0) \\ &\quad \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_0, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_0\right) \mathrm{Wi}(\boldsymbol{V}_k | \boldsymbol{P}_0, \eta_0) \end{aligned}$$

Variational posterior:
As a result of applying the mean field update equation (4.2) we get an approximation variational posterior of the form:

$$\begin{aligned} q^{\star}(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= q^{\star}(\boldsymbol{z})\, q^{\star}(\upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) \\ &= \left[\prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{r}_n)\right] \left[\prod_{k=1}^{K-1} \mathrm{Beta}\left(\upsilon_k | \gamma_{k,1}, \gamma_{k,2}\right) \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_k, (\lambda_k \boldsymbol{\Sigma}_k)^{-1}\right) \mathrm{Wi}(\boldsymbol{\Sigma}_k | \boldsymbol{L}_k, \nu_k) \right. \\ &\quad \left. \prod_{k=1}^{K} \mathrm{N}\left(\boldsymbol{\beta}_k | \boldsymbol{M}_k, \boldsymbol{V}_k^{-1}, \boldsymbol{K}_k\right) \mathrm{Wi}(\boldsymbol{V}_k | \boldsymbol{P}_k, \eta_k)\right] \end{aligned}$$

Variational E-step:
The responsibilities are the variational parameters of the following Categorical distribution:

$$q^{\star}(\boldsymbol{z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} r_{nk}^{z_{nk}}$$

The responsibilities can be calculated as:

$$\begin{aligned} r_{nk} \propto \tilde{\boldsymbol{V}}_k^{\frac{1}{2}} \tilde{\boldsymbol{\Sigma}}_k^{\frac{1}{2}} \exp\Bigg(& -\frac{m}{2\lambda_k} - \frac{\nu_k}{2} (\boldsymbol{x}_n - \boldsymbol{m}_k)^T \boldsymbol{L}_k (\boldsymbol{x}_n - \boldsymbol{m}_k) \\ & -\frac{1}{2} \mathrm{Tr}\left\{\boldsymbol{K}_k^{-1} \boldsymbol{X}_n \boldsymbol{X}_n^T\right\} - \frac{\eta_k}{2} (\boldsymbol{y}_n - \boldsymbol{M}_k \boldsymbol{X}_n)^T \boldsymbol{P}_k (\boldsymbol{y}_n - \boldsymbol{M}_k \boldsymbol{X}_n) \\ & + \mathbb{E}_{q(\upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}[\log \upsilon_k] + \sum_{j=1}^{k-1} \mathbb{E}_{q(\upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}[\log(1 - \upsilon_i)] \Bigg) \end{aligned}$$

We here use:

$$\begin{aligned} \mathrm{E}_{q(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}[\log \upsilon_k] &= \psi\left(\gamma_{k,1}\right) - \psi\left(\gamma_{k,1} + \gamma_{k,2}\right) \\ \mathrm{E}_{q(\boldsymbol{z}, \upsilon, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V})}[\log(1 - \upsilon_k)] &= \psi\left(\gamma_{k,2}\right) - \psi\left(\gamma_{k,1} + \gamma_{k,2}\right) \\ \log \tilde{\boldsymbol{\Sigma}}_k &= \sum_{j=1}^{m} \psi\left(\frac{\nu_k + 1 - j}{2}\right) + m \log 2 + \log |\boldsymbol{\Sigma}_k| \\ \log \tilde{\boldsymbol{V}}_k &= \sum_{j=1}^{d} \psi\left(\frac{\eta_k + 1 - j}{2}\right) + d \log 2 + \log |\boldsymbol{V}_k| \end{aligned}$$

Variational M-step:
The update equation for the remaining variational parameters, which are needed for the variational M-step, are as follows:

$$\begin{aligned}
\gamma_{k,1} &= \sum_{n=1}^{N} r_{nk} + \gamma_{0,1} \\
\gamma_{k,2} &= \sum_{n=1}^{N} \sum_{j=k+1}^{K} r_{nj} + \gamma_{0,2} \\
\lambda_k &= \lambda_0 + N_k \\
\boldsymbol{m}_k &= \frac{1}{\lambda_k}(N_k \overline{\boldsymbol{x}}_k + \lambda_0 \boldsymbol{m}_0) \\
\upsilon_k &= \upsilon_0 + N_k \\
\boldsymbol{L}_k^{-1} &= \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \frac{\lambda_0 N_k}{\lambda_k + N_k}(\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)(\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)^T \\
\boldsymbol{K}_k &= \sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{X}_n^T + \boldsymbol{K}_0 \\
&= N_k \boldsymbol{R}_k + N_k \overline{\boldsymbol{X}}_k \overline{\boldsymbol{X}}_k^T + \boldsymbol{K}_0 \\
\boldsymbol{M}_k &= \left[\sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{y}_n^T + \boldsymbol{K}_0 \boldsymbol{M}_0^T \right] \boldsymbol{K}_k^{-1} \\
&= \left[N_k \overline{\boldsymbol{X}\boldsymbol{Y}}_k + \boldsymbol{K}_0 \boldsymbol{M}_0^T \right] \boldsymbol{K}_k^{-1} \\
\eta_k &= \eta_0 + N_k \\
\boldsymbol{P}_k^{-1} &= \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T + \sum_{n=1}^{N} r_{nk} \boldsymbol{y}_n \boldsymbol{y}_n^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T \\
&= \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T + N_k \boldsymbol{Q}_k + N_k \overline{\boldsymbol{y}}_k \overline{\boldsymbol{y}}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T
\end{aligned}$$

We use the following statistics of the data:

$$\begin{aligned}
N_k &= \sum_{n=1}^{N} r_{nk} \\
\overline{\boldsymbol{x}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n \\
\boldsymbol{S}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)^T
\end{aligned}$$

$$\begin{aligned}
\overline{\boldsymbol{X}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \\
\boldsymbol{R}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{X}_i - \overline{\boldsymbol{X}}_k)(\boldsymbol{X}_i - \overline{\boldsymbol{X}}_k)^T \\
\overline{\boldsymbol{X}\boldsymbol{Y}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{y}_n^T \\
\overline{\boldsymbol{y}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{y}_n \\
\boldsymbol{Q}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{y}_n - \overline{\boldsymbol{y}}_k)(\boldsymbol{y}_n - \overline{\boldsymbol{y}}_k)^T
\end{aligned}$$

C Probability Distributions

In the following we list some of the probability distributions that are used in this thesis. The definitions are taken from Appendix B of [7].

Beta Distribution

The Beta distribution is a distribution over a continuous variable $\mu \in [0,1]$. The two parameters a and b are constrained to be positive.

$$\text{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \tag{C.1}$$

$$\mathbb{E}[\mu] = \frac{a}{a+b} \tag{C.2}$$

$$\text{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)} \tag{C.3}$$

$$\text{mode}[\mu] = \frac{a-1}{a+b-2} \tag{C.4}$$

$\Gamma(a)$ is the Gamma function. The Beta distribution is a special case of the Dirichlet distribution for $K = 2$.

Categorical Distribution

The categorical or discrete distribution is a distribution over a K-dimensional binary variable $\boldsymbol{x}$ with components $x_k \in \{0,1\}$. The parameters are $\{\mu_k\}$. These are constrained by $0 \leq \mu_k \leq 1$ and $\sum_k \mu_k = 1$.

$$p(\boldsymbol{x}) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{C.5}$$

$$\mathbb{E}[x_k] = \mu_k \tag{C.6}$$

$$\text{var}[x_k] = \mu_k(1-\mu_k) \tag{C.7}$$

$$\text{cov}[x_j x_k] = I_{jk}(1-\mu_k) \tag{C.8}$$

$$\text{H[x]} = -\sum_{k=1}^{M} \mu_k \ln \mu_k \tag{C.9}$$

I_{jk} is the j,k element of the identity matrix. The conjugate prior for the parameters μ_k is the Dirichlet distribution.

Dirichlet Distribution

This is a distribution over K random variables μ_k, where $k = 1,\ldots,K$. The random variables are constrained to:

$$0 \leqslant \mu_k \leqslant 1, \quad \sum_{k=1}^{K} \mu_k = 1 \tag{C.10}$$

Using $\mu = (\mu_1, \ldots, \mu_K)^{\mathrm{T}}$ and $\alpha = (\alpha_1, \ldots, \alpha_K)^{\mathrm{T}}$ we have:

$$\mathrm{Dir}(\mu|\alpha) = C(\alpha)\prod_{k=1}^{K}\mu_k^{\alpha_k-1} \tag{C.11}$$

$$\mathbb{E}[\mu_k] = \frac{\alpha_k}{\widehat{\alpha}} \tag{C.12}$$

$$\mathrm{var}[\mu_k] = \frac{\alpha_k(\widehat{\alpha}-\alpha_k)}{\widehat{\alpha}^2(\widehat{\alpha}+1)} \tag{C.13}$$

$$\mathrm{cov}\big[\mu_j\mu_k\big] = -\frac{\alpha_j\alpha_k}{\widehat{\alpha}^2(\widehat{\alpha}+1)} \tag{C.14}$$

$$\mathrm{mode}[\mu_k] = \frac{\alpha_k-1}{\widehat{\alpha}-K} \tag{C.15}$$

$$\mathbb{E}[\ln\mu_k] = \psi(\alpha_k) - \psi(\widehat{\alpha}) \tag{C.16}$$

$$\mathrm{H}[\boldsymbol{\mu}] = -\sum_{k=1}^{K}(\alpha_k-1)\{\psi(\alpha_k)-\psi(\widehat{\alpha})\} - \ln C(\boldsymbol{\alpha}) \tag{C.17}$$

H denotes the entropy. We are using the following definitions:

$$C(\alpha) = \frac{\Gamma(\widehat{\alpha})}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \tag{C.18}$$

$$\widehat{\alpha} = \sum_{k=1}^{K}\alpha_k \tag{C.19}$$

$\psi(a)$ is the digamma function:

$$\psi(a) = \frac{d}{da}\ln\Gamma(a) \tag{C.20}$$

The parameters α_k are constrained to positive values. The Dirichlet distribution is a conjugate prior to the categorical distribution. H denotes the entropy.

Gamma Distribution

This is a probability distribution over a positive random variable $\tau > 0$. The parameters are a and b, which are subject to the constraints $a > 0$ and $b > 0$.

$$\mathrm{Ga}(\tau|a,b) = \frac{1}{\Gamma(a)}b^a\tau^{a-1}e^{-b\tau} \tag{C.21}$$

$$\mathbb{E}[\tau] = \frac{a}{b} \tag{C.22}$$

$$\mathrm{var}[\tau] = \frac{a}{b^2} \tag{C.23}$$

$$\mathrm{mode}[\tau] = \frac{a-1}{b} \quad \text{for } \alpha \geqslant 1 \tag{C.24}$$

$$\mathbb{E}[\ln\tau] = \psi(a) - \ln b \tag{C.25}$$

$$\mathrm{H}[\tau] = \ln\Gamma(a) - (a-1)\psi(a) - \ln b + a \tag{C.26}$$

$\psi(a)$ is the digamma function (see equation C.20). The Gamma distribution is the conjugate prior for the precision of a univariate Gaussian. H denotes the entropy.

Normal Distribution

The normal distribution is also called Gaussian. In the univariate case it is a distribution over a continuous variable $x \in (-\infty, \infty)$. It has two parameters, the mean $\mu \in (-\infty, \infty)$ and the variance $\sigma^2 > 0$.

$$\mathrm{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \tag{C.27}$$

$$\mathbb{E}[x] = \mu \tag{C.28}$$

$$\mathrm{var}[x] = \sigma^2 \tag{C.29}$$

$$\mathrm{mode}[x] = \mu \tag{C.30}$$

$$\mathrm{H}[x] = \frac{1}{2}\ln\sigma^2 + \frac{1}{2}(1+\ln(2\pi)) \tag{C.31}$$

The inverse of the variance is called precision and the square root of the variance is called standard deviation. The conjugate prior for the mean is a normal distribution. The conjugate prior for the precision is a Gamma distribution. The conjugate prior for the variance is an inverse Gamma distribution.
In the multivariate case of a D-dimensional vector $\boldsymbol{x}$, the Gaussian has a D-dimensional mean vector $\boldsymbol{\mu}$ and a $D \times D$ covariance matrix $\boldsymbol{\Sigma}$. The covariance matrix must be symmetric and positive definite.

$$\mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} \tag{C.32}$$

$$\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu} \tag{C.33}$$

$$\mathrm{cov}[\boldsymbol{x}] = \boldsymbol{\Sigma} \tag{C.34}$$

$$\mathrm{mode}[\boldsymbol{x}] = \boldsymbol{\mu} \tag{C.35}$$

$$\mathrm{H}[\boldsymbol{x}] = \frac{1}{2}\ln|\boldsymbol{\Sigma}| + \frac{D}{2}(1+\ln(2\pi)) \tag{C.36}$$

The conjugate prior on a multivariate Normal is again a multivariate Normal. The conjugate prior on the precision of a multivariate Normal is the Wishart distribution. The conjugate prior on the covariance matrix is an inverse Wishart distribution.

Wishart Distribution

The Wishart distribution is the conjugate prior for the precision of a multivariate Normal distribution.

$$\mathrm{Wi}(\boldsymbol{\Lambda}|\boldsymbol{W},\nu) = B(\boldsymbol{W},\nu)|\boldsymbol{\Lambda}|^{(\nu-D-1)/2}\exp\left(-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{W}^{-1}\boldsymbol{\Lambda}\right)\right) \tag{C.37}$$

$$B(\boldsymbol{W},\nu) = |\boldsymbol{W}|^{-\nu/2}\left(2^{\nu D/2}\pi^{D(D-1)/4}\prod_{i=1}^{D}\Gamma\left(\frac{\nu+1-i}{2}\right)\right)^{-1} \tag{C.38}$$

$$\mathbb{E}[\boldsymbol{\Lambda}] = \nu\boldsymbol{W} \tag{C.39}$$

$$\mathbb{E}[\ln|\boldsymbol{\Lambda}|] = \sum_{i=1}^{D}\psi\left(\frac{\nu+1-i}{2}\right) + D\ln 2 + \ln|\boldsymbol{W} \tag{C.40}$$

$$\mathrm{H}[\boldsymbol{\Lambda}] = -\ln B(\boldsymbol{W},\nu) - \frac{(\nu-D-1)}{2}\mathbb{E}[\ln|\boldsymbol{\Lambda}|] + \frac{\nu D}{2} \tag{C.41}$$

$\boldsymbol{W}$ is a D by D symmetric and positive definite matrix. $\psi(a)$ is the digamma function (see equation C.20). ν is called the *number of degrees of freedom* and is restricted to $\nu > D-1$. The Wishart distribution is a generalization of the Gamma distribution to the multivariate case. H denotes the entropy.

D Derivations

D.1 Variational Bayes EM Algorithm for GMM

Simplifying the Update Equation for $\boldsymbol{L}_k^{-1}$

For simplifying the update equation for $\boldsymbol{L}_k^{-1}$ we need this definition:

$$\boldsymbol{S}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\,(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_k)^T$$

And the following transformation, where we make use of the definition of $\overline{\boldsymbol{x}}_k$:

$$\begin{aligned}
\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n\boldsymbol{x}_n^T &= \sum_{n=1}^{N} r_{nk}\,(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k + \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k + \overline{\boldsymbol{x}}_k)^T \\
&= \sum_{n=1}^{N} r_{nk}\left[(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)^T + \overline{\boldsymbol{x}}_k\overline{\boldsymbol{x}}_k^T + 2\,(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)\overline{\boldsymbol{x}}_k^T\right] \\
&= \sum_{n=1}^{N} r_{nk}\left[(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)^T\right] + \sum_{n=1}^{N} r_{nk}\left[\overline{\boldsymbol{x}}_k\overline{\boldsymbol{x}}_k^T\right] + \sum_{n=1}^{N} r_{nk}\left[2\,(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)\overline{\boldsymbol{x}}_k^T\right] \\
&= N_k\boldsymbol{S}_k + N_k\overline{\boldsymbol{x}}_k\overline{\boldsymbol{x}}_k^T + 2\sum_{n=1}^{N} r_{nk}\left[(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)\overline{\boldsymbol{x}}_k^T\right] \\
&= N_k\boldsymbol{S}_k + N_k\overline{\boldsymbol{x}}_k\overline{\boldsymbol{x}}_k^T + 2\left[(N_k\overline{\boldsymbol{x}}_k - N_k\overline{\boldsymbol{x}}_k)\overline{\boldsymbol{x}}_k^T\right] \\
&= N_k\boldsymbol{S}_k + N_k\overline{\boldsymbol{x}}_k\overline{\boldsymbol{x}}_k^T
\end{aligned}$$

Using these expressions we now can simplify $\boldsymbol{T}$. We first multiply out and then inspect the constant, linear and quadratic terms with respect to $\boldsymbol{\mu}_k$.

$$\begin{aligned}
\boldsymbol{T} = {} & \lambda_0\left(\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T - \boldsymbol{\mu}_k\boldsymbol{m}_0^T - \boldsymbol{m}_0\boldsymbol{\mu}_k^T + \boldsymbol{m}_0\boldsymbol{m}_0^T\right) + \boldsymbol{L}_0^{-1} + \sum_{n=1}^{N} r_{nk}\left(\boldsymbol{x}_n\boldsymbol{x}_n^T - \boldsymbol{x}_n\boldsymbol{\mu}_k^T - \boldsymbol{\mu}_k\boldsymbol{x}_n^T + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\right) \\
& - \lambda_k\left(\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T - \boldsymbol{\mu}_k\boldsymbol{m}_k^T - \boldsymbol{m}_k\boldsymbol{\mu}_k^T + \boldsymbol{m}_k\boldsymbol{m}_k^T\right)
\end{aligned}$$

The quadratic term is as follows:

$$(\text{quadratic}) = \lambda_0\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + \sum_{n=1}^{N} r_{nk}\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T - \lambda_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T = \left(\lambda_0 + \sum_{n=1}^{N} r_{nk} - \lambda_k\right)\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T = 0$$

This follows from the definition of λ_k.
The linear term is:

$$\begin{aligned}
(\text{linear}) &= -2\lambda_0\boldsymbol{m}_0\boldsymbol{\mu}_k^T + \sum_{n=1}^{N} 2r_{nk}\boldsymbol{x}_n\boldsymbol{\mu}_k^T + 2\lambda_k\boldsymbol{m}_k\boldsymbol{\mu}_k^T \\
&= 2\left(-\lambda_0\boldsymbol{m}_0 + \sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n + \lambda_k\boldsymbol{m}_k\right)\boldsymbol{\mu}_k^T = 0
\end{aligned}$$

The last step is due to the definition of $\boldsymbol{m}_k$.
The only remaining term is the constant one:

$$\begin{aligned}
(\text{const}) &= \boldsymbol{L}_0^{-1} + \lambda_0 \boldsymbol{m}_0 \boldsymbol{m}_0^T + \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n \boldsymbol{x}_n^T - \lambda_k \boldsymbol{m}_k \boldsymbol{m}_k^T \\
&= \boldsymbol{L}_0^{-1} + \lambda_0 \boldsymbol{m}_0 \boldsymbol{m}_0^T + N_k \boldsymbol{S}_k + N_k \overline{\boldsymbol{x}}_k \overline{\boldsymbol{x}}_k^T - \lambda_k \boldsymbol{m}_k \boldsymbol{m}_k^T \\
&= \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \lambda_0 \boldsymbol{m}_0 \boldsymbol{m}_0^T + N_k \overline{\boldsymbol{x}}_k \overline{\boldsymbol{x}}_k^T - \frac{1}{\lambda_k} \lambda_k^2 \boldsymbol{m}_k \boldsymbol{m}_k^T \\
&= \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \lambda_0 \boldsymbol{m}_0 \, \boldsymbol{m}_0^T + N_k \overline{\boldsymbol{x}}_k \overline{\boldsymbol{x}}_k^T - \frac{1}{\lambda_k} (\lambda_0 \boldsymbol{m}_0 + N_K \overline{\boldsymbol{x}}_k)(\lambda_0 \boldsymbol{m}_0 + N_K \overline{\boldsymbol{x}}_k)^T \\
&= \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \left(\lambda_0 - \frac{\lambda_0^2}{\lambda_k} \right) \boldsymbol{m}_0 \boldsymbol{m}_0^T + \left(N_k - \frac{N_k^2}{\lambda_k} \right) \overline{\boldsymbol{x}}_k \overline{\boldsymbol{x}}_k^T - \frac{1}{\lambda_k} 2 (\lambda_0 \boldsymbol{m}_0) \cdot (N_K \overline{\boldsymbol{x}}_k)^T \\
&= \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \frac{\lambda_0 N_k}{\lambda_k + N_k} \boldsymbol{m}_0 \boldsymbol{m}_0^T + \frac{\lambda_0 N_k}{\lambda_k + N_k} \overline{\boldsymbol{x}}_k \overline{\boldsymbol{x}}_k^T - \frac{\lambda_0 N_K}{\lambda_k + N_k} 2 (\boldsymbol{m}_0) \cdot (\overline{\boldsymbol{x}}_k)^T \\
&= \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \frac{\lambda_0 N_k}{\lambda_k + N_k} (\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)(\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)^T
\end{aligned}$$

Our simplified update equation thus is:

$$\boldsymbol{L}_k^{-1} = \boldsymbol{L}_0^{-1} + N_k \boldsymbol{S}_k + \frac{\lambda_0 N_k}{\lambda_k + N_k} (\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)(\boldsymbol{m}_0 - \overline{\boldsymbol{x}}_k)^T$$

Variational Lower Bound

The variational lower bound on the marginal likelihood of the variational mixture of Gaussians is:

$$\begin{aligned}
\mathrm{L} &= \sum_{\boldsymbol{z}} \iiint q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \log \left\{ \frac{p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \right\} \mathrm{d}\boldsymbol{\pi} \, \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\Sigma} \\
&= \mathbb{E}[\log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] - \mathbb{E}[\log q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] \\
&= \mathbb{E}[\log p(\boldsymbol{x} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + \mathbb{E}[\log p(\boldsymbol{z} | \boldsymbol{\pi})] + \mathbb{E}[\log p(\boldsymbol{\pi})] + \mathbb{E}[\log p(\boldsymbol{\mu}, \boldsymbol{\Sigma})] \\
&\quad - \mathbb{E}[\log q(\boldsymbol{z})] - \mathbb{E}[\log q(\boldsymbol{\pi})] - \mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})]
\end{aligned}$$

All expectations are with respect to $q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The terms involving expectations over the logs of the q distributions are the negative entropies of these distributions.
According to the definition of the likelihood in equation 4.3, we have for the first term:

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{x} | \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] &= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}\left[z_{nk} \log \mathrm{N}\left(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1} \right) \right] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}] \cdot \mathbb{E}\left[-\frac{D}{2} \log 2\pi + \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right] \\
&= \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}] \cdot \left\{ -D \log 2\pi + \mathbb{E}[\log |\boldsymbol{\Sigma}_k|] - \mathbb{E}\left[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \right] \right\} \\
&= \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \cdot \left\{ -D \log 2\pi + \log \widetilde{\Sigma}_k - D \lambda_k^{-1} - \nu_k (\boldsymbol{x}_n - \boldsymbol{m}_k)^T \boldsymbol{L}_k (\boldsymbol{x}_n - \boldsymbol{m}_k) \right\}
\end{aligned}$$

The first three terms in the bracket can be written as:

$$\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \cdot \left\{ -D \log 2\pi + \log \tilde{\Sigma}_k - D \lambda_k^{-1} \right\} = \frac{1}{2} \sum_{k=1}^{K} N_k \cdot \left[-D \log 2\pi + \log \widetilde{\Sigma}_k - D \lambda_k^{-1} \right] \tag{D.1}$$

The last term in the bracket can be further simplified. We use the previously derived statement $\boldsymbol{y}^T \boldsymbol{\Sigma}_k \boldsymbol{y} = \mathrm{Tr}\left(\boldsymbol{y}\boldsymbol{y}^T \boldsymbol{\Sigma}_k\right)$ to show that:

$$\begin{aligned}\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\cdot\left\{-\upsilon_k\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)^T\boldsymbol{L}_k\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)\right\} &= -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{Tr}\left[r_{nk}\upsilon_k\cdot\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)^T\cdot\boldsymbol{L}_k\right]\\ &= -\frac{1}{2}\sum_{k=1}^{K}\mathrm{Tr}\left[\sum_{n=1}^{N} r_{nk}\upsilon_k\cdot\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)^T\cdot\boldsymbol{L}_k\right]\end{aligned}$$

We can simplify a part of this term to:

$$\begin{aligned}\sum_{n=1}^{N} r_{nk}\upsilon_k\cdot\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)^T &= \upsilon_k\sum_{n=1}^{N} r_{nk}\cdot\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k+\boldsymbol{x}_n-\overline{\boldsymbol{x}}_k\right)\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k+\boldsymbol{x}_n-\overline{\boldsymbol{x}}_k\right)^T\\ &= \upsilon_k\sum_{n=1}^{N} r_{nk}\cdot\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)^T+\upsilon_k\sum_{n=1}^{N} r_{nk}\cdot\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}_k\right)\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}_k\right)^T\\ &\quad+\upsilon_k\sum_{n=1}^{N} r_{nk}\cdot 2\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}_k\right)^T\\ &= \upsilon_k N_k\cdot\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)^T+\upsilon_k N_k\boldsymbol{S}_k+\upsilon_k\cdot 2\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\left(\sum_{n=1}^{N} r_{nk}\overline{\boldsymbol{x}}_n-\sum_{n=1}^{N} r_{nk}\overline{\boldsymbol{x}}_k\right)^T\\ &= \upsilon_k N_k\cdot\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)^T+\upsilon_k N_k\boldsymbol{S}_k\end{aligned}$$

The last term in the bracket therefore is equivalent to:

$$\begin{aligned}\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\cdot\left\{-\upsilon_k\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)^T\boldsymbol{L}_k\left(\boldsymbol{x}_n-\boldsymbol{m}_k\right)\right\} &= -\frac{1}{2}\sum_{k=1}^{K}\mathrm{Tr}\left[\upsilon_k N_k\cdot\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)^T\boldsymbol{L}_k\right]-\frac{1}{2}\sum_{k=1}^{K}\mathrm{Tr}\left[\upsilon_k N_k\boldsymbol{S}_k\boldsymbol{L}_k\right]\\ &= -\frac{1}{2}\sum_{k=1}^{K} N_k\upsilon_k\cdot\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)\boldsymbol{L}_k\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)^T-\frac{1}{2}\sum_{k=1}^{K} N_k\upsilon_k\,\mathrm{Tr}\left[\boldsymbol{S}_k\boldsymbol{L}_k\right]\end{aligned}$$

Combining these results we get the following solution for the first term of the variational lower bound L:

$$\mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Sigma})] = \frac{1}{2}\sum_{k=1}^{K} N_k\left\{\log\widetilde{\Sigma}_k - D\lambda_k^{-1}-\nu_k\,\mathrm{Tr}\left(\boldsymbol{S}_k\boldsymbol{L}_k\right)-\nu_k\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)^{\mathrm{T}}\boldsymbol{L}_k\left(\overline{\boldsymbol{x}}_k-\boldsymbol{m}_k\right)-D\log(2\pi)\right\}$$

According to the definition of the conjugate prior in equation 4.4, $p(\boldsymbol{z}|\boldsymbol{\pi})$ is a Categorical distribution. For the second term of L we get:

$$\mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{\pi})] = \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[z_{nk}\log\pi_k\right] = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\log\widetilde{\pi}_k$$

According to the definition of the conjugate prior in equation 4.4, $\log p(\boldsymbol{\pi})$ is a Dirichlet distribution. For the third term in L we get:

$$\mathbb{E}[\log p(\boldsymbol{\pi})] = \log C\left(\boldsymbol{\alpha}_0\right)+\left(\alpha_0-1\right)\sum_{k=1}^{K}\mathbb{E}\left[\log\pi_k\right] = \log C\left(\boldsymbol{\alpha}_0\right)+\left(\alpha_0-1\right)\sum_{k=1}^{K}\log\widetilde{\pi}_k$$

According to the definition of the conjugate prior in equation 4.4, $\log p(\boldsymbol{\mu},\boldsymbol{\Sigma})$ is a Normal-Wishart distribution. We use equation 19 in [63] for simplifying the term $\frac{1}{2}\log|\lambda_0\Sigma_k|$:

$$\det(c\boldsymbol{A}) = c^n\det(\boldsymbol{A}),\qquad \text{if } \boldsymbol{A}\in\mathbb{R}^{n\times n}$$

We thus have for the forth term in L:

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{\mu},\boldsymbol{\Sigma})] &= \sum_{k=1}^{K}\mathbb{E}\left[\log \mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_0,(\lambda_0\boldsymbol{\Sigma}_k)^{-1}\right)\right]+\sum_{k=1}^{K}\mathbb{E}\left[\log \mathrm{Wi}\left(\boldsymbol{\Sigma}_k|\boldsymbol{L}_0,\nu_0\right)\right] \\
&= \sum_{k=1}^{K}\mathbb{E}\left\{-\frac{D}{2}\log 2\pi+\frac{1}{2}\log|\lambda_0\boldsymbol{\Sigma}_k|-\frac{1}{2}\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)^T\left(\lambda_0\boldsymbol{\Sigma}_k\right)\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)\right\} \\
&\quad+\sum_{k=1}^{K}\mathbb{E}\left\{\log B\left(\boldsymbol{L}_0,\nu_0\right)+\frac{\nu_0-D-1}{2}\log|\boldsymbol{\Sigma}_k|-\frac{1}{2}\mathrm{Tr}\left[\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right]\right\} \\
&= \sum_{k=1}^{K}\mathbb{E}\left\{-\frac{D}{2}\log 2\pi+\frac{D}{2}\log\lambda_0+\frac{1}{2}\log|\boldsymbol{\Sigma}_k|-\frac{1}{2}\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)^T\left(\lambda_0\boldsymbol{\Sigma}_k\right)\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)\right\} \\
&\quad+\sum_{k=1}^{K}\mathbb{E}\left\{\log B\left(\boldsymbol{L}_0,\nu_0\right)+\frac{\nu_0-D-1}{2}\log|\boldsymbol{\Sigma}_k|-\frac{1}{2}\mathrm{Tr}\left[\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right]\right\} \\
&= \frac{K\cdot D}{2}\log\frac{\lambda_0}{2\pi}+\frac{1}{2}\sum_{k=1}^{K}\log\widetilde{\Sigma}_k-\frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left\{\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)^T\left(\lambda_0\boldsymbol{\Sigma}_k\right)\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)\right\} \\
&\quad+K\cdot\log B\left(\boldsymbol{L}_0,\nu_0\right)+\frac{\nu_0-D-1}{2}\sum_{k=1}^{K}\log\widetilde{\Sigma}_k-\frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left\{\mathrm{Tr}\left[\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right]\right\}
\end{aligned} \tag{D.2}$$

This expression leaves us with two expectations, that still need to be calculated.
Using equation C.39 we obtain:

$$\sum_{k=1}^{K}\mathbb{E}\left\{\mathrm{Tr}\left[\boldsymbol{L}_0^{-1}\boldsymbol{\Sigma}_k\right]\right\}=\sum_{k=1}^{K}\mathrm{Tr}\left\{\boldsymbol{L}_0^{-1}\cdot\mathbb{E}\left[\boldsymbol{\Sigma}_k\right]\right\}=\sum_{k=1}^{K}\nu_k\cdot\mathrm{Tr}\left\{\boldsymbol{L}_0^{-1}\boldsymbol{L}_k\right\}$$

For the second expectation we need equation C.33 and $\mathbb{E}\left[\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\right]=\boldsymbol{m}_k\boldsymbol{m}_k^T+\lambda_k^{-1}\boldsymbol{\Sigma}_k^{-1}$ (see equation 318 in [63]). For the second expectation we get:

$$\begin{aligned}
\sum_{k=1}^{K}\mathbb{E}\left\{\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)^T\left(\lambda_0\boldsymbol{\Sigma}_k\right)\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)\right\} &= \lambda_0\sum_{k=1}^{K}\mathbb{E}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k\cdot\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)\left(\boldsymbol{\mu}_k-\boldsymbol{m}_0\right)^T\right]\right\} \\
&= \lambda_0\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k\cdot\left(\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T-2\boldsymbol{\mu}_k\boldsymbol{m}_0^T+\boldsymbol{m}_0\boldsymbol{m}_0^T\right)\right]\right\} \\
&= \lambda_0\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k\cdot\left(\boldsymbol{m}_k\boldsymbol{m}_k^T+\lambda_k^{-1}\boldsymbol{\Sigma}_k^{-1}-2\boldsymbol{m}_k\boldsymbol{m}_0^T+\boldsymbol{m}_0\boldsymbol{m}_0^T\right)\right]\right\} \\
&= \lambda_0\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}\left[\lambda_k^{-1}\boldsymbol{I}+\boldsymbol{\Sigma}_k\cdot\left(\boldsymbol{m}_k\boldsymbol{m}_k^T-2\boldsymbol{m}_k\boldsymbol{m}_0^T+\boldsymbol{m}_0\boldsymbol{m}_0^T\right)\right]\right\} \\
&= \lambda_0\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\Sigma}_k}\left\{D\cdot\lambda_k^{-1}+\mathrm{Tr}\left[\boldsymbol{\Sigma}_k\cdot\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)^T\right]\right\} \\
&= \frac{KD\lambda_0}{\lambda_k}+\lambda_0\sum_{k=1}^{K}\mathbb{E}_{\Sigma_k}\left\{\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)\boldsymbol{\Sigma}_k\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)^T\right\} \\
&= \frac{KD\lambda_0}{\lambda_k}+\lambda_0\sum_{k=1}^{K}\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)\cdot\mathbb{E}_{\boldsymbol{\Sigma}_k}\left[\boldsymbol{\Sigma}_k\right]\cdot\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)^T \\
&= \frac{KD\lambda_0}{\lambda_k}+\lambda_0\sum_{k=1}^{K}\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)\cdot\left(\nu_k\boldsymbol{L}_k\right)\cdot\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)^T
\end{aligned}$$

Substituting the two expectations back, we get for the forth term:

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{\mu},\boldsymbol{\Sigma})] &= \frac{1}{2}\sum_{k=1}^{K}\left\{D\log\left(\lambda_0/2\pi\right)+\log\widetilde{\Sigma}_k-\frac{D\lambda_0}{\lambda_k}-\lambda_0\nu_k\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)^{\mathrm{T}}\boldsymbol{L}_k\left(\boldsymbol{m}_k-\boldsymbol{m}_0\right)\right\} \\
&\quad+K\log B\left(\boldsymbol{L}_0,\nu_0\right)+\frac{\left(\nu_0-D-1\right)}{2}\sum_{k=1}^{K}\log\widetilde{\Sigma}_k-\frac{1}{2}\sum_{k=1}^{K}\nu_k\,\mathrm{Tr}\left(\boldsymbol{L}_0^{-1}\boldsymbol{L}_k\right)
\end{aligned}$$

According to equation 4.7 the variational distribution for $\log q(\boldsymbol{z})$ is a Categorical distribution: $q^*(z) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}}$. We get for the fifth term:

$$\mathbb{E}[\log q(\boldsymbol{z})] = \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}[z_{nk}]\cdot\log r_{nk} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\cdot\log r_{nk}$$

According to equation 4.6 the variational distribution for $\log q(\boldsymbol{\pi})$ is a Dirichlet distribution. We get for the sixth term:

$$\begin{aligned}\mathbb{E}[\log q(\boldsymbol{\pi})] &= \log C(\boldsymbol{\alpha}) + (\alpha_k - 1)\sum_{k=1}^{K}\mathbb{E}[\log\pi_k] \\ &= \log C(\boldsymbol{\alpha}) + (\alpha_k - 1)\sum_{k=1}^{K}\log\widetilde{\pi}_k\end{aligned}$$

According to equation 4.10 the variational distribution for $\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a Normal-Wishart distribution. We approach this problem in a similar manner to equation D.2. We use the following expectation:

$$\begin{aligned}\frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T(\lambda_k\boldsymbol{\Sigma}_k)(\boldsymbol{\mu}_k - \boldsymbol{m}_k)\right\} &= \frac{\lambda_k}{2}\sum_{k=1}^{K}\mathbb{E}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k(\boldsymbol{\mu}_k - \boldsymbol{m}_k)(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T\right]\right\} \\ &= \frac{\lambda_k}{2}\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k(\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T - 2\boldsymbol{\mu}_k\boldsymbol{m}_k^T + \boldsymbol{m}_k\boldsymbol{m}_k^T)\right]\right\} \\ &= \frac{\lambda_k}{2}\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k\left(\mathbb{E}_{\boldsymbol{\mu}_k}\left\{\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\right\} - 2\mathbb{E}_{\boldsymbol{\mu}_k}\left\{\boldsymbol{\mu}_k\right\}\boldsymbol{m}_k^T + \boldsymbol{m}_k\boldsymbol{m}_k^T\right)\right]\right\} \\ &= \frac{\lambda_k}{2}\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}\left[\boldsymbol{\Sigma}_k\left(\boldsymbol{m}_k\boldsymbol{m}_k^T + \lambda_k^{-1}\boldsymbol{\Sigma}_k^{-1} - 2\boldsymbol{m}_k\boldsymbol{m}_k^T + \boldsymbol{m}_k\boldsymbol{m}_k^T\right)\right]\right\} \\ &= \frac{\lambda_k}{2}\sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{\Sigma}_k}\left\{\mathrm{Tr}[\boldsymbol{I}]\right\} \\ &= \frac{KD}{2}\end{aligned}$$

Using this expectation we get for the seventh term:

$$\begin{aligned}\mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})] &= \sum_{k=1}^{K}\mathbb{E}\left[\log\mathrm{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\lambda_k\boldsymbol{\Sigma}_k)^{-1}\right)\right] + \sum_{k=1}^{K}\mathbb{E}\left[\log\mathrm{Wi}(\boldsymbol{\Sigma}_k|\boldsymbol{L}_k, \nu_k)\right] \\ &= \sum_{k=1}^{K}\mathbb{E}\left\{-\frac{D}{2}\log 2\pi + \frac{D}{2}\log\lambda_k + \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T(\lambda_k\boldsymbol{\Sigma}_k)(\boldsymbol{\mu}_k - \boldsymbol{m}_k)\right\} \\ &\quad + \sum_{k=1}^{K}\mathbb{E}\left\{\log B(\boldsymbol{L}_k, \nu_k) + \frac{\nu_k - D - 1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\mathrm{Tr}\left[\boldsymbol{L}_k^{-1}\boldsymbol{\Sigma}_k\right]\right\} \\ &= \frac{K\cdot D}{2}\log\frac{\lambda_k}{2\pi} + \frac{1}{2}\sum_{k=1}^{K}\log\widetilde{\boldsymbol{\Sigma}}_k - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{\mu}_k - \boldsymbol{m}_k)^T(\lambda_k\boldsymbol{\Sigma}_k)(\boldsymbol{\mu}_k - \boldsymbol{m}_k)\right\} \\ &\quad + K\cdot\log B(\boldsymbol{L}_k, \nu_k) + \frac{\nu_k - D - 1}{2}\sum_{k=1}^{K}\log\widetilde{\boldsymbol{\Sigma}}_k - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\left\{\mathrm{Tr}\left[\boldsymbol{L}_k^{-1}\boldsymbol{\Sigma}_k\right]\right\} \\ &= \frac{K\cdot D}{2}\log\frac{\lambda_k}{2\pi} + \frac{1}{2}\sum_{k=1}^{K}\log\widetilde{\boldsymbol{\Sigma}}_k - \frac{KD}{2} \\ &\quad + K\cdot\log B(\boldsymbol{L}_k, \nu_k) + \frac{\nu_k - D - 1}{2}\sum_{k=1}^{K}\log\widetilde{\boldsymbol{\Sigma}}_k - \frac{1}{2}\sum_{k=1}^{K}\nu_k\mathbb{E}\left\{\mathrm{Tr}\left[\boldsymbol{L}_k^{-1}\boldsymbol{L}_k\right]\right\} \\ &= \frac{K\cdot D}{2}\log\frac{\lambda_k}{2\pi} + \frac{1}{2}\sum_{k=1}^{K}\log\widetilde{\boldsymbol{\Sigma}}_k - \frac{KD}{2} \\ &\quad + K\cdot\log B(\boldsymbol{L}_k, \nu_k) + \frac{\nu_k - D - 1}{2}\sum_{k=1}^{K}\log\widetilde{\boldsymbol{\Sigma}}_k - \frac{1}{2}\sum_{k=1}^{K}\nu_k\cdot D\end{aligned}$$

We rewrite this term by using the entropy H of a Wishart distribution (see equation C.41). The final form of the seventh term thus is:

$$\mathbb{E}[\log q(\boldsymbol{\mu}, \boldsymbol{\Sigma})] = \sum_{k=1}^{K} \left\{ \frac{1}{2} \log \widetilde{\Sigma}_k + \frac{D}{2} \log \left(\frac{\lambda_k}{2\pi} \right) - \frac{D}{2} - \mathrm{H}[q(\boldsymbol{\Sigma}_k)] \right\}$$

D.2 Variational Bayes EM Algorithm for DP-GLM

Simplifying the Update Equation for $\boldsymbol{P}_k^{-1}$

In this section we show how the update equation for $\boldsymbol{P}_k^{-1}$ can be simplified.
As we assume a Wishart distribution, $\boldsymbol{P}_k^{-1}$ should contain terms including $\boldsymbol{V}_k$, but be independent of $\boldsymbol{\beta}_k$. Thus the linear and quadratic terms with respect to $\boldsymbol{\beta}_k$ should cancel out. We verify this in the following.
If we multiply the terms in 4.24 out, we get for $\boldsymbol{P}_k^{-1}$:

$$\begin{aligned}
\boldsymbol{P}_k^{-1} = & \sum_{n=1}^{N} r_{nk} \left(\boldsymbol{y}_n \boldsymbol{y}_n^T - \boldsymbol{y}_n \boldsymbol{X}_n^T \boldsymbol{\beta}_n^T - \boldsymbol{\beta}_n \boldsymbol{X}_n \boldsymbol{y}_n^T + \boldsymbol{\beta}_k \boldsymbol{X}_n \boldsymbol{X}_n^T \boldsymbol{\beta}_k^T \right) \\
& + \boldsymbol{\beta}_k \boldsymbol{K}_0 \boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k \boldsymbol{K}_0 \boldsymbol{M}_0^T - \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{\beta}_k^T + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T \\
& + \boldsymbol{P}_0^{-1} \\
& - \left(\boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{M}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T + \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T \right)
\end{aligned}$$

The quadratic terms of $\boldsymbol{P}_k^{-1}$ are:

$$\text{(quadratic)} = \sum_{n=1}^{N} r_{nk} \boldsymbol{\beta}_k \boldsymbol{X}_n \boldsymbol{X}_n^T \boldsymbol{\beta}_k^T + \boldsymbol{\beta}_k \boldsymbol{K}_0 \boldsymbol{\beta}_k^T - \boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T = \boldsymbol{\beta}_k \left(\sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{X}_n^T - \boldsymbol{K}_0 - \boldsymbol{K}_k \right) \boldsymbol{\beta}_k^T = 0$$

We used the definition of $\boldsymbol{K}_k$ here.
The linear terms of $\boldsymbol{P}_k^{-1}$ are:

$$\begin{aligned}
\text{(linear)} &= -\sum_{n=1}^{N} r_{nk} \left(\boldsymbol{y}_n \boldsymbol{X}_n^T \boldsymbol{\beta}_k^T + \boldsymbol{\beta}_k \boldsymbol{X}_n \boldsymbol{y}_n^T \right) - \boldsymbol{\beta}_k \boldsymbol{K}_0 \boldsymbol{M}_0^T - \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{\beta}_k^T + \boldsymbol{\beta}_k \boldsymbol{K}_k \boldsymbol{M}_k^T + \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{\beta}_k^T \\
&= \left(-\sum_{n=1}^{N} r_{nk} \boldsymbol{y}_n \boldsymbol{X}_n^T - \boldsymbol{M}_0 \boldsymbol{K}_0 + \boldsymbol{M}_k \boldsymbol{K}_k \right) \boldsymbol{\beta}_k^T + \boldsymbol{\beta}_k \left(-\sum_{n=1}^{N} r_{nk} \boldsymbol{X}_n \boldsymbol{y}_n^T - \boldsymbol{K}_0 \boldsymbol{M}_0^T + \boldsymbol{K}_k \boldsymbol{M}_k^T \right) = 0
\end{aligned}$$

We used the definition of $\boldsymbol{M}_k$ here.
For the remaining constant terms of $\boldsymbol{P}_k^{-1}$ we use these definitions:

$$\begin{aligned}
\overline{\boldsymbol{y}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{y}_n \\
\boldsymbol{Q}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{y}_n - \overline{\boldsymbol{y}}_k)(\boldsymbol{y}_n - \overline{\boldsymbol{y}}_k)^T
\end{aligned}$$

The constant terms are:

$$\begin{aligned}
\text{(const)} &= \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T + \sum_{n=1}^{N} r_{nk} \boldsymbol{y}_n \boldsymbol{y}_n^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T \\
&= \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T + N_k \boldsymbol{Q}_k + N_k \overline{\boldsymbol{y}}_k \overline{\boldsymbol{y}}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T
\end{aligned}$$

For the simplification of the term we used a similar approach as in equation 4.23.
The simplified update equation for $\boldsymbol{P}_k^{-1}$ thus is:

$$\boldsymbol{P}_k^{-1} = \boldsymbol{P}_0^{-1} + \boldsymbol{M}_0 \boldsymbol{K}_0 \boldsymbol{M}_0^T + N_k \boldsymbol{Q}_k + N_k \overline{\boldsymbol{y}}_k \overline{\boldsymbol{y}}_k^T - \boldsymbol{M}_k \boldsymbol{K}_k \boldsymbol{M}_k^T$$

D.3 Gibbs Sampling for DP-GLM with Normal-Inverse-Gamma Priors

Likelihood
Gaussian likelihood of an input x_i:

$$F_x\left(x_i|\theta_{x,i}\right) = \mathrm{N}\left(x_i|\mu_i, \sigma^2_{x,i}\right)$$

Gaussian likelihood of all inputs $\boldsymbol{x}_k = x_1, \ldots, x_{n_k}$ of a cluster k:

$$\mathrm{L}_x\left(\boldsymbol{x}_k|\theta_{x,k}\right) = \prod_{n=1}^{n_k} \mathrm{N}\left(x_n|\mu_k, \sigma^2_{k,x}\right) \qquad n_k = \sum_{n=1}^{N} \delta\left(z_i - k\right)$$

Gaussian likelihood of a response y_i. β_0 is the offset and β_1 the slope of the linear model.

$$F_y\left(y_i|x_i, \theta_{y,i}\right) = \mathrm{N}\left(y_i|\boldsymbol{x}_n\boldsymbol{\beta}_i, \sigma^2_{i,y}\right) \qquad \boldsymbol{x}_n = [\,\hat{\boldsymbol{x}}_i\,] \quad \text{with} \quad \hat{\boldsymbol{x}}_i = [\,1 \quad x_i\,]$$
$$\boldsymbol{\beta}_i = [\,\beta_0 \quad \beta_1\,]^T$$

Gaussian likelihood of all responses $\boldsymbol{y}_k = y_1, \ldots, y_{n_k}$ of a cluster k:

$$\mathrm{L}_y\left(\boldsymbol{y}_k|\boldsymbol{x}_k, \theta_{y,k}\right) = \prod_{n=1}^{n_k} \mathrm{N}\left(y_n|\boldsymbol{X}_n\boldsymbol{\beta}_n, \sigma^2_{i,y}\right) \qquad n_k = \sum_{n=1}^{N} \delta\left(z_i - k\right)$$

Prior / base measure
The conjugate prior on the mean μ and covariance σ_x of a cluster is a Normal-inverse-Gamma distribution G_x.
The conjugate prior on the parameters $\boldsymbol{\beta}$ and σ^2_x of the corresponding GLM is a Multivariate-Normal-inverse-Gamma distribution G_y.

$$G_x\left(\theta_x\right) = \mathrm{N}\left(\mu|m_{0,x}, \sigma^2_x V_{0,x}\right)\mathrm{Ga}^{-1}\left(\sigma^2_x|a_{0,x}, b_{0,x}\right)$$
$$G_y\left(\theta_y\right) = \mathrm{N}\left(\boldsymbol{\beta}|\boldsymbol{m}_{0,y}, \sigma^2_y\boldsymbol{V}_{0,y}\right)\mathrm{Ga}^{-1}\left(\sigma^2_y|a_{0,y}, b_{0,y}\right)$$

Conditional posterior of cluster parameters $\theta_{x,k}$
The conditional posterior of the cluster parameters μ and σ_x, like the conjugate prior, also is a Normal-inverse-Gamma distribution.

$$H_x\left(\theta_{x,k}|\boldsymbol{x}_k\right) = \frac{G_x\left(\theta_{x,k}\right)\mathrm{L}_x\left(\boldsymbol{x}_k|\theta_{x,k}\right)}{\int_{\theta_x} G_x(\theta_x)\mathrm{L}_x\left(\boldsymbol{x}_k|\theta_x\right)} = \mathrm{N}\left(\mu|m_{n,x}, \sigma^2_x V_{n,x}\right)\mathrm{Ga}^{-1}\left(\sigma^2_x|a_{n,x}, b_{n,x}\right)$$

It has the following, updated parameters: [67]

$$V^{-1}_{n,x} = V^{-1}_{0,x} + n_k$$
$$\frac{m_{n,x}}{V_{n,x}} = V^{-1}_{0,x}m_{0,x} + n_k\overline{x} \qquad \overline{x} = \frac{1}{n_k}\sum_{j=1}^{n_k} x_j$$
$$a_{n,x} = a_{0,x} + n_k/2$$
$$b_{n,x} = b_{0,x} + \frac{1}{2}\left[m^2_{0,x}V^{-1}_{0,x} + \sum_{j=1}^{n_k} x^2_j - m^2_{n,x}V^{-1}_{n,x}\right]$$

Conditional posterior of GLM parameters $\theta_{y,k}$
The conditional posterior of the GLM $\boldsymbol{\beta}$ and σ_y, like the conjugate prior, is also a Multivariate-Normal-inverse-Gamma distribution.

$$H_y\left(\theta_{y,k}|\boldsymbol{y}_k\right) = \frac{G_y\left(\theta_{y,k}\right)\mathrm{L}_y\left(\boldsymbol{y}_k|\boldsymbol{x}_k, \theta_{y,k}\right)}{\int_{\theta_y} G_y(\theta_y)\mathrm{L}_y\left(\boldsymbol{y}_k|\boldsymbol{x}_k, \theta_y\right)} = \mathrm{N}\left(\boldsymbol{\beta}|\boldsymbol{m}_{n,y}, \sigma^2_y\boldsymbol{V}_{n,y}\right)\mathrm{Ga}^{-1}\left(\sigma^2_y|a_{n,y}, b_{n,y}\right)$$

It has the following, updated parameters: [67]

$$
\begin{aligned}
\boldsymbol{m}_{n,y} &= \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{V}_{0,y}\right)^{-1}\left(\boldsymbol{V}_{0,y}\boldsymbol{m}_{0,y} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right) \\
\boldsymbol{V}_{n,y} &= \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{V}_{0,y}\right) \\
a_{n,y} &= a_{0,y} + \frac{n_k}{2} \\
b_{n,y} &= b_{0,y} + \frac{1}{2}\left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{m}_{0,y}^{\mathrm{T}}\boldsymbol{V}_{0,y}\boldsymbol{m}_{0,y} - \boldsymbol{m}_{n,y}^{\mathrm{T}}\boldsymbol{V}_{n,y}\boldsymbol{m}_{n,y}\right)
\end{aligned}
$$

Here $\boldsymbol{X}$ and $\boldsymbol{y}$ are:

$$
\begin{aligned}
\boldsymbol{X} &= [\,\hat{\boldsymbol{x}}_1 \quad \dots \quad \hat{\boldsymbol{x}}_{n_k}\,]^T \qquad \text{with} \quad \hat{\boldsymbol{x}}_j = [\,1 \quad x_j\,] \quad \forall j \in \{1,\dots,n_k\} \\
\boldsymbol{y} &= [\,y_1 \quad \dots \quad y_{n_k}\,]^T
\end{aligned}
$$

Conditional posterior of cluster assignments z_i
The conditional posterior of cluster assignments takes the following form:

$$
\begin{aligned}
p\left(z_i = k, k \leq K\right) &\propto \frac{n_{k,-i}}{N + \alpha_0 - 1} F\left(x_i|\theta_{x,k}^{(t-1)}\right) F\left(y_i|x_i, \theta_{k,y}^{(t-1)}\right) \qquad n_{k,-i} = \sum_{j \neq i} \delta\left(z_j - k\right) \\
&\propto \frac{n_{k,-i}}{N + \alpha_0 - 1} \mathrm{N}\left(x_i|\mu_i, \sigma_{i,x}^2\right) \mathrm{N}\left(y_i|\boldsymbol{x}_n\boldsymbol{\beta}_i, \sigma_{i,y}^2\right) \\
p\left(z_i = K+1\right) &\propto \frac{\alpha_0}{N + \alpha_0 - 1} \int_{\theta_x}\int_{\theta_y} F\left(x_i|\theta_x\right) F\left(y_i|x_i, \theta_y\right) G_x\left(\theta_x\right) G_y\left(\theta_y\right) d\theta_x\, d\theta_y \\
&\propto \frac{\alpha_0}{N + \alpha_0 - 1} \int_{\theta_x} F\left(x_i|\theta_x\right) G_x\left(\theta_x\right) d\theta_x \int_{\theta_y} F\left(y_i|x_i, \theta_y\right) G_y\left(\theta_y\right) d\theta_y \\
&\propto \frac{\alpha_0}{N + \alpha_0 - 1} \int_{\mu}\int_{\sigma_x} \mathrm{N}\left(x_i|\mu, \sigma_x^2\right) \mathrm{N}\left(\mu|m_{0,x}, \sigma_x^2 V_{0,x}\right) \mathrm{Ga}^{-1}\left(\sigma_x^2|a_{0,x}, b_{0,x}\right) d\mu\, d\sigma_x \\
&\qquad \int_{\boldsymbol{\beta}}\int_{\sigma_y} \mathrm{N}\left(y_i|\boldsymbol{x}_n\boldsymbol{\beta}, \sigma_y^2\right) \mathrm{N}\left(\boldsymbol{\beta}|\boldsymbol{m}_y, \sigma_y^2\boldsymbol{V}_{0,y}\right) \mathrm{Ga}^{-1}\left(\sigma_y^2|a_{0,y}, b_{0,y}\right) d\boldsymbol{\beta}\, d\sigma_y \\
&\propto \frac{\alpha_0}{N + \alpha_0 - 1} \cdot \frac{1}{\pi^{N/2} 2^N} \cdot \frac{\left|V_{n,x}\right|^{\frac{1}{2}}}{\left|V_{0,x}\right|^{\frac{1}{2}}} \cdot \frac{b_{0,x}^{a_{0,x}}}{b_{n,x}^{a_{n,x}}} \cdot \frac{\Gamma\left(a_{n,x}\right)}{\Gamma\left(a_{0,x}\right)} \cdot \\
&\qquad \cdot \frac{1}{(2\pi)^{N/2}} \cdot \sqrt{\frac{\det\left(\boldsymbol{V}_{0,y}\right)}{\det\left(\boldsymbol{V}_{n,y}\right)}} \cdot \frac{b_{0,y}^{a_{0,y}}}{b_{n,y}^{a_{n,y}}} \cdot \frac{\Gamma\left(a_{n,y}\right)}{\Gamma\left(a_{0,y}\right)}
\end{aligned}
$$

In the last step we calculate the marginal likelihoods according to [67].

E Hyperparameters

General Settings

The following general settings are held constant for all experiments:

- Iterations of Gibbs sampler: 1000
- Maximal Iterations of VBEM: 500
- Convergence criterion of VBEM: We stop the algorithm for less than 1% change in the VLB.
- Size of the test data set: 1/5 of the training data set.

Implemented Prior

As mentioned in chapter 7 in our implementation we use the inverse Wishart prior instead of the Wishart prior from the derivations in chapter 4. This results in the following implemented conjugate prior:

$$\begin{aligned} p(\boldsymbol{z}, \nu, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{V}) &= p(\boldsymbol{z} \mid \boldsymbol{\pi})\, p(\nu)\, p(\boldsymbol{\mu} \mid \boldsymbol{\Sigma})\, p(\boldsymbol{\Sigma})\, p(\boldsymbol{\beta} \mid \boldsymbol{V})\, p(\boldsymbol{V}) \\ &= \prod_{n=1}^{N} \mathrm{Cat}(\boldsymbol{z}_n | \boldsymbol{\pi}) \prod_{k=1}^{\infty} \mathrm{Beta}\left(\nu_k | \gamma_{0,1}, \gamma_{0,2}\right) \prod_{k=1}^{\infty} \mathrm{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_0, \lambda_0^{-1} \boldsymbol{\Sigma}_k\right) \mathrm{Wi}^{-1}(\boldsymbol{\Sigma}_k | \boldsymbol{L}_0, \nu_0) \\ &\quad \prod_{k=1}^{\infty} \mathrm{N}(\boldsymbol{\beta}_k | \boldsymbol{M}_0, \boldsymbol{V}_k, \boldsymbol{K}_0)\, \mathrm{Wi}^{-1}(\boldsymbol{V}_k | \boldsymbol{P}_0, \eta_0) \end{aligned}$$

In comparison to the setting in section 4.4 the matrices $\boldsymbol{\Sigma}_k$ and $\boldsymbol{V}_k$ are not inverted.

Hyperparameters of the Implemented Prior

The following settings of the hyperparameters are chosen for all experiments. In each run of the experiment, there is a new random seed for drawing the hyperparameters from uniform distributions. m is the length of the input vector and d is the length of the output vector:

- Normal-inverse-Wishart prior:
  - $\boldsymbol{m}_0$: Drawn from a uniform distribution on the interval $\left[\boldsymbol{m}_{low}, \boldsymbol{m}_{high}\right)$. $\boldsymbol{m}_{low}$ and $\boldsymbol{m}_{high}$ represent the lowest and highest value of the input data on the training set.
  - λ_0: Drawn from a uniform distribution on the interval $[0, 0.1)$.
  - $\boldsymbol{L}_0$: Identity matrix with dimension m times a real value ψ_0. ψ_0 is drawn from a uniform distribution on the interval $[0, 10)$.
  - ν_0: $m + 1$.
- Matrix-Normal-inverse-Wishart prior:
  - $\boldsymbol{M}_0$: Zero matrix with dimensions d by $m + 1$.
  - $\boldsymbol{K}_0$: Diagonal matrix with dimension $m + 1$ by $m + 1$. All elements except of the last diagonal element are multiplied by a real value ψ_1. The last diagonal element, representing the variance of the offset, is multiplied by a real value ψ_2. ψ_1 is drawn from a uniform distribution on the interval $[0, 10)$. ψ_2 is drawn from a uniform distribution on the interval $[0, 1000)$.
  - $\boldsymbol{P}_0$: d by d identity matrix times a real value ψ_3. ψ_3 is drawn from a uniform distribution on the interval $[0, 0.1)$

- η_0: $d \cdot (m+1)+1$

- Stick-breaking prior or Dirichlet prior:
  - Dirichlet prior:
    - Number of components K: 100
  - Stick-breaking prior:
    - Truncation level of the variational distribution K: 100
    - Hyperparameter $\gamma_{0,1}$ of Beta distribution: 1