# Finite-Time Bounds for Sampling-Based Fitted Dynamic Programming

**Rémi Munos** remi.munos@inria.fr
*Sequential Learning project, INRIA Futurs Lille, France*

**Editor:**

**Joint work with Csaba Szepesvári and Andras Antos** (Munos and Szepesvári, 2006), (Munos, 2007), (Antos et al., 2006), (Antos et al., 2007)

## 1. Problem formulation

**We collected data:** $\{(X_n, A_n, R_n, Y_y)_{n\leq N}\}$

- Assume a generative model: $Y_n \sim p(\cdot|X_n, A_n)$
- Passively observe trajectories under behavior policy: $A_n \sim \pi_b(\cdot|X_n)$

**Properties:**

- **Setting:** Off policy batch learning with general function approximation $\mathcal{F}$ (state space is big or continuous)
- **What is known:** on-policy with linear function approximation works: TD (Tsitsiklis and Van Roy, 1997), LSTD (Bradtke and Barto, 1996), off-policy batch learning with "averagers" (Gordon, 1995) is stable.
- **Method:** fitted value iteration or policy iteration: Very popular! Ex: LSPI (Lagoudakis and Parr, 2003), Fitted Q iterations (Ernst et al., 2005) for recent algorithms.
- **Problem:** may diverge! (famous counter-examples (Baird, 1995; Tsitsiklis and Van Roy, 1996))
- **Goal:** Design a policy $\pi$ with near-optimal performance with high probability, ie. with proba $1-\delta$, provide a bound on $||V^* - V^\pi||$ in terms of the number of samples $N$, the capacity of the function space $\mathcal{F}$, $\delta$, ...

**Approximate Value Iteration**

$$V_{k+1} = \mathcal{P}TV_k,$$

where $\mathcal{P}$ = approximation operator (eg. projection, regression, supervised learning), $T$ = Bellman operator (i.e. $Tf(x) = \max_a[r(x,a) = \gamma \int f(y)P(dy|x,a)]$).

- **Propagation of error:** define $\pi_k$ = greedy policy wrt $V_k$. Write $\varepsilon_k = TV_k - \mathcal{P}TV_k$ the approximation error. Then (Bertsekas and Tsitsiklis, 1996):

$$\limsup_{k\to\infty} ||V^* - V^{\pi_k}||_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{k\to\infty} ||\varepsilon_k||_\infty \tag{1}$$

- **Each step = projection:**

$$V_{k+1} = \arg\min_{f\in\mathcal{F}} ||TV_k - f||$$

Statistical learning results for regression: Let $g$ a function, $\mu$ a distribution, such that some data $(x_n, y_n)_{n\leq N}$ are collected, with $x_n \sim \mu$, and $y_n$ is a noisy estimate of $g(x_n)$. Then the solution to the empirical least squares regression problem:

$$\arg\min_{f\in\mathcal{F}} \sum_n |f(x_n) - y_n|^2$$

is close to the projection of $g$ onto $\mathcal{F}$, ie. the solution to

$$\arg\min_{f\in\mathcal{F}} ||f - g||_{2,\mu},$$

when $N$ is large. Statistical learning theory uses capacity measures (VC dimension, metric entropy) of $\mathcal{F}$ to bound the difference between the empirical loss (learning error) and the functional loss (generalization error). The minimized error is defined in terms of $L_p$-norms (or variants) (eg. Neural Networks, linear regession, SVM, kernel methods, ...) but not with $L_\infty$ norm (except for "averagers" (Gordon, 1995) for which $||\mathcal{P}||_\infty \leq 1$).

- **Problem:** Dynamic Programming uses $L_\infty$ norms (property: $||T||_\infty < 1$) whereas Statistical Learning theory (and Approximation theory) uses $L_p$ norms (property: $||\mathcal{P}||_p \leq 1$). Thus the combined operator $\mathcal{P}T$ is neither a contraction in Ł$_\infty$ nor in $L_p$.

**Tools:**

- Statistical Learning: provide bound on each sampling-based Bellman iterate
- $L_p$-norm analysis in DP: for the propagation of error

## 2. $L_p$ analysis of AVI

We have:

$$\text{Under A1, } \limsup_{k\to\infty} ||V^* - V^{\pi_k}||_\infty \leq \frac{2\gamma}{(1-\gamma)^2} C(\mu)^{1/p} \limsup_{k\to\infty} ||\varepsilon_k||_{p,\mu} \quad (2)$$

$$\text{Under A2, } \limsup_{k\to\infty} ||V^* - V^{\pi_k}||_{p,\rho} \leq \frac{2\gamma}{(1-\gamma)^2} C(\rho,\mu)^{1/p} \limsup_{k\to\infty} ||\varepsilon_k||_{p,\mu} \quad (3)$$

where assumption **A1 [concentration of the transition kernel]** says that there exists $C(\mu) < \infty$ such that, for all $x, a$,

$$P(dy|x,a) \leq C(\mu)\mu(dy)$$

(Example: if $\mu$ is the uniform measure then this assumption says that the transition probability kernel $P(\cdot|x,a)$ admits a uniformly bounded density).

And assumption **A2 [concentration of the discounted future-state distributions]** assumes that there exists $C(\rho,\mu) < \infty$, for all $x, a$,

$$(1-\gamma)^2 \sum_{t\geq 1} t\gamma^{t-1} \mathbb{P}\big(X_t \in dy | X_0 \sim \rho, A_0, A_1, \cdots\big) \leq C(\rho,\mu)\mu(dy).$$

**Property:** Dynamics which fail to satisfy these assumptions would probably preclude any sample-based estimation of the performance.

**Where do these $L_p$ bounds come from?** Assume that for two positive vectors $u$ and $v$ are such that $u \leq Pv$, with $P$ a stochastic matrix. Of course we deduce that $||u||_\infty \leq ||v||_\infty$, but moreover if $\rho$ and $\mu$ are probability distributions such that componentwise $\rho P \leq C\mu$, with $C \geq 1$ a constant, then we deduce that

$$||u||_{p,\rho} \leq C^{1/p} ||v||_{p,\mu}.$$

Indeed we have

$$\begin{aligned} ||u||_{p,\rho}^p = \int_{x\in X} \rho(dx)|u(x)|^p &\leq \int_{x\in X} \rho(dx)\big[\int_{y\in X} P(dy|x)v(y)\big]^p \\ &\leq \int_{x\in X} \rho(dx) \int_{y\in X} P(dy|x)v(y)^p \\ &\leq C \int_{y\in X} \mu(dy)v(y)^p = C||v||_{p,\mu}^p, \end{aligned}$$

using Jensen's inequality.

Well, in AVI, this is the case. We may prove that we have the following componentwise bound:

$$\limsup_{k\to\infty} V^* - V^{\pi_k} \leq \limsup_{k\to\infty} (I - \gamma P^{\pi_k})^{-1} \Big(\sum_{l=0}^{k-1} \gamma^{k-l} \big[(P^{\pi^*})^{k-l} + P^{\pi_k} P^{\pi_{k-1}} \dots P^{\pi_{l+2}} P^{\pi_{l+1}}\big] |\varepsilon_l| \Big), \tag{4}$$

which implies both the $L_\infty$ bound (1) and the $L_p$ bounds (2) and (3).

**Extension to other approximate DP methods:** The $L_\infty$ bounds for **Approximate policy iteration**:

$$\limsup_{k\to\infty} ||V^* - V^{\pi_k}||_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{k\to\infty} ||V_k - V^{\pi_k}||_\infty$$

and the **Bellman residual** bound:

$$||V^* - V^\pi||_\infty \leq \frac{2}{1-\gamma} ||TV - V||_\infty$$

have their counterpart $L_p$ bounds too!

## 3. Finite sample bound on each AVI iteration

**Two sources of error:**

- The Bellman operator $T$ (which makes use of an expectation over next states) needs to be estimated from samples (sampling-based Bellman operator: $\hat{T}$)
- The projection (approximation) operators uses samples (sampling-based projection: $\hat{\mathcal{P}}$).

**A single update:** Draw $N$ points $(X_i \sim \mu)_{i \le N}$, and then from each of those points, for all possible actions, draw $M$ samples of observed rewards $(R_j^{i,a})_{j \le M}$ and next states $(Y_j^{i,a} \sim P(\cdot|X_i, a))_{j \le M}$ using generative model.

Then define the sampling based Bellman backed up values:

$$\hat{V}(X_i) = \max_a \frac{1}{M} \Big[ \sum_j [R_j^{i,a} + \gamma V(Y_j^{i,a})] \Big]$$

and the sampling-based projection onto $\mathcal{F}$:

$$\hat{\mathcal{P}}\hat{T}V = \arg\min_{f \in \mathcal{F}} \sum_i |f(X_i) - \hat{T}V(X_i)|^p. \tag{5}$$

**Sample bound:** We have (neglecting $\log N$ terms) with probability $1 - \delta$,

$$||\hat{\mathcal{P}}\hat{T}V - TV||_{p,\mu} \le d(TV, \mathcal{F}) + O\Big\{ \Big( \frac{V_{\mathcal{F}} \log \delta^{-1}}{N} \Big)^{1/2p} + \Big( \frac{\log \delta^{-1}}{M} \Big)^{1/2} \Big\}$$

where $d(TV, \mathcal{F}) = \inf_{f \in \mathcal{F}} ||TV - f||_{p,\mu}$ and $V_{\mathcal{F}}$ is a capacity measure of $\mathcal{F}$ (pseudo-dimension).

## 4. AVI: Putting things together

Sampling-based fitted VI: repeat $K$ times the previous update (5): $V_{k+1} = \hat{\mathcal{P}}\hat{T}V_k$ (where either using the same set of samples throughout all iterations or regenerate a fresh set at each iteration). Then, with probability $1 - \delta$, we have:

$$||V^* - V^{\pi_K}||_\infty \le \frac{2\gamma}{(1-\gamma)^2} C(\rho, \mu)^{1/p} \Big[ d(T\mathcal{F}, \mathcal{F}) + O\Big\{ \Big( \frac{V_{\mathcal{F}} \log \delta^{-1}}{N} \Big)^{1/2p} + \Big( \frac{\log \delta^{-1}}{M} \Big)^{1/2} \Big\} \Big] + O(\gamma^K)$$

where $d(T\mathcal{F}, \mathcal{F}) = \sup_{g \in \mathcal{F}} \inf_{f \in \mathcal{F}} ||Tg - f||_{p,\mu}$ is the inherent Bellman residual of $\mathcal{F}$.

**Analysis of this result:**

- This explains the counter examples of (Baird, 1995; Tsitsiklis and Van Roy, 1996) for which $d(T\mathcal{F}, \mathcal{F}) = \infty$
- Question: if the space $\mathcal{F}$ grows, does $d(T\mathcal{F}, \mathcal{F})$ decrease?
- Answer: Yes! if the MDP is smooth (ie. $P(dy|\cdot, a)$ and $r(\cdot, a)$ are Lipschitz)
- Thus fitted policy iteration is a sound method!
- **Bias-variance tradeoff**: when $\mathcal{F}$ grows, the approximation error $d(T\mathcal{F}, \mathcal{F})$ decreases (**bias term**) but the estimation error $O((V_{\mathcal{F}}/N)^{1/2p})$ (**variance term**) increases, but may be made smaller by using more samples (to avoid overfitting).

**Numerical experiment:** This is an optimal replacement problem (see e.g. (Rust, 1996)). We consider approximation of the value function using polynomials of degree $l$.

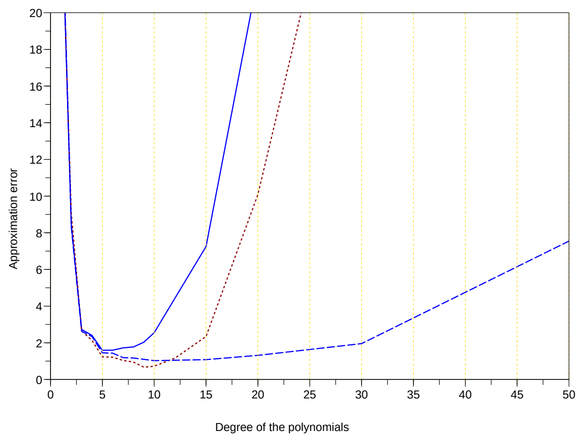


Figure 1: Approximation errors $||V^* - V_K||_\infty$ of the function $V_K$ returned by sampling-based FVI after $K = 20$ iterations, for different values of the polynomials degree $l$, for $N = 100$, $M = 10$ (plain curve), $N = 100$, $M = 100$ (dot curve), and $N = 1000$, $M = 10$ (dash curve) samples. The plotted values are the average over 100 independent runs.

## 5. What about if we must follow a fixed policy?

We observe a plant under control (behavior policy $\pi_b$) and from those collected data $(X_t, A_t \sim \pi_b(\cdot|X_n), R_t, X_{t+1} \sim P(\cdot|X_t, A_t), \dots)$, can we design a near-optimal policy?

**What are the additional assumptions?**

- **Exploration**: The behavior policy $\pi_b > 0$ and the MDP following $\pi_b$ is stationary: $X_t \sim \mu$,
- Since the samples are correlated, we need a forgetting property of the process. We assume the Markov chain $(X_t)$ is $\beta$-**mixing** with exponential rate: $\sup_{t\geq 1} \big|\mathbb{P}(X_{t+m} \in B|X_1, \cdots, X_t) - \mathbb{P}(X_{t+m} \in B)\big|\big| \leq O(e^{-bm^\kappa})$ (i.e. future depends weakly on the past)

Let us use **fitted policy iteration** (fitted Q iteration should work also...)

### 5.1 Policy evaluation by Bellman residual minimization

Here we use Q-functions instead of value functions.

Repeat $K$ policy iteration steps, where at each iteration $k$, we find a approximation $Q_k \in \mathcal{F}$ of $V^{\pi_k}$ that minimizes the norm of the Bellman residual:

$$\arg\min_{Q\in\mathcal{F}} ||Q - T^{\pi_k}Q||_{2,\mu}. \tag{6}$$

However we need to be careful when writting a sampling-based version of that problem. Indeed, the solution to

$$\arg\min_{Q\in\mathcal{F}} \sum_t |Q(X_t, A_t) - [R_t + \gamma Q(X_{t+1}, \pi_k(X_{t+1}))]|^2 \quad (7)$$

is not consistent with the solution to (6) (see eg. (Sutton and Barto, 1987), (Munos, 2003), (Lagoudakis and Parr, 2003)): the sampling-based estimate is biased.

Indeed, defining the function $h(x, a, y) = r(x, a) + \gamma Q(y, \pi_k(y))$, the problem (6) minimizes $\mathbb{E}_{(X,A)\sim\mu}[(Q(X, A) - (\mathbb{E}_{Y\sim P(\cdot|X,A)}[h(X, A, Y)])^2]$ whereas (7) is a sampled-based minimization of $\mathbb{E}_{(X,A)\sim\mu, Y\sim P(\cdot|X,A)}[(Q(X, A) - h(X, A, Y))^2]$. The difference between these quantities is the variance of $h(x, a, Y)$, ie:

$$\mathbb{E}[(Q(x, a) - h(x, a, Y))^2] - [Q(x, a) - (\mathbb{E}[h(x, a, Y)])^2] = \mathbb{V}ar\,[h(x, a, Y)]\,.$$

Thus we defined the **modified Bellman residual empirical minimization problem**:

$$\arg\min_{Q\in\mathcal{F}} \Big\{ \sum_t |Q(X_t, A_t) - [R_t + \gamma Q(X_{t+1}, \pi_k(X_{t+1}))]|^2 - \arg\min_{g\in\mathcal{F}} \sum_t |g(X_t, A_t) - [R_t + \gamma Q(X_{t+1}, \pi_k(X_{t+1}))]|^2 \Big\}$$

which is a unbiased estimate and yields a solution consistent to the solution of (6)

**Linear approximation space** In case $\mathcal{F}$ is linear, then the modified Bellman residual problem is nothing else than LSPI (Lagoudakis and Parr, 2003) which provides a finite-time performance bound for this algorithm.

**Result:** High probability bound on the performance loss in terms of the number of samples $N$ and of iterations $K$: Under A1, with probability $1-\delta$, we have (neglecting $\log N$ terms):

$$||V^* - V^{\pi_K}||_\infty \le \frac{2\gamma}{(1-\gamma)^3}\sqrt{C(\mu)}\Big[d(\mathcal{F}, T\mathcal{F}) + O\Big(\big(\frac{[V_{\mathcal{F}} + \log(1/\delta)]^{1+1/\kappa}}{N}\big)^{1/4}\Big)\Big] + O(\gamma^K), \quad (8)$$

where: $d(\mathcal{F}, T\mathcal{F}) = \sup_{g\in\mathcal{F}} \sup_\pi \inf_{f\in\mathcal{F}} ||T^\pi g - f||_\mu$ is the inherent Bellman error of $\mathcal{F}$, and $V_{\mathcal{F}}$ is a capacity measure of $\mathcal{F}$ which depends on the pseudo-dimension and the VC-crossing dimension of $\mathcal{F}$ (i.e. VC-dimension of $\{\{x \in X, f(x) \ge g(x)\}, f, g \in \mathcal{F}\}$ ).

### 5.2 Policy evaluation by FVI:

Repeat K times:

- **Fitted policy evaluation step:** find an approximation of $Q^{\pi_k}$ by repeating $M$ steps of fitted value iteration:
  - Define $I_k \stackrel{\text{def}}{=} \{t \in [1, N],\, A_t = \pi_k(X_t)\}$.
  - Define $(Q_k^m)_{0\le m\le M}$ by: for $0 \le m < M$,

$$\begin{cases} v_t^m \stackrel{\text{def}}{=} R_t + \gamma Q_k^m(X_{t+1}, \pi_k(X_{t+1})), \text{ for all } t \in I_k \\ Q_k^{m+1} \stackrel{\text{def}}{=} \arg\min_{f\in\mathcal{F}} \sum_{t\in I_k} \big[f(X_t, A_t) - v_t^m\big]^2 \end{cases}$$

- **Policy improvement step**: define the new policy $\pi_{k+1}$ by:

$$\pi_{k+1}(x) \stackrel{\text{def}}{=} \arg\max_a Q_k^M(x,a)$$

Return policy $\pi_K$.

**Results:** Similar to (8) with $O(\gamma^{\min(K,M)})$ instead of $O(\gamma^K)$.

## References

A. Antos, Cs. Szepesvári, and R. Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. In *COLT-19*, pages 574–588, 2006.

A. Antos, Cs. Szepesvári, and R. Munos. Value-iteration based fitted policy iteration: learning with a single trajectory. In *2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL 2007)*, pages 330–337. IEEE, April 2007. (Honolulu, Hawaii, Apr 1–5, 2007.).

L. Baird. Residual algorithms: Reinforcement learning with function approximation. In Armand Prieditis and Stuart Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 30–37, San Francisco, CA, 1995. Morgan Kaufmann.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

S.J. Bradtke and A.G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.

D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

G.J. Gordon. Stable function approximation in dynamic programming. In Armand Prieditis and Stuart Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 261–268, San Francisco, CA, 1995. Morgan Kaufmann.

M. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

R. Munos. Error bounds for approximate policy iteration. In *19th International Conference on Machine Learning*, pages 560–567, 2003.

R. Munos and Cs. Szepesvári. Finite time bounds for sampling based fitted value iteration. Technical report, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Kende u. 13-17, Budapest 1111, Hungary, 2006.

Rémi Munos. Performance bounds in lp norms for approximate value iteration. *SIAM J. Control and Optimization*, 2007.

J. Rust. Numerical dyanmic programming in economics. In H. Amman, D. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*. Elsevier, North Holland, 1996.

R.S. Sutton and A.G. Barto. Toward a modern theory of adaptive networks: Expectation and prediction. In *Proc. of the Ninth Annual Conference of Cognitive Science Society*. Erlbaum, Hillsdale, NJ, USA, 1987.

J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22:59–94, 1996.

J.N. Tsitsiklis and B. Van Roy. An analysis of temporal difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42:674–690, 1997.