Least-Angle Temporal Difference Learning: LATD(λ)

Manuel Loth*, Rémi Coulom*, Manuel Davy†, Philippe Preux*

28 avril 2006

Résumé

Les méthodes à noyaux suscitent depuis quelques années un vif intérêt dans la communauté de l'apprentissage automatique. L'apprentissage par renforcement a cependant connu peu de propositions d'utilisation de ces méthodes. Nous présentons ici nos travaux en cours sur une extension du TD(λ) aux noyaux, par l'emploi de la régression équi-angulaire, issue de la sélection de variables. Cette méthode offre de multiples avantages : noyaux multiples, robustesse, parcimonie, absence d'hyper-paramètres, complexité linéaire, qui lui permettent de surpasser les méthodes existantes sur nos premières expérimentations.

1 TD(λ)

Considérons une sous-famille des processus de décision de Markov : Un systême est décrit par un vecteur d'état. L'espace des états est continu, et le temps est discrétisé. Un nombre fini d'actions applicables à chaque pas de temps permettent de passer d'un état à l'autre. Cette dynamique est déterministe et connue. Un but est fixé en associant à chaque couple (état,action) une récompense. On souhaite apprendre la politique (choix de l'action en fonction de l'état courant) qui maximise une somme à long terme des récompenses : $\sum_{t=t_0}^{\infty} \gamma^{t-t_0} r_t, \gamma \in]0;1]$.
La fonction valeur V d'une politique π associe à chaque état cette somme. Si π fait passer de x à x' avec une récompense r, $V(x) = r + \gamma V(x')$.
TD(λ)[3] estime la valeur de la politique optimale en appliquant une politique basée sur l'estimation de sa propre valeur, et en corrigeant régulièrement cette estimation ($\hat{V}$) en fonction des erreurs (différences temporelles) constatées : $td = r - (\hat{V}(x) - \gamma \hat{V}(x'))$. Par ex., on part d'un état aléatoire et applique une politique gloutonne sur $\hat{V}$ pendant un temps donné (épisode). Les différences temporelles constatées en chaque état sont imputées de façon graduelle aux erreurs d'estimation de $\hat{V}$ sur les états précédents, et on effectue un pas de descente de gradient sur les coefficients de l'approximateur :

*LIFL, UMR CNRS, Villeneuve d'Ascq, loth@lifl.fr, remi.coulom@univ-lille3.fr, philippe.preux@univ-lille3.fr
†LAGIS, UMR CNRS, Villeneuve d'Ascq, manuel.davy@ec-lille.fr

Si on estime V par $\hat{V}(x) = \sum \beta_i \phi_i(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}$, on obtient finalement la mise à jour suivante après chaque épisode

$$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \alpha \boldsymbol{\Phi} \mathbf{L} (\mathbf{r} - \mathbf{H}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\beta})$$

où
- $\boldsymbol{\Phi} = (\ldots \boldsymbol{\phi}(x_i) \ldots)^{\mathsf{T}}$
- $\mathbf{r} - \mathbf{H}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\beta}$ est le vecteur des différences temporelles,
- $\mathbf{L}$ répartit celles-ci sur les états antérieurs

2 Méthodes à noyaux

Nous considérons les méthodes à noyaux au sens le plus large : estimation d'une fonction par une combinaison linéaire de features, ceux-ci étant définis par un fonction duale k en associant à chaque point x_i de l'ensemble d'apprentissage $\phi_i = k(xi, \cdot)$
Nous considérons le cas général ou on souhaite estimer une fonction par $\sum \beta_i \phi_i(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}$ où les features sont surabondants.
Le problème à résoudre est de fournir une bonne approximation de la cible en utilisant un nombre restreint de features, ie en laissant (ou mettant) un maximum de poids à 0 (principe de parcimonie).

2.1 Least-angle regression : ϕ-LARS

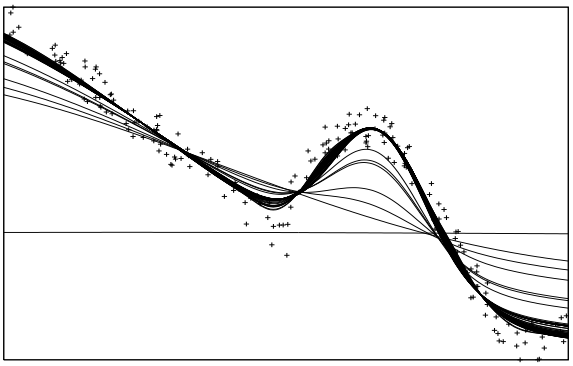
Une stratégie est de minimiser la fonction coût régularisée $\|y - \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\beta}\|^2 + \mu \sum |\boldsymbol{\beta}_i|$, ce qui revient à minimiser $\|y - \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\beta}\|^2$ sous la contrainte $\sum |\boldsymbol{\beta}_i| < t$ (2.1)
LARS est une méthode de sélection de variables qui s'étend naturellement à la sélection de features [2].
Qualifions d'actifs les features de coefficents non nuls et notons $\mathcal{A}$ l'ensemble de ceux-ci. A chaque pas, un nouveau feature est activé ou désactivé et les poids des features actifs mis à jour, et on obtient la solution de (2.1) pour un certain t, t croissant à chaque pas.
Le principe est le suivant : Pour minimiser $\|\mathbf{y} - \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\beta}\|^2$, on en annule la dérivée $\boldsymbol{\Phi}(\mathbf{y} - \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\beta})$, que l'on peut voir comme le vecteur des corrélations des features avec le résidu $(y - \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\beta})$.
Sous des conditions de normalisation, les corrélations sont comparables et on opère alors :

$\mathcal{A} \leftarrow \{\text{argmax}(\boldsymbol{\phi}_i(\mathbf{y} - \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\beta}))\}$
invariant : features actifs équi-corrélés au résidu
répéter
- trouver la direction équi-angulaire sur $\mathcal{A}$

– modifier les poids dans cette direction jusqu'à ce qu'un feature inactif ϕ_j soit autant corrélé que les actifs ou que le poids d'un feature actif ϕ_k s'annule.
– $\mathcal{A} \leftarrow \mathcal{A} \cup \{j\}$ ou $\mathcal{A} - \{k\}$

jusqu'à [critère de parcimonie/qualité]

L'algorithme est (pour chaque pas) linéaire dans le nombre de features candidats et quadratique dans le nombre de features actifs, qui évolue de 0 à un nombre typiquement petit.
La méthode a entre autres avantages de pouvoir associer plusieurs features à chaque point (par ex. des gaussiennes de différentes largeurs de bande).



exemple d'estimations successives. La dernière utilise 10 features

3 Least-Angle TD(λ)

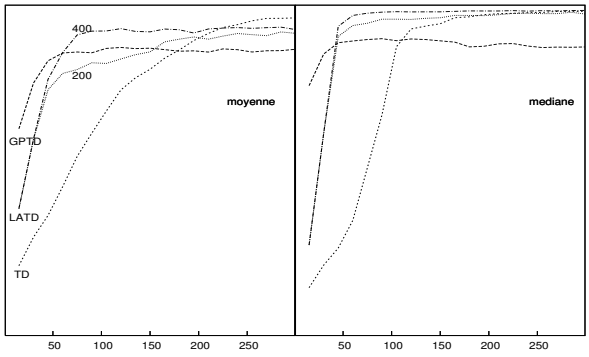
Dans les mises à jour du TD(λ), on peut de même voir $\mathbf{r_{es}} = \mathbf{L}(\mathbf{r} - \mathbf{H}\mathbf{\Phi}^\mathsf{T}\boldsymbol{\beta})$ comme le résidu de l'épisode et $\mathbf{\Phi r_{es}}$ comme le vecteur des corrélations de chaque feature à celui-ci.
Nous proposons de réduire ce résidu par l'ajout de features et de poids associés, sélectionnés par un Least-Angle. Les features candidats sont
– l'application d'un certain nombre de noyaux aux états de l'épisode :
$k_1(x_1, \cdot), k_2(x_1, \cdot), \ldots, k_1(x_i, \cdot), \ldots$
– les features déjà inclus dans l'estimation. Ceux-ci conduisent à une mise à jour des poids existants.

Diverses considérations amènent à prendre pour critère d'arrêt la réduction du résidu d'un pourcentage donné (20%).

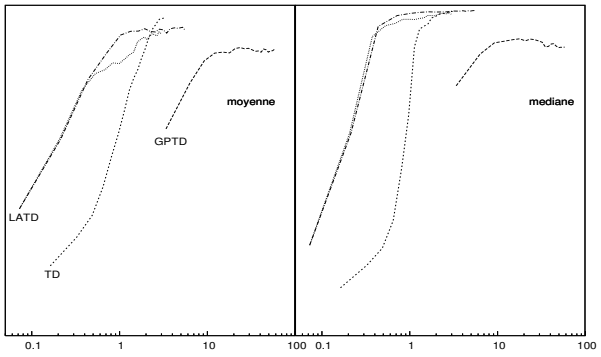
4 Expérimentations

Nous avons comparé les résultats obtenus sur le problème du pendule inversé par
– TD(λ) sur une grille 14×14 de features gaussiens.
– GPTD (processus gaussiens [1]) sur un noyau gaussien avec un nombre maximum de bases actives de 200.
– LATD(λ) sur 3 noyaux gaussiens avec un nombre maximum de bases actives de 200 et 400.

Pour chaque méthode, on a effectué une série de 100 apprentissages sur 300 épisodes de $4s$. La qualité de l'estimation a été evaluée au cours de l'apprentissage en effectuant 100 épisodes hors apprentissage partant de points répartis en grille sur l'espace d'états, et en additionnant les récompenses obtenues. La suite des points de départs des épisodes durant l'apprentissage est la même pour chaque méthode.



Qualité / épisode



Qualité / temps (échelle logarithmique)

On constate que les deux méthodes à noyaux convergent plus rapidement en terme d'épisodes. Contrairement à GPTD, la qualité obtenue par LATD est plus constante : on obtient une grosse majorité de bonnes solutions et quelques échecs, alors que la qualité atteinte par GPTD est très variable.
En terme de temps de calcul, LATD est, contrairement à GPTD, quasi-linéaire.
Notre algorithme a donc tendance à réunir le meilleur des noyaux et de TD(λ) : convergence rapide et faible complexité.

Références

[1] Y. Engel, S. Mannor, and R. Meir. Gaussian process with reinforcement learning. In *Proc. of the Int'l Conf. on Machine Learning (ICML)*, 2005.

[2] V. Guigue. *Méthodes à noyaux pour la représentation et la discrimination de signaux non-stationnaires.* PhD thesis, Institut National des Sciences Appliquées de Rouen, 2005.

[3] R.S. Sutton and A.G. Barto. *Reinforcement learning : an introduction.* MIT Press, 1998.