Apprentissage statistique et Big Data, focus sur l'algorithme online-EM

Olivier Cappé

Laboratoire Traitement et Communication de l'Information
CNRS, Télécom ParisTech, 75013 Paris

8 octobre 2013

L'apprentissage automatique

Apprentissage automatique *(machine learning)*

Apprendre à effectuer des tâches à partir d'exemples

De façon supervisée Avec des exemples de données et de résultats souhaités
↪ Classification

De façon non supervisée Avec uniquement des exemples de données
↪ Clustering

De façon séquentielle En traitant un flux de données de façon causale
↪ Prédiction séquentielle

Par extension, problèmes dont le traitement présente des analogies avec ce qui précède
↪ Régression en grande dimension

Quelques idées sur l'apprentissage statistique

Je m'intéresse aux **approches statistiques** de l'apprentissage

1. Comme en statistique usuelle j'utilise un **modèle probabiliste des données**

   par exemple, $Y = f(x)\beta + \epsilon$, avec ϵ aléatoire

2. Mais ce qui m'intéresse c'est **prédire** | **classifier** | **reconstruire** plus qu'estimer

   prédire Y' par $f(x')\hat{\beta}$ pour un nouvel x'

   plus que $\hat{\beta}$ en lui même

Quelques idées sur l'apprentissage statistique ...

3. D'autant plus que je crois que le **modèle** utilisé est **instrumental** (probablement faux)

$$\text{on doit avoir } Y = \mathcal{F}(x) + \epsilon, \text{ avec } \mathcal{F} \text{ assez compliquée}$$

4. De ce fait, j'ai intérêt à faire **croître la complexité du modèle avec le nombre de données disponibles**

$$\text{si j'observe } Y_1, \dots, Y_n, \text{ j'utilise plutôt le modèle}$$

$$Y_i = \sum_{j=1}^{k_n} f_j(x_i)\beta_j + \epsilon_i, \text{ où } k_n \uparrow \text{ avec } n$$

5. Le **choix / sélection de modèle** est crucial

$$\text{choisir } \hat{k}_n \text{ pour que } \sum_{j=1}^{\hat{k}_n} f_j(x')\hat{\beta}_j \text{ soit le plus proche de } Y'$$

Quelques idées sur l'apprentissage statistique . . . . . .

6 L'**optimisation** est un outil central

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \|Y - x\beta\|^2 \text{ plutôt que } \hat{\beta} = (x^T x)^{-1} x^T Y$$

de façon à penser à $\underset{\beta:\|\beta\|_1 \leq \lambda}{\operatorname{argmin}} \|Y - x\beta\|^2$

qu'il va falloir **résoudre numériquement**. . .

7 Comme les modèles sont faux j'ai intérêt à les **combiner**

J'estime $\hat{\beta}_1, \ldots, \hat{\beta}_k$ séparément dans les

modèles $Y = f_1(x)\beta_1 + \epsilon$, $\ldots$, $Y = f_k(x)\beta_k + \epsilon$,

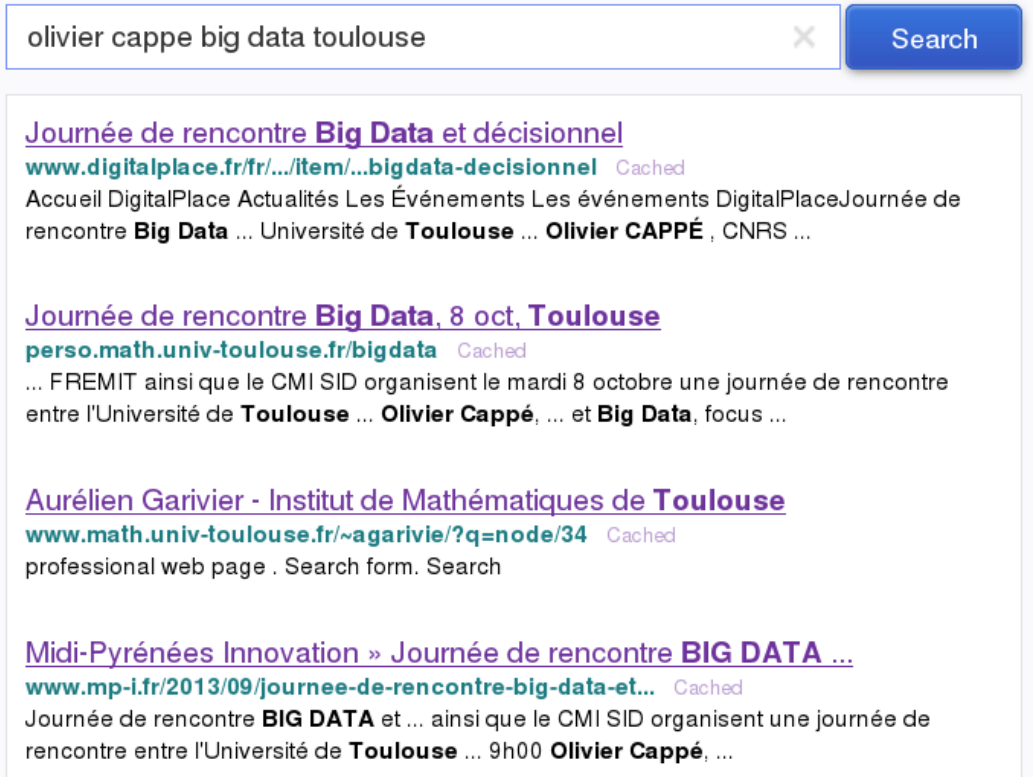
j'utilise l'ensemble $\{f_1(x')\hat{\beta}_1, \ldots, f_k(x')\hat{\beta}_k\}$ pour prédire Y'

Quelques idées sur l'apprentissage statistique .....

8 L'approche bayésienne permet de retrouver beaucoup des méthodes précédentes $(k_n \uparrow n,\ \|\beta\|_1 \leq \lambda,\ \{f_1(x')\hat{\beta}_1, \ldots, f_k(x')\hat{\beta}_k\})$ ainsi que certaines de leurs variantes

J'imagine que β (voire k) est une variable aléatoire

- Il est trop tard pour s'interroger sur l'opportunité du Big* Data...
- Néanmoins son potentiel reste assez largement spéculatif



* Suffisamment grand pour espérer répondre à des questions complexes

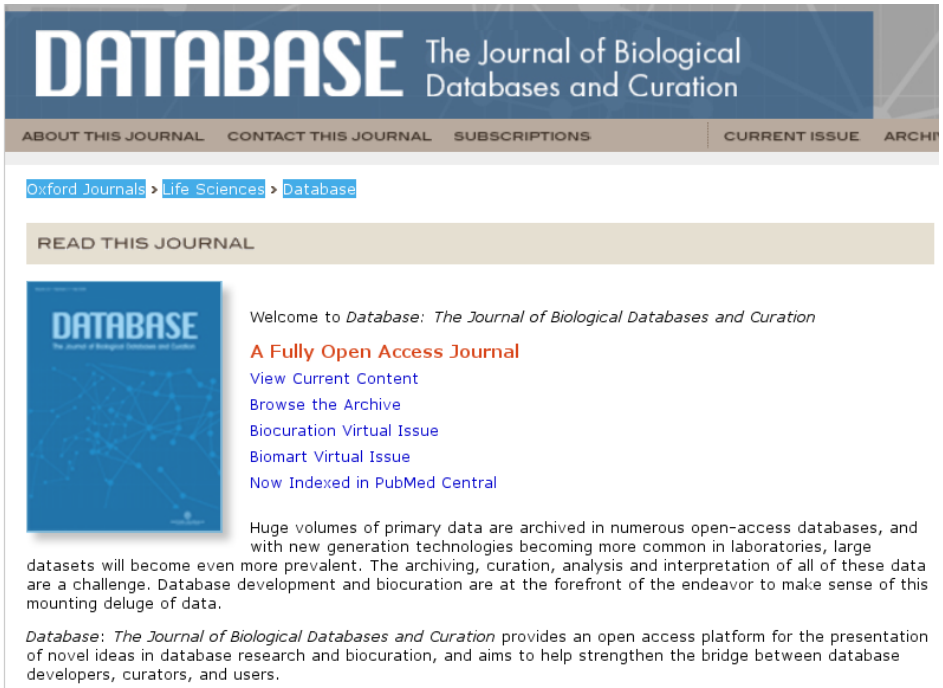
Un phénomène qui impacte aussi la science

Big Data

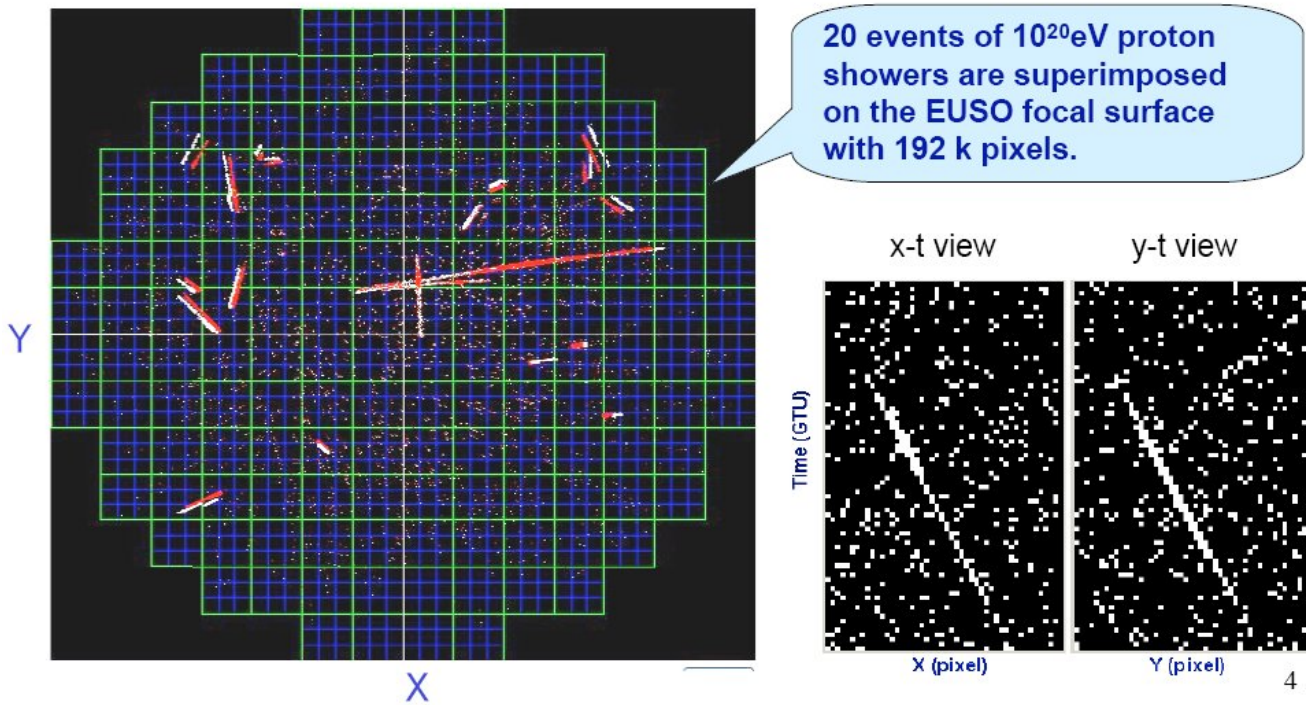
Web, réseaux sociaux, commerce en ligne, systèmes d'information d'entreprise. . .

mais aussi

Sciences du vivant, sciences du climat, physique des particules, sciences humaines, . . .



Vers une "science des données" ?



Expérience JEM-EUSO Capteur de 2.10^5 pixels / 2.5μs pour détecter une centaine de particules de très haute énergie par jour (source Balázs Kégl, LAL/LRI)

$\hookrightarrow$ détecteur obtenu par des méthodes d'apprentissage statistique (boosting) à partir de données préliminaires et de simulations

Une demande très forte sur l'enseignement

Qui mêle des compétences en mathématiques appliquées (statistiques, optimisation) et en informatique

	A		B	
	A1	A2	B1	B2
P1	Concepts Fondamentaux de la Sécurité (30H, 3ECTS) INF721	Statistique (30H, 3ECTS) MDI 220	Bases de Données (30H, 3ECTS) INF225	Economie de l'Internet et des Données Personnelles (20H, 3ECTS) SES720
P2	L'Ecosystème du Big Data (20H, 3ECTS) SES721	Visualisation d'Information (30H, 3ECTS) INF229	Bases de Données Avancées (60H, 5ECTS) INF345	
P3	Systèmes Répartis (60H, 5ECTS) INF346		Machine-Learning (60H, 5ECTS) MDI 343	
P4	Machine-Learning Avancé (60H, 5ECTS) INFMDI341		Données du Web (60H, 5ECTS) INF344	

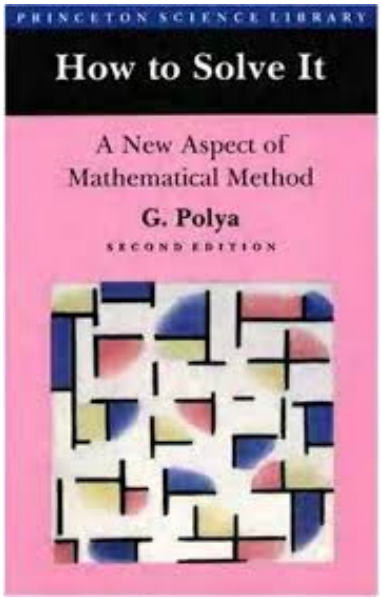
L'exemple du Master spécialisé Big Data à Télécom ParisTech (Stephan Clémençon)

Modélisation et Big Data

Il est nécessaire (et utile) de **modéliser** le Big Data

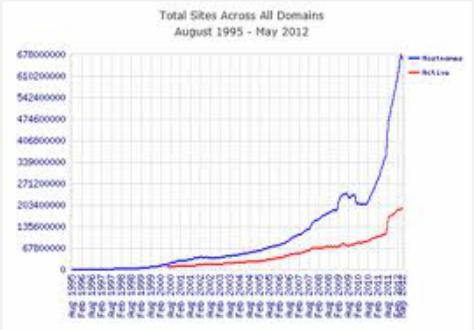
If you cannot solve the proposed problem, try to solve first some related problem. Could you imagine a more accessible related problem?

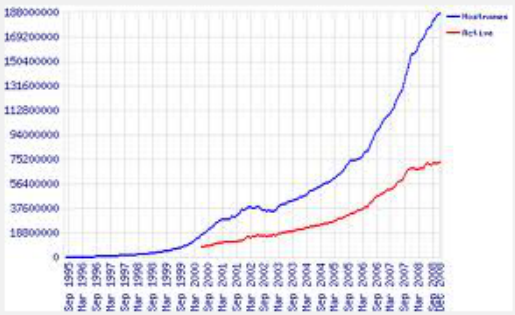
George Pólya,
How to Solve It (1945)

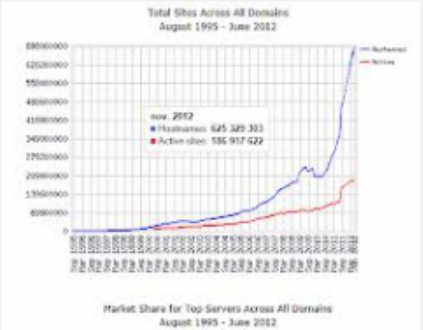


Grand n, grand p

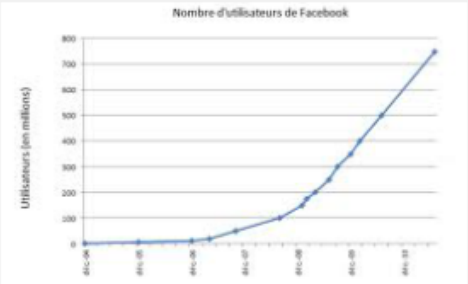
Plus de données





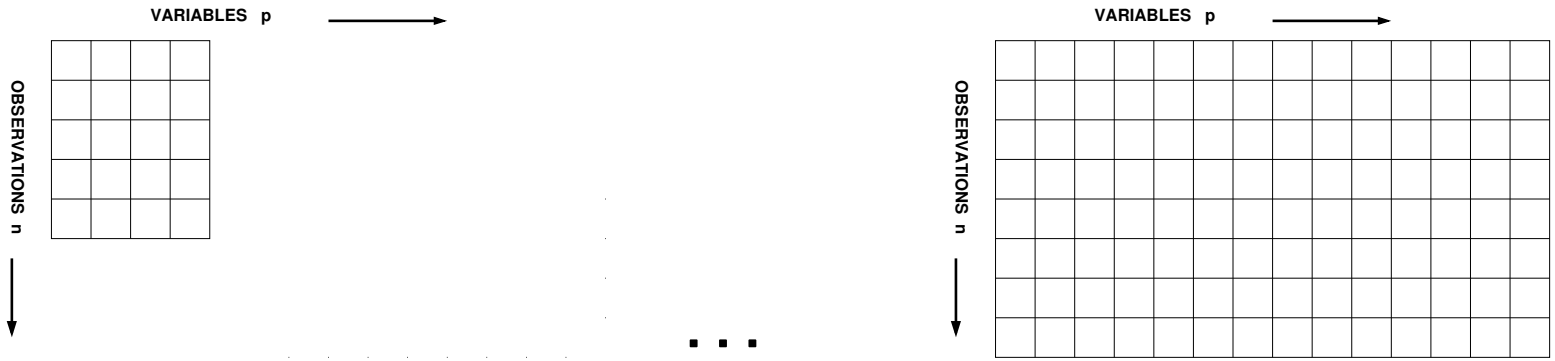


mais aussi plus de statistiques!



Grand n, grand p

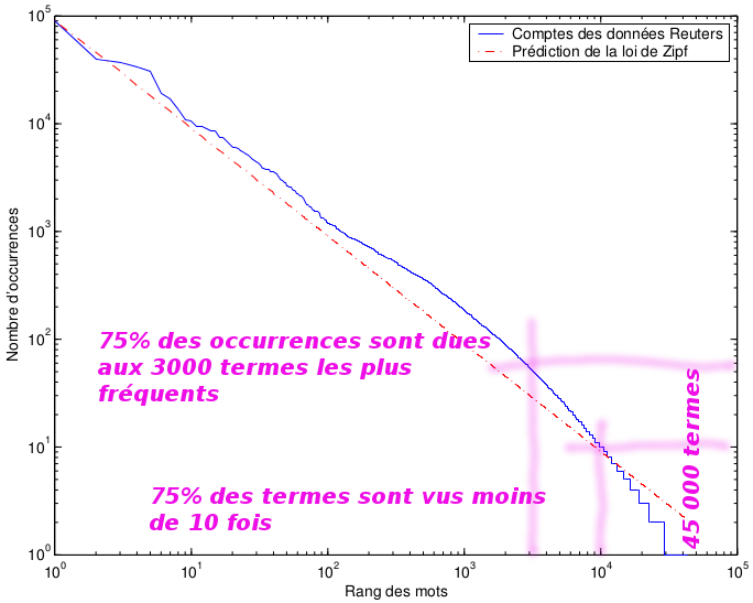
On imagine une séquence d'expériences dans lesquelles les nombres d'observations et de variables explicatives augmentent simultanément



Pour modéliser certaines situations (expérience "Microarray" en génomique), il est possible de considérer aussi des cas où $p > n$

“Lois” de Zipf

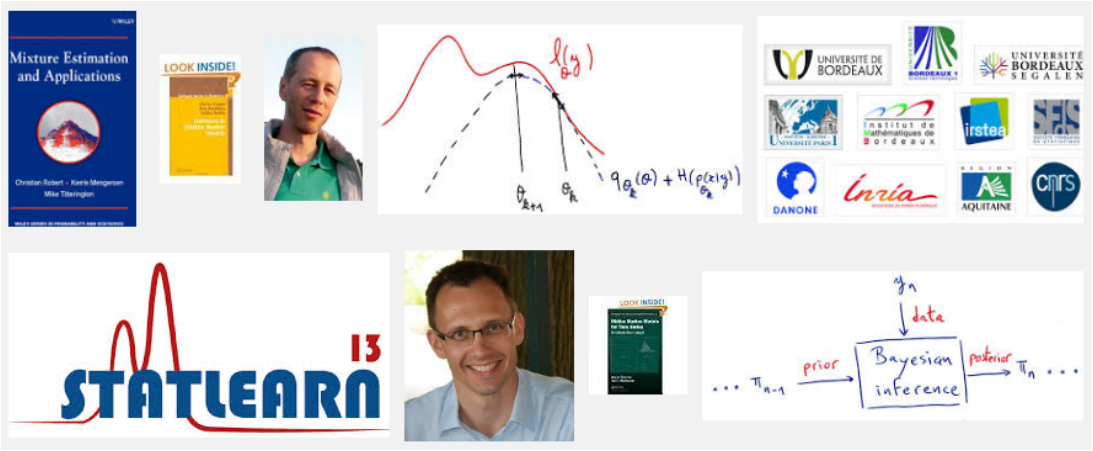
Rare n’est pas synonyme d’inintéressant



- Comment échantillonner les données ?
- Peut on construire (avec une complexité acceptable) des résumés exhaustifs des données ?

Contrôle du taux de faux positifs

Q: `olivier cappe big data`



Q: `olivier cappe big data toulouse`

Compromis performance / robustesse / calcul

Le volume de données nécessite de réduire la complexité des traitements

Idées : Algorithmes sous optimaux, utilisation de résumés des données...

- Régression sous-échantillonnée en observations et en prédicteurs
- Régressions marginales

Le calcul n'est plus une abstraction

Contraintes : structure des ressources de calcul, accès aux données

Algorithmes *online*, distribués

Modèles à données latentes

Un modèle à données latentes est un modèle statistique étendu $\{p_\theta(x,y)\}_{\theta\in\Theta}$ où seul Y peut être observé

- Les estimés θ_n du paramètre ne peuvent dépendre que des observations $Y_1,\ldots,Y_n$
- Le modèle $\{f_\theta(y)\}_{\theta\in\Theta}$ est défini par marginalisation : $f_\theta(y)=\int p_\theta(x,y)dx$

Les données $\{Y_t\}_{t\geq 1}$ sont supposées indépendantes et distribuées sous la loi marginale π (et $\pi\notin\{f_\theta\}_{\theta\in\Theta}$)

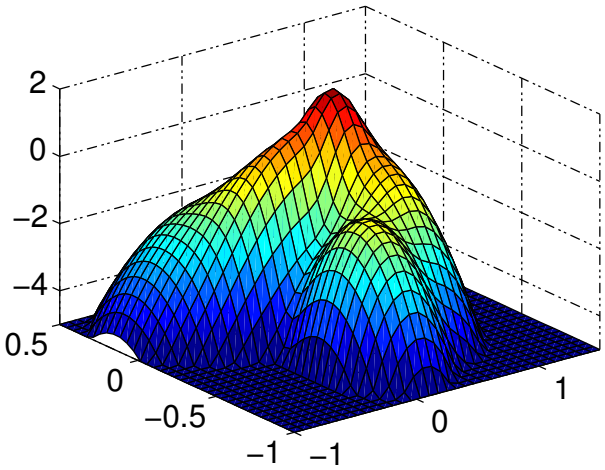
Exemple : Modèle de mélange

Densité de mélange

$$f(y) = \sum_{i=1}^{m} \alpha_i f_i(y)$$

Interprétation en terme de données latentes

$$\mathrm{P}(X_t = i) = \alpha_i$$
$$Y_t | X_t = i \sim f_i(y)$$



Souvent utilisé pour réaliser une version probabiliste du *clustering* (traitement de la parole, du langage naturel. . .)

L'algorithme EM usuel vise à déterminer numériquement l'estimateur du maximum de vraisemblance

$$\theta_n = \arg\max_{\theta} \sum_{t=1}^{n} \log f_\theta(Y_t)$$

Expectation-Maximisation (Dempster, Laird & Rubin, 1977)

A l'étape k, étant donné l'estimé courant θ_n^k du paramètre

Etape E Calculer

$$q_{n,\theta_n^k}(\theta) = \sum_{t=1}^{n} \mathrm{E}_{\theta_n^k}\left[\log p_\theta(X_t, Y_t) \middle| Y_t\right]$$

Etape M Mettre à jour l'estimé du paramètre

$$\theta_n^{k+1} = \arg\max_{\theta\in\Theta} q_{n,\theta_n^k}(\theta)$$

Principe

1. C'est un algorithme MM (maximisation d'un minorant) de par l'inégalité de Jensen

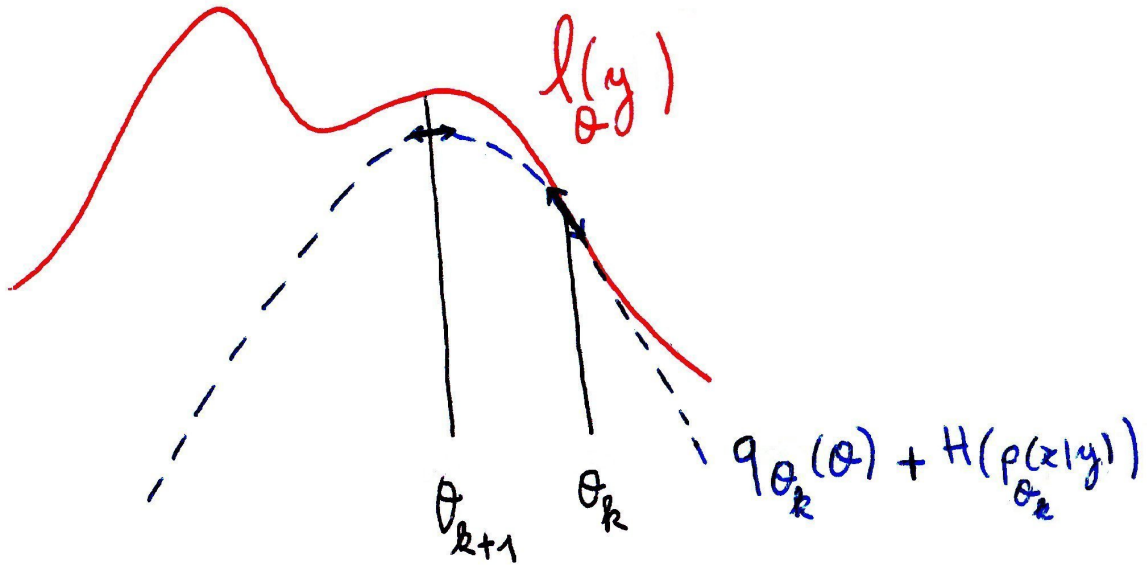
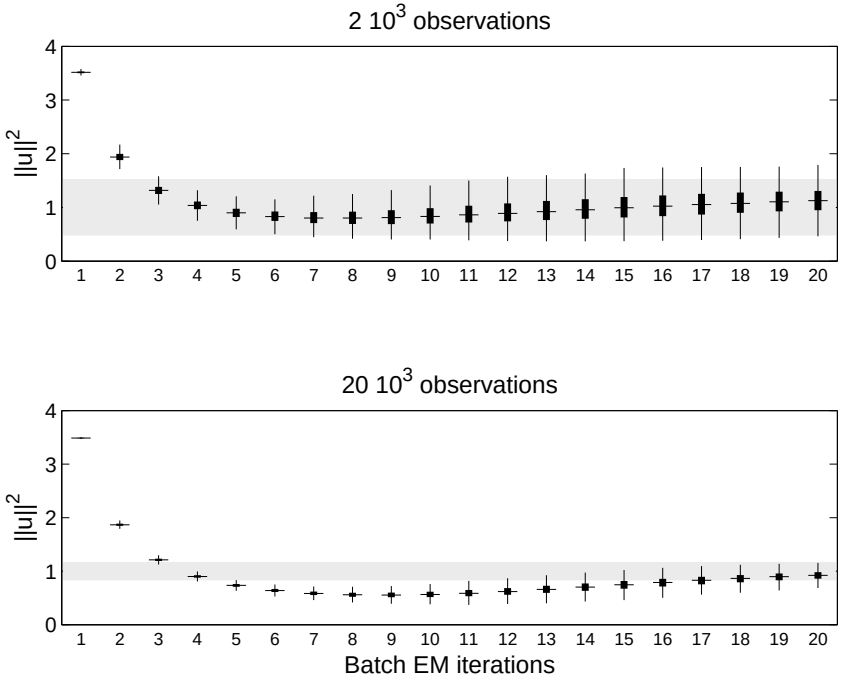


Figure: La quantité intermédiaire de l'EM minore localement la log-vraisemblance

2. Qui ne peut s'arrêter qu'en un point stationnaire de la log-vraisemblance (du fait de la relation dite de Fisher)

L'algorithme EM n'a pas un comportement satisfaisant lorsque n est grand

Le coût de calcul (et de stockage) de l'algorithme EM (pour un nombre fixé d'itération) est proportionnel à n pour un résultat qui ne s'améliore pas quand $n \uparrow \infty$



Algorithme online EM [C & Moulines, 2009–2011]

Pour remédier à ces défauts, on cherche une variante de l'algorithme EM

- qui à chaque nouvelle observation Y_n, met à jour l'estimé θ_n
- avec un coût de calcul (et de stockage) fixe par observation
- peut être interrompue à tout moment (et n'a pas besoin de connaître au préalable le nombre d'observations)
- est robuste vis à vis de contraintes d'accès au données
- se comporte comme le maximum de vraisemblance lorsque n est grand (mesuré via un TLC)

Ingrédient 1. Modèle de la famille exponentielle

Pour que la récursion de l'EM soit explicite on doit supposer

Un modèle de la famille exponentielle

$$p_\theta(x,y) = \exp\left(s(x,y)\psi(\theta) - A(\theta)\right)$$

où $s(x,y)$ est une statistique exhaustive pour la loi p_θ

pour lequel le maximum de vraisemblance est explicite

$$S \mapsto \bar{\theta}(S) = \arg\max_\theta \; S\psi(\theta) - A(\theta)$$

Ingrédient 1. EM dans la famille exponentielle

La k-ème itération de l'algorithme EM

Etape E

$$S_n^{k+1} = \frac{1}{n} \sum_{t=1}^{n} \mathrm{E}_{\theta_n^k} \left[s(X_t, Y_t) \middle| Y_t \right]$$

Etape M

$$\theta_n^{k+1} = \bar{\theta} \left(S_n^{k+1} \right)$$

peut être décrite uniquement via la statistique exhaustive

$$S_n^{k+1} = \frac{1}{n} \sum_{t=1}^{n} \mathrm{E}_{\bar{\theta}(S_n^k)} \left[s(X_t, Y_t) \middle| Y_t \right]$$

Ingrédient 2. L'algorithme EM limite

En faisant tendre n vers l'infini on obtient

Version limite de l'algorithme EM

Mise à jour de la statistique exhaustive

$$S^k = \mathrm{E}_\pi \left(\mathrm{E}_{\bar{\theta}(S^{k-1})} \left[s(X_1, Y_1) \middle| Y_1 \right] \right)$$

Mise à jour du paramètre

$$\theta^k = \bar{\theta} \left\{ \mathrm{E}_\pi \left(\mathrm{E}_{\theta^{k-1}} \left[s(X_1, Y_1) \middle| Y_1 \right] \right) \right\}$$

Avec les même arguments que pour l'algorithme EM, on montre que

1. la divergence de Kullback-Leibler $D(\pi | f_{\theta^k})$ décroît de façon monotone avec k
2. θ_k converge vers $\{\theta : \nabla_\theta D(\pi | f_\theta) = 0\}$

Ingrédient 3. L'approximation stochastique

- On recherche les solutions de

$$\mathrm{E}_{\pi}\left(\mathrm{E}_{\bar{\theta}(S)}\left[s(X_1, Y_1)\middle| Y_1\right]\right) - S = 0$$

- En voyant $\mathrm{E}_{\bar{\theta}(S)}\left[s(X_n, Y_n)\middle| Y_n\right]$ comme une observation bruitée de $\mathrm{E}_{\pi}\left(\mathrm{E}_{\bar{\theta}(S)}\left[s(X_1, Y_1)\middle| Y_1\right]\right)$, on reconnaît un cas d'utilisation de l'approximation stochastique (ou algorithme de Robbins-Monro) :

$$S_n = S_{n-1} + \gamma_n\left(\mathrm{E}_{\bar{\theta}(S_{n-1})}\left[s(X_n, Y_n)\middle| Y_n\right] - S_{n-1}\right)$$

où (γ_n) est une séquence de pas décroissants

L'algorithme

Online EM

Etape E via l'approximation stochastique

$$S_n = (1-\gamma_n)S_{n-1} + \gamma_n \mathrm{E}_{\theta_{n-1}}\left[s(X_n, Y_n) \middle| Y_n\right]$$

Etape M

$$\theta_n = \bar{\theta}(S_n)$$

Natural language processing

`Online EM for Unsupervised Models, Percy Liang & Dan Klein, NAACL Conference, 2009`

Context Text processing using word representation ($\approx$ 10k words) from large document corpora ($\approx$ 100k documents)

Tasks Tagging, classification, alignment (variant of mixture or HMM with multinomial observations)

Online EM for Unsupervised Models

Percy Liang Dan Klein
Computer Science Division, EECS Department
University of California at Berkeley
Berkeley, CA 94720
{pliang,klein}@cs.berkeley.edu

Abstract

The (batch) EM algorithm plays an important role in unsupervised induction, but it sometimes suffers from slow convergence. In this paper, we show that online variants (1) provide significant speedups and (2) can even find *better* solutions than those found by batch EM. We support these findings on four unsupervised tasks: part-of-speech tagging, document classification, word segmentation, and word alignment.

1 Introduction

In unsupervised NLP tasks such as tagging, parsing, and alignment, one wishes to induce latent linguistic structures from raw text. Probabilistic modeling has emerged as a dominant paradigm for these problems, and the EM algorithm has been a driving force for learning models in a simple and intuitive manner. However, on some tasks, EM can converge slowly. For instance, on unsupervised part-of-speech tagging, EM requires over 100 iterations to near convergence on the Wall-Street Journal (subset) of examples. Online algorithms have the potential to speed up learning by making updates more frequently. However, these updates can be seen as noisy approximations to the full batch update, and this noise can in fact impede learning.

This tradeoff between speed and stability is familiar to online algorithms for convex supervised learning problems—e.g., Perceptron, MIRA, stochastic gradient, etc. Unsupervised learning raises two additional issues: (1) Since the EM objective is non-convex, we often get convergence to different local optima of varying quality; and (2) we evaluate on accuracy metrics which are at best loosely correlated with the EM likelihood objective (Liang and Klein, 2008). We will see that these issues can lead to surprising results.

In Section 4, we present a thorough investigation of online EM, mostly focusing on stepwise EM, which it dominates incremental EM. For stepwise EM, we find that choosing a good stepsize and initial size is important but can fortunately be done adequately without supervision. With a good stepsize, stepwise EM reaches the same (or better)

In this application, the use of mini-batch blocking was found useful:

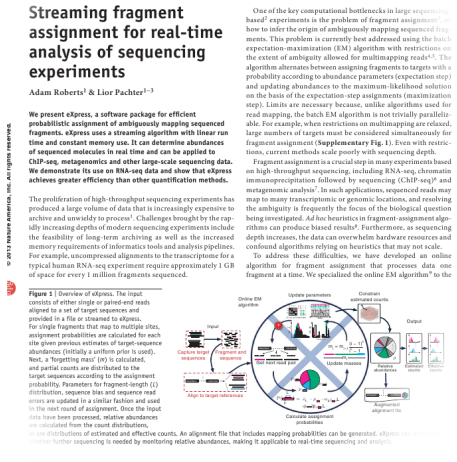
- Apply the proposed algorithm considering $Y_{mk+1}, Y_{mk+2} \dots Y_{m(k+1)}$ as one observation (with m of the order of a few k documents)

Large-scale probabilistic sequence alignment

Streaming fragment assignment for real-time analysis of sequencing experiments, Adam Roberts & Lior Pachter, *Nature*, 2013

Context High-throughput sequencing experiments in biology

Task Learn parameters of probabilistic models for sequence alignments (finite-valued sequences with insertion/deletion/substitution probabilities) from 10M sequence fragments (reads) with 10k target sequences (transcripts)



Streaming fragment assignment for real-time analysis of sequencing experiments

Adam Roberts & Lior Pachter

Note that the actual goal is the retrospective probabilistic sequence assignment (E-step)

Use a single pass of online EM through all the data, followed by a few more batch EM steps

- Roberts, A. & Pachter, L. (2013). Streaming fragment assignment for real-time analysis of sequencing experiments. Nature.
- Cappé, O. (2011). Online EM algorithm for hidden Markov models. J. Comput. Graph. Statist.
- Cappé, O. (2011). Online Expectation-Maximisation. In *Mixtures*, Wiley.
- Rohde, D. & Cappé, O. (2011). Online maximum-likelihood estimation for latent factor models. IEEE Statistical Signal Processing Workshop.
- Liang, P. & Klein, D. (2009). Online EM for Unsupervised Models. NAACL Conference.
- Cappé, O. & Moulines, E. (2009). On-line expectation-maximization algorithm for latent data models. J. Roy. Statist. Soc. B.
- Sato, M. & Ishii, S. (2000). On-line EM algorithm for the normalized Gaussian network. Neural Computation.
- Sato, M. (2000). Convergence of on-line EM algorithm. International Conference on Neural Information Processing.
- Neal, R. M. & Hinton, G. E. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, MIT Press.
- Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. J. Roy. Statist. Soc. B.

Merci de votre attention