Apprentissage automatique

des principes et des algorithmes

Balázs Kégl

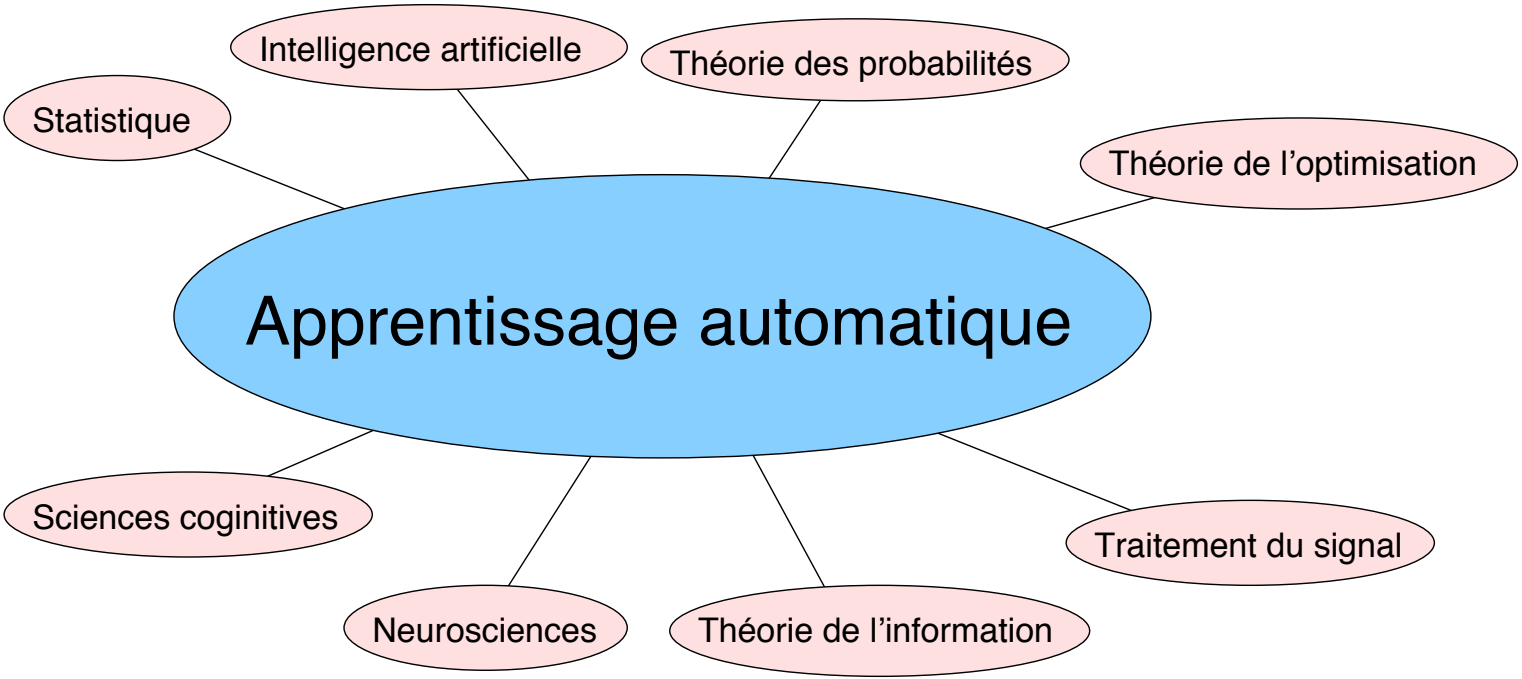
Laboratoire de l'accélérateur linéaire
CNRS

6 mars 2006

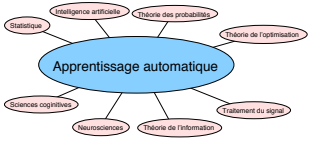
Résumé

- Introduction
- Familles de problèmes
  - classification, régression, estimation de densité
  - exemple : AUGER
- Méthodes

Apprentissage automatique au carrefour

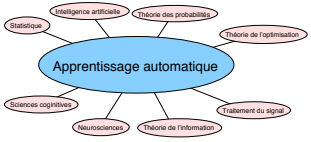


Apprentissage automatique au carrefour



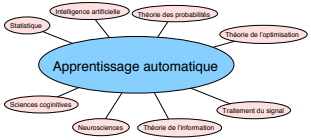
- Statistique
  - but : apprendre des fonctions à partir des données
  - e.g. : fonctions de densité, régression
  - différences
    - fonctions complexes – statistique non-paramétrique
    - grands jeux de données, grandes dimensions – algorithmique
    - sources d'inspiration

Apprentissage automatique au carrefour



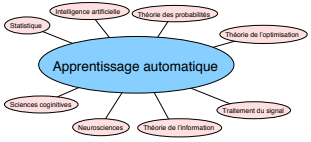
- Intelligence artificielle
  - but : imiter ou reproduire des comportements intelligents “naturels”
  - source de problèmes “classiques” (reconnaissance d’écriture, parole, etc.)
  - différences
    - approche inductive – apprentissage a partir des exemples
    - approche probabiliste

Apprentissage automatique au carrefour



- Théorie des probabilités
  - outils d'analyse de modèles théoriques
- Théorie de l'optimisation
  - outils algorithmiques

Apprentissage automatique au carrefour

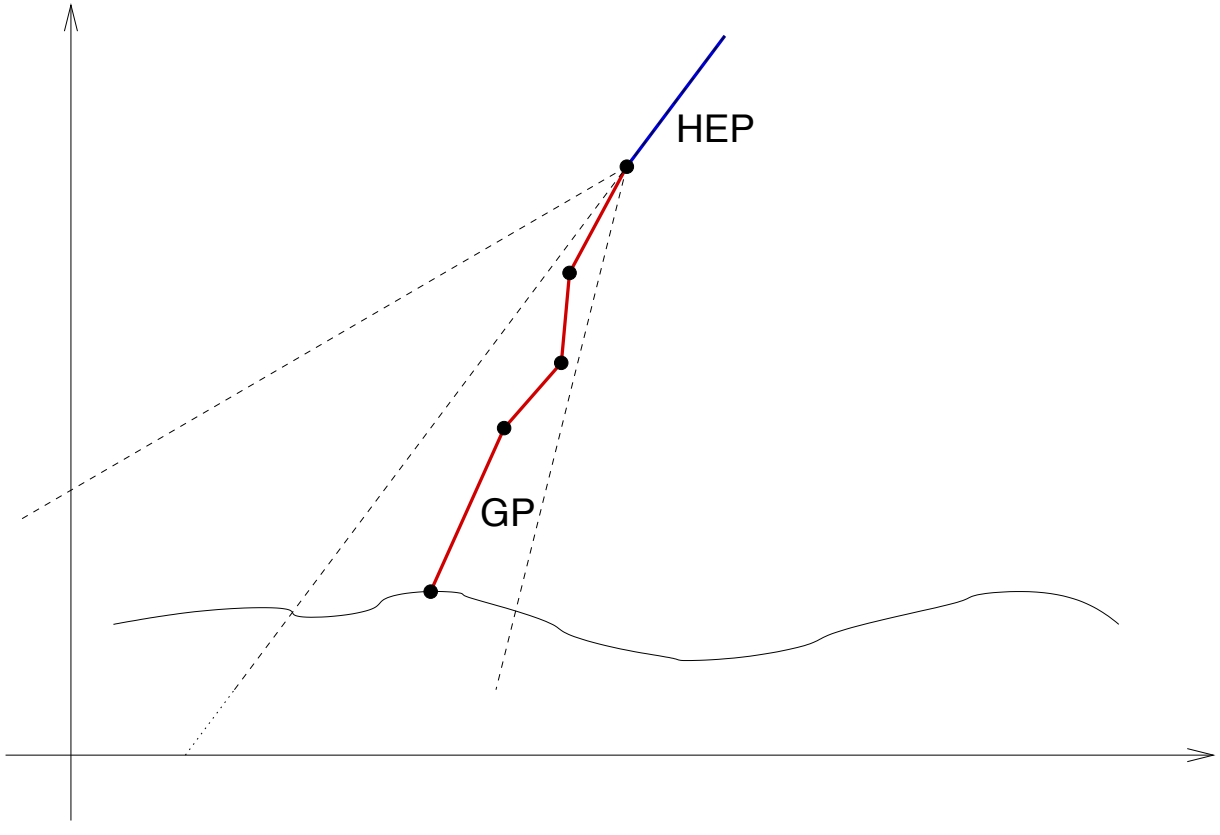


- Sciences cognitives, neurosciences
  - sources d'inspiration
- Théorie de l'information, traitement du signal
  - problèmes et méthodologies partagés

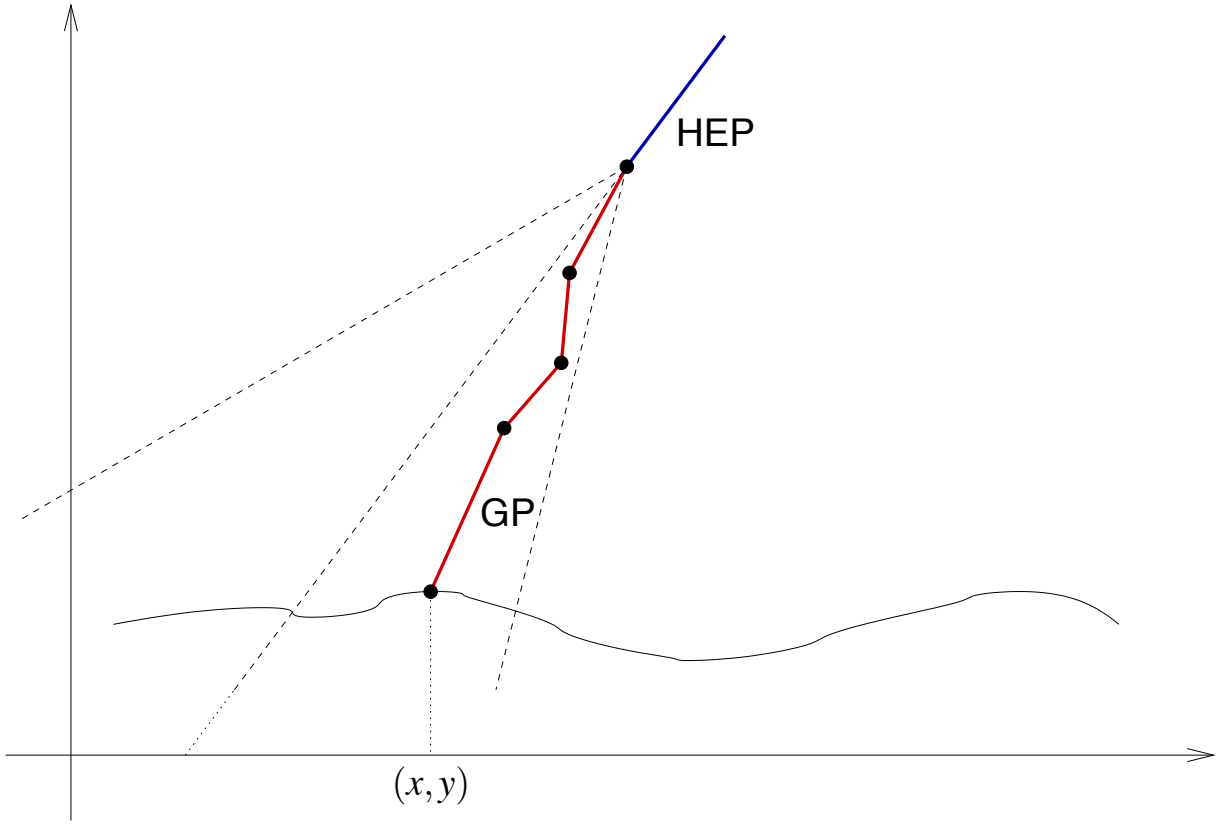
Résumé

- Introduction
- Familles de problèmes
  - classification, régression, estimation de densité
  - exemple : AUGER
- Méthodes

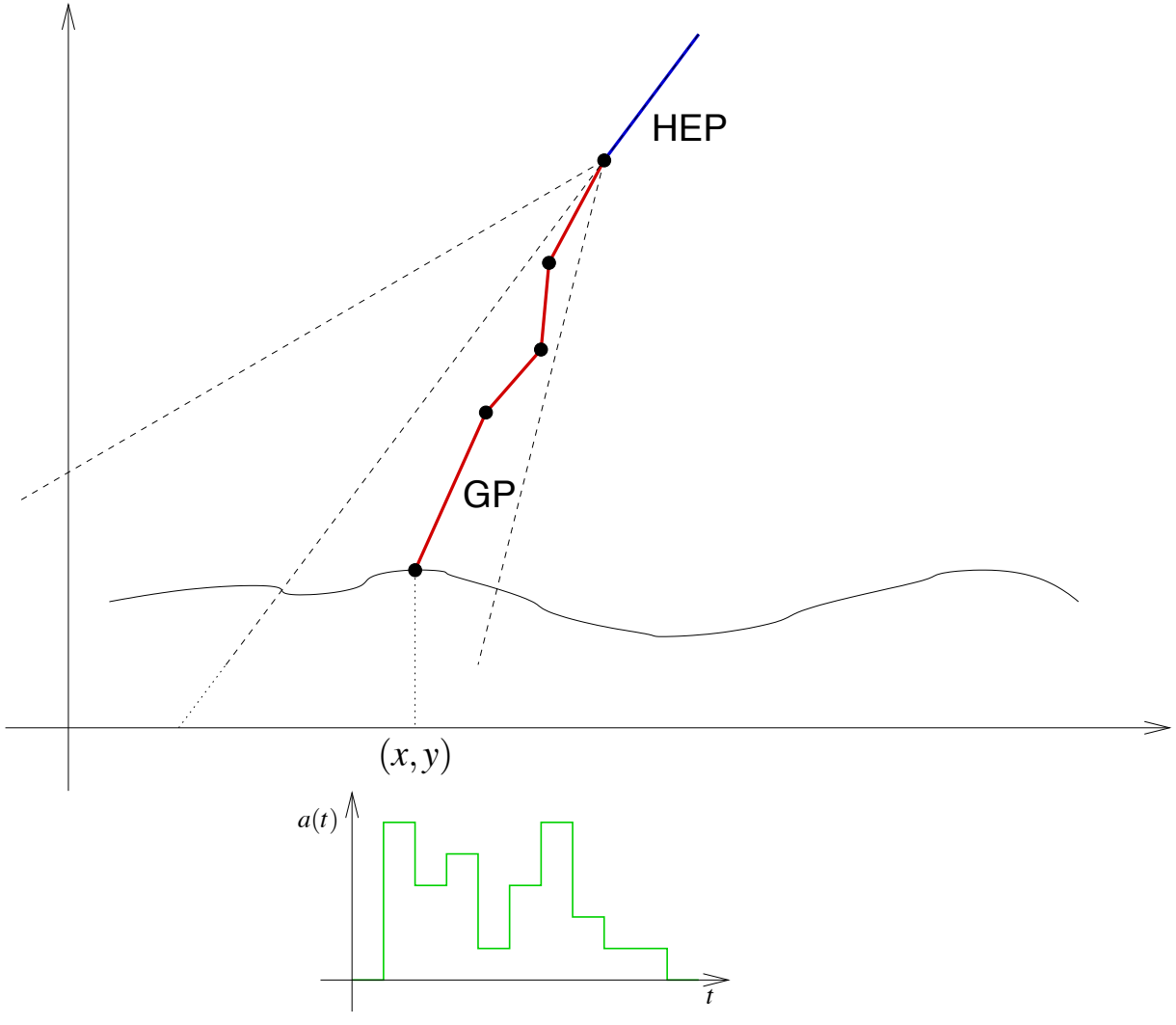
Exemple : AUGER



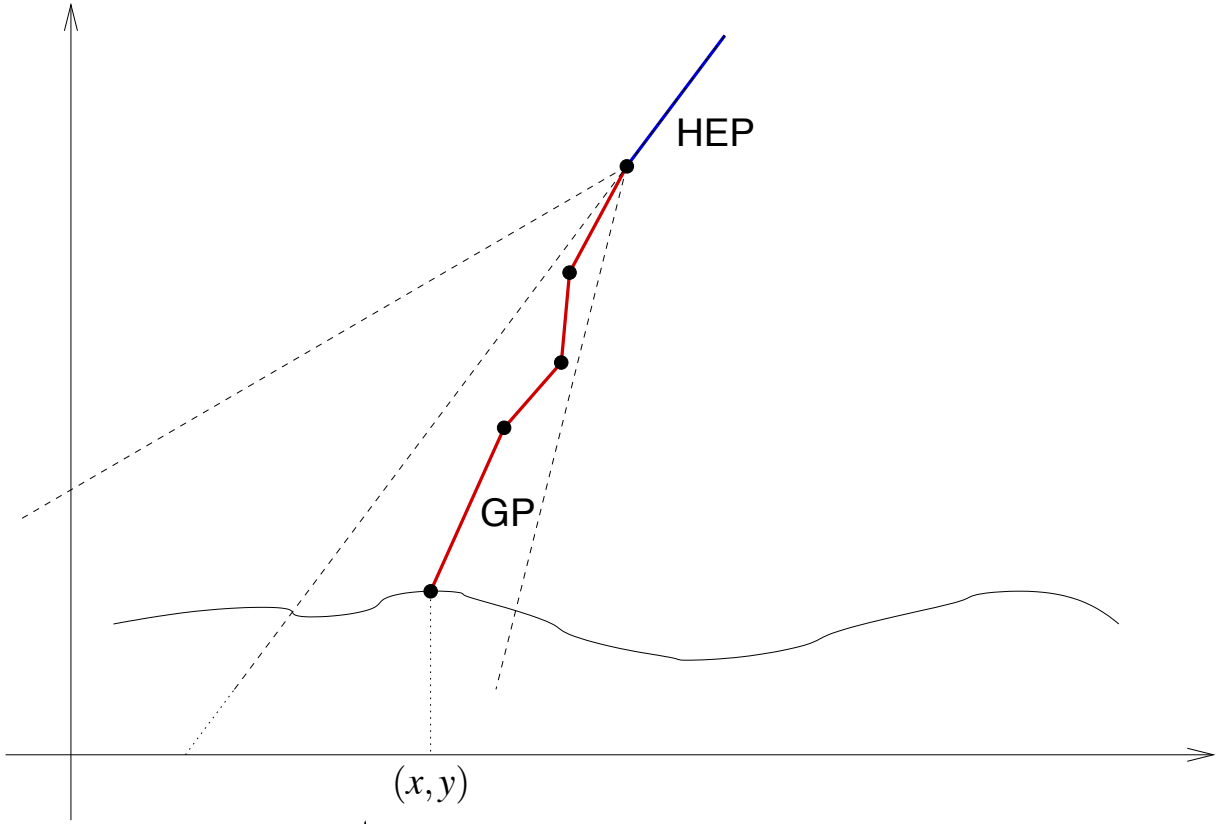
Exemple : AUGER

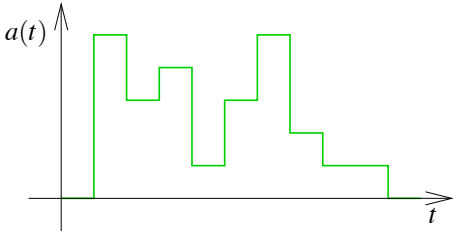


Exemple : AUGER



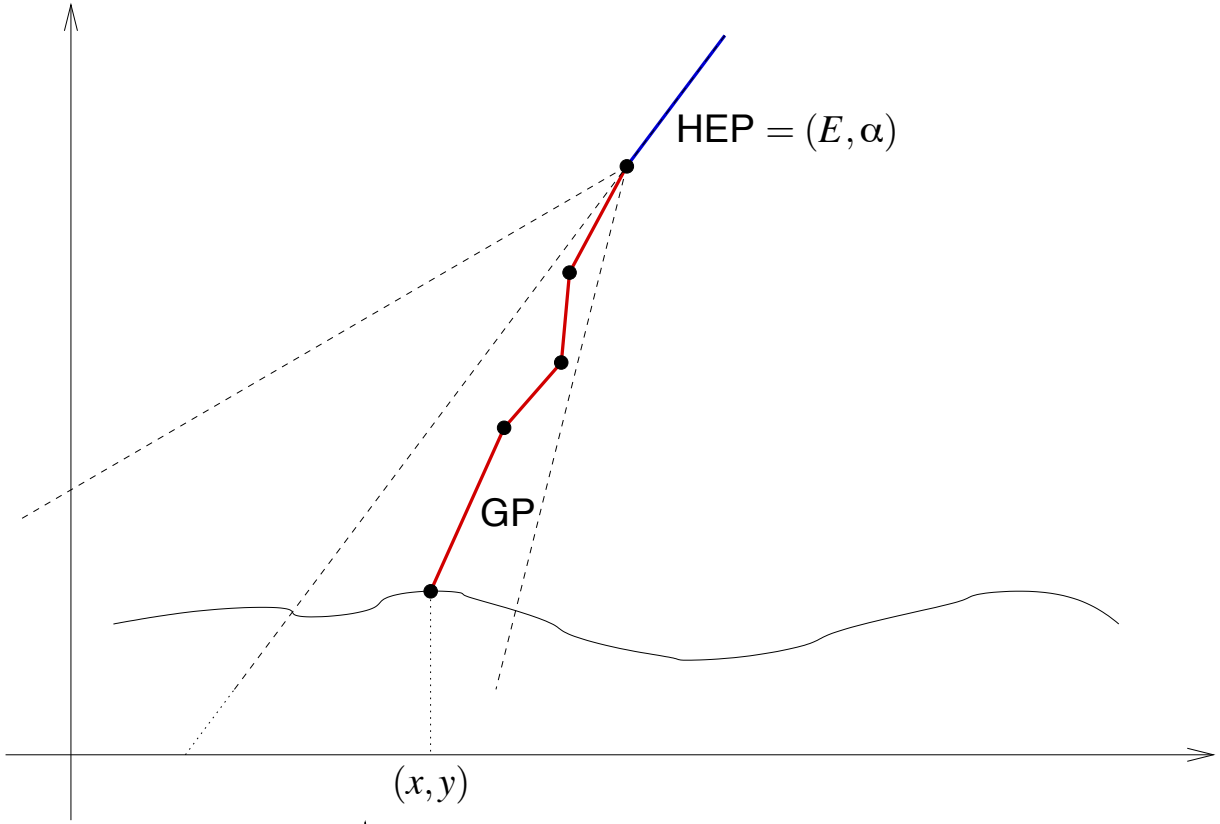
Exemple : AUGER

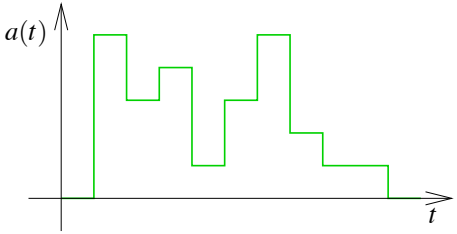




$$= \mathbf{a} = \big(a(1), \ldots, a(T)\big)$$

Exemple : AUGER





$= \mathbf{a} = \big(a(1), \ldots, a(T)\big)$

Formalisation du problème d'AUGER

- Observation:
  - $j = 1, \ldots, N$ cuves: $(x_j, y_j, \mathbf{a}_j)$
  - $\mathbf{x} = (x_1, y_1, a_{1,1}, \ldots, a_{1,T}, \ \ldots \ , x_N, y_N, a_{N,1}, \ldots, a_{N,T}) \in \mathbb{R}^{N(T+2)}$
- Prédiction:
  - énergie $E \in \mathbb{R}$: régression
  - type du particule $\alpha \in \{\text{proton}, \text{fer}\}$: classification
- Estimation de densité $p(\mathbf{x}, E, \alpha)$

Classification

- Setup :
  - observation : $X \in \mathbb{R}^d$
  - étiquette, classe, catégorie : $Y \in \{-1, 1\}$
  - (X, Y) vient d'une distribution (fixe mais inconnue)
- Objectif:
  - trouver une fonction f qui minimise la probabilité d'erreur

$$R(f) = \mathbb{P}\{f(X) \neq Y\} = \mathbb{E}_{(X,Y)}\{\mathbb{I}\{f(X) \neq Y\}\} = \mathbb{E}_{(X,Y)}\{L(f,(X,Y))\}$$

Classification

- Minimisation du risque
  - perte (coût) de f sur un point (X,Y) :

$$L(f,(X,Y)) = \mathbb{I}\{f(X) \neq Y\}$$

  - le risque de f est l'espérance de la perte :

$$R(f) = \mathbb{E}_{(X,Y)}\{L(f,(X,Y))\}$$

Classification

- Applications typiques:
  - reconnaissance de forme (caractères manuscrits, parole, empreintes digitales, etc.)
  - catégorisation de textes (spam / nospam, par sujet, etc.)
  - catégorisation d'expressions génétiques (cellule seine / cancéreuse)
  - diagnose automatique (symptômes $\rightarrow$ diagnose)

Comment construire f?

- Intelligence artificielle classique : “à la main”
  - systèmes experts, bases des règles logiques
  - demande des connaissances “profondes”
  - systèmes transparents
  - règles sont difficiles à extraire
  - pas de garanties de performance
  - décisions oui/non, pas de probabilités

Comment construire f?

- Apprentissage automatique : induction
  - échantillon d'entraînement :

$$D_n = \big((\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\big)$$

  - ensemble de fonctions candidates : F
  - risque empirique (moyenne d'erreur) :

$$\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} L(f, (\mathbf{x}_i, y_i)) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{f(\mathbf{x}_i) \neq y_i\}$$

  - principe de minimisation du risque empirique :

$$\widehat{f} = \arg\min_{f \in F} \widehat{R}(f)$$

Comment construire f?

- Problème potentiel : overfitting
  - $\widehat{R}(f)$ est petit mais pas $R(f)$
- Statistique “classique” :
  - f est parametrisé avec peu de paramètres
  - $\lim_{n\to\infty} \widehat{R}(f) = R(f)$

Comment construire f?

- Problème potentiel : overfitting
  - $\widehat{R}(f)$ est petit mais pas $R(f)$
- Apprentissage automatique – statistique non-paramétrique
  - f est complexe, flexible
  - le nombre des paramètres est comparable au nombre de variables (d) et au nombre de points d'entraînement (n), peut augmenter avec n
  - le contrôle de complexité ($\sim$ nombre de paramètres) est inévitable

Histoire

- Algorithmes
  - 1958 : Perceptron [Rosenblatt, ’58] – [Minsky–Papert ’69]
  - 1986 : Réseaux de neurones et l’algorithme de rétro-propagation [Rumelhart–Hinton–Williams, ’86]
  - 1995: Machines à vecteurs de support [Boser–Guyon–Vapnik, ’92], [Cortes–Vapnik, ’95]
  - 1997: boosting, AdaBoost [Freund, ’95], [Freund–Schapire, ’97]

L'approche générale

- Modèle linéaire généralisé : $f(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j h_j(\mathbf{x})$
  - Perceptron : $h_j(\mathbf{x}) = x^{(j)}$
  - Réseaux de neurones : $h_j(\mathbf{x}) = \sigma(\mathbf{w}_j^T \mathbf{x})$
  - Machines à vecteurs de support : $h_j(\mathbf{x}) = K(\mathbf{x}_j, \mathbf{x})$
  - AdaBoost : $h_j(\mathbf{x})$ arbitraire

Régression

- Setup:
  - observation : $X \in \mathbb{R}^d$
  - réponse : $Y \in \mathbb{R}$
  - (X, Y) vient d'une distribution (fixe mais inconnue)
- Objectif:
  - trouver un fonction f qui minimise l'erreur quadratique

$$R(f) = \mathbb{E}_{(X,Y)} \left\{ (f(X) - Y)^2 \right\}$$

Régression

- Minimisation du risque
  - perte (coût) de f sur un point (X,Y) :

$$L(f,(X,Y)) = (f(X)-Y)^2$$

  - le risque de f est l'espérance de la perte :

$$R(f) = \mathbb{E}_{(X,Y)}\{L(f,(X,Y))\}$$

Régression

- Applications typiques:
  - prédiction de séries de temps (météorologie, bourse)
  - identification du système
  - prédiction de durées, coûts, etc.
- Méthodes:
  - régression linéaire
  - splines, noyaux, SVM, processus gaussiens

Estimation de densité

- Setup:
  - observation : $X \in \mathbb{R}^d$
  - objectif : trouver la distribution $p(X)$ de l'observation X
  - plus difficile que la classification et la régression
  - si on connaît $p(X,Y)$, les problèmes de classification et régression sont triviales

Estimation de densité

- Classification
  - Suppose que les deux distributions de classe

$$p(X|Y=+1) \quad \text{et } p(X|Y=-1)$$

    et les deux probabilités a-priori

$$\mathbb{P}\{Y=+1\} \quad \text{et } \mathbb{P}\{Y=-1\}$$

    sont connues

  - alors par le théorème de Bayes :

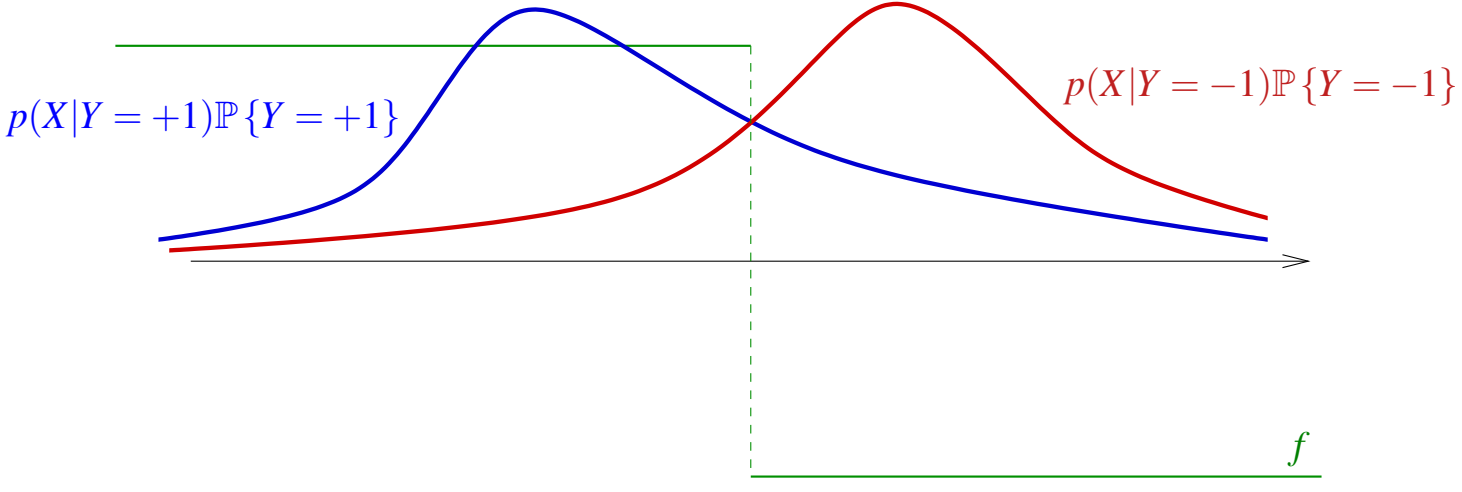
$$\mathbb{P}\{Y=+1|X\} = \frac{p(X|Y=+1)\mathbb{P}\{Y=+1\}}{p(X)}$$

$$\mathbb{P}\{Y=-1|X\} = \frac{p(X|Y=-1)\mathbb{P}\{Y=-1\}}{p(X)}$$

Estimation de densité

- Classification
  - donc la décision optimale est

$$f(X) = \begin{cases} +1 & \text{si } p(X|Y=+1)\mathbb{P}\{Y=+1\} \geq p(X|Y=-1)\mathbb{P}\{Y=-1\} \\ -1 & \text{sinon} \end{cases}$$



Estimation de densité

- Régression
  - si les distributions conditionnelles

$$p_{\mathbf{x}}(Y|X=\mathbf{x})$$

    sont connues
  - alors l'espérance conditionnelle

$$\mu(\mathbf{x}) = \mathbb{E}_Y\left\{p_{\mathbf{x}}(Y|X=\mathbf{x})\right\}$$

    minimise le risque quadratique

Estimation de densité

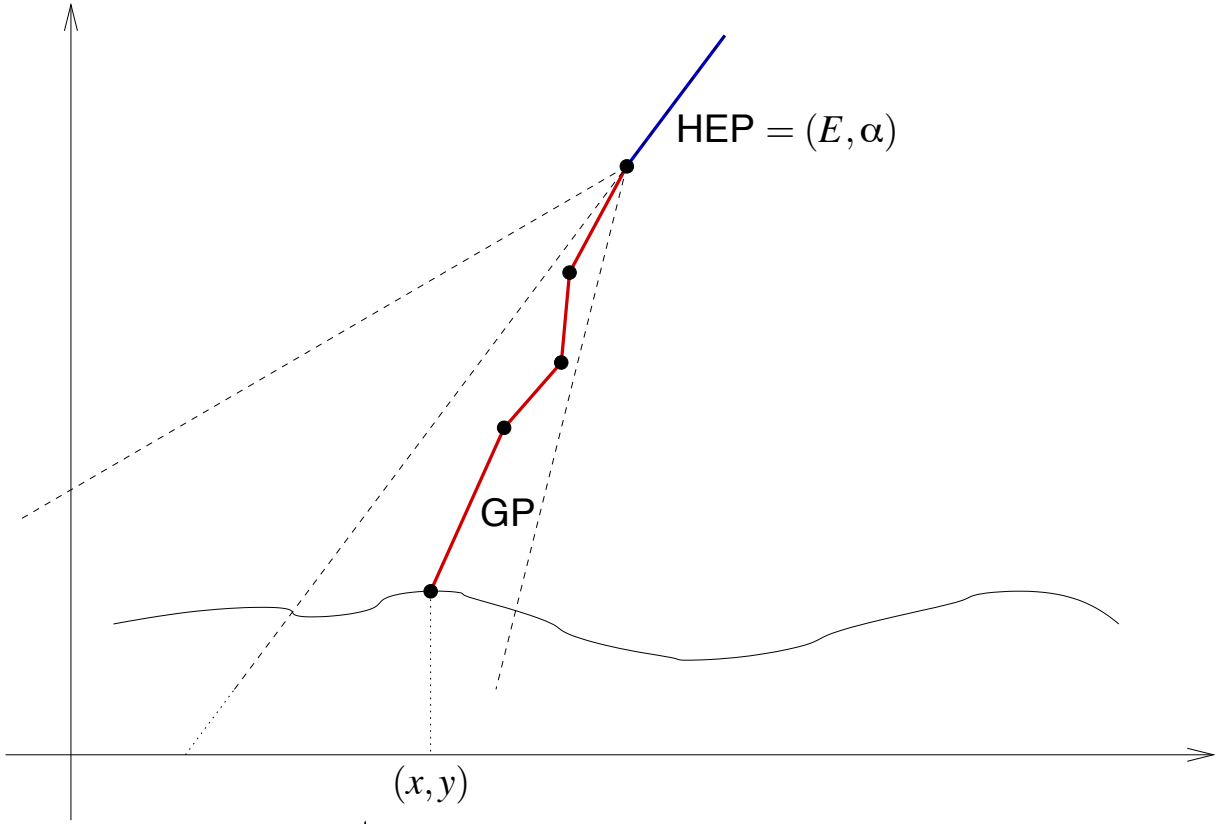
- Principe : maximum de vraisemblance
  - données : $D_n = (\mathbf{x}_1, ..., \mathbf{x}_n)$

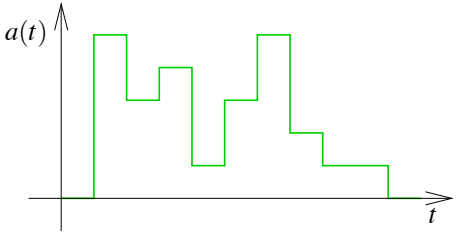
$$\widehat{p} = \arg\max_{p \in F} \prod_{i=1}^{n} p(\mathbf{x}_i) = \arg\max_{p \in F} \sum_{i=1}^{n} \log p(\mathbf{x}_i) = \arg\max_{p \in F} \frac{1}{n} \sum_{i=1}^{n} \log p(\mathbf{x}_i)$$

  - si la perte est définie comme $L(p, \mathbf{x}) = -\log p(\mathbf{x})$, alors

$$\widehat{p} = \arg\min_{p \in F} \widehat{R}(p)$$

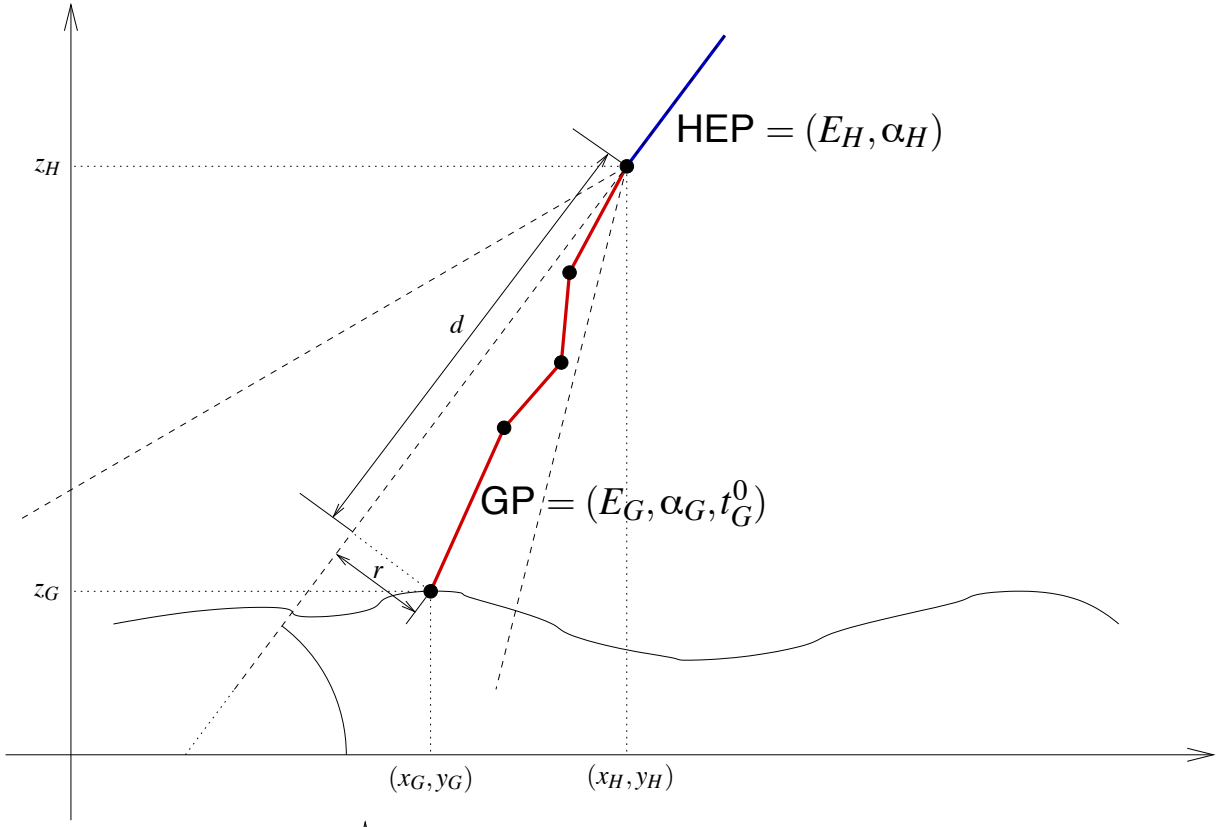
Exemple : AUGER

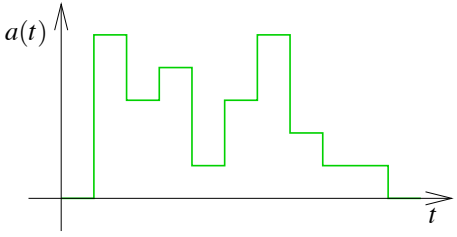




$= \mathbf{a} = \big(a(1), \ldots, a(T)\big)$

Exemple : AUGER



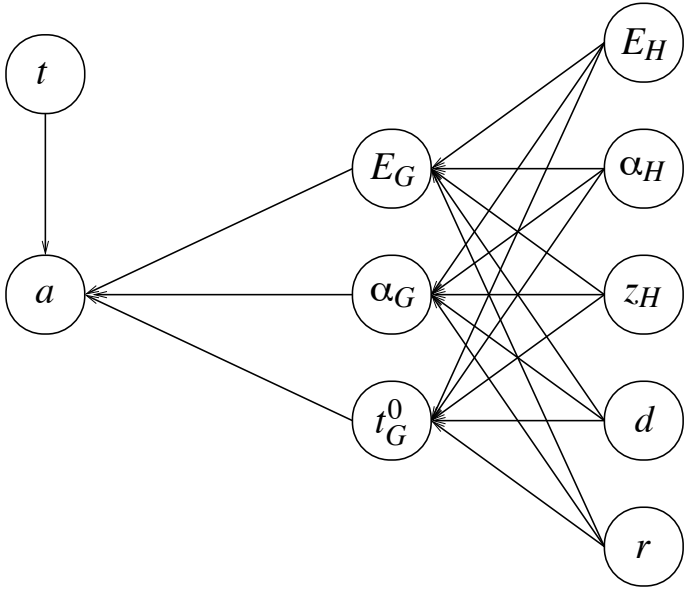


$= \mathbf{a} = \big(a(1), \ldots, a(T)\big)$

Exemple : AUGER

- Variables cachées $\rightarrow$ modèles graphiques

$$p(a|\alpha_H) = \int p(a,\alpha_G|\alpha_H)d\alpha_G = \int p(a|\alpha_G,\alpha_H)p(\alpha_G|\alpha_H)d\alpha_G$$
$$= \int p(a|\alpha_G)p(\alpha_G|\alpha_H)d\alpha_G$$



Estimation de densité

- Modèles graphiques
  - outils de modélisation : modularité, factorisation
  - outils d'inférence, de simulation
  - outils d'estimation : EM (expectation-maximization)

Conclusion

- Apprentissage automatique – modélisation statistique
- Induction : généralisation à partir des données
- Estimation de densité, régression, classification