Le cas linéaire

Marie-Odile Berger

October 17, 2016

Importance du modèle linéaire

Définition: un modèle linéaire est un ensemble d'objects s'écrivant sous la forme

$$\mathcal{F} = \sum_{i=1}^{n} \alpha_i u_i, \alpha_i \in R(ouC)$$

où u_i est la base de l'ensemble des formes.

Exemples de base

- des vecteurs de R^n lorsque les descripteurs ont n éléments
- des bases de R^n choisies pour bien représenter linéairement les données (voir la section sur l'ACP)
- des fonctions: parmi les représentation les plus utilisées de fonction, on trouve les séries de Fourrier et les fonctions splines

Intérêt du modèle linéaire

Il existe des outils très efficaces issus de l'algèbre linéaire permettant

- étant donnée une forme, de trouver les α_i la représentant
- étant donnés un ensemble de formes, de trouver la base des u_i permettant de caractériser au mieux cet ensemble de manière linéaire.

Comparatif linéaire/non linéaire

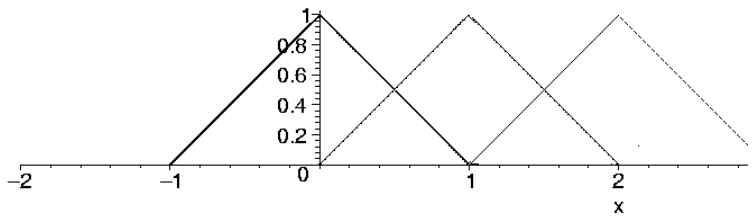
Resoudre	Trouver x tel que
$AX = b$	$f(x) = b$
est direct dans le cas linéaire (qu'il y ait unicité ou infinité de solutions)	avec f quelconque est difficile en général et sans garantie de temps d'éxecution ni de complétude des solutions

Exemple des B splines

Soit $B_i(t)$ la base des fonctions splines.
spline de degré 1:

$$B(t) = \begin{cases} 0 \ si \ t < 0 \\ t \ si \ t < 1 \\ 2 - t \ si \ t < 2 \\ 0 \ si \ 2 <= t \end{cases}$$

On définit $B_i(t) = B(t - i + 1)$.
La fonction polygonale passant par les $Q_i = (x_i, y_i)_{\{1 \leq i \leq n\}}$ s'écrit
$f(t) = \sum_{i=1}^{n} (x_i B_i(t), y_i B_i(t))$.
il s'agit d'une fonction linéaire et on vérifie que
$f(i) = (x_i B_i(i), y_i B_i(i)) = (x_i, y_i) = Q_i$.



splines de degré 2

définir des fonctions par morceaux de degré 2, présentant une continuité C^0 et C^1:

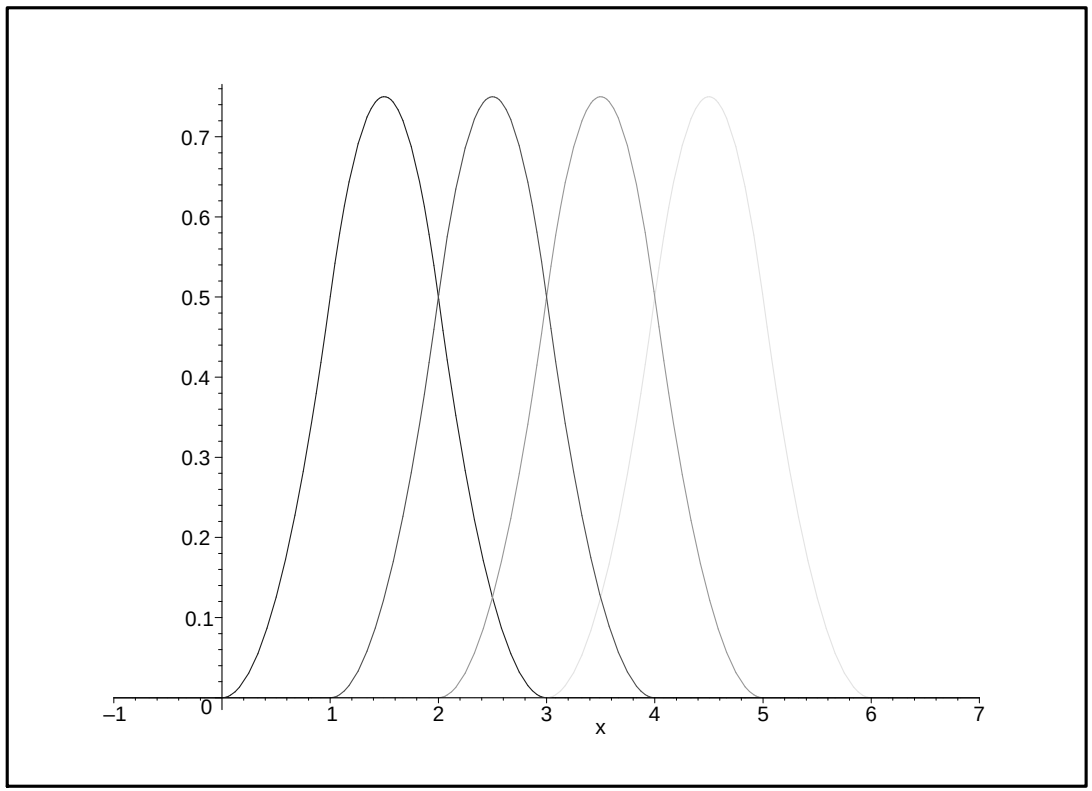
en chaque noeud, 2 contraintes $f_g = f_d$ et $f'_g = f'_d$

2 intervalles	2x3 contraintes	6 inconnues
3 intervalles	2x4 contraintes	9 inconnues

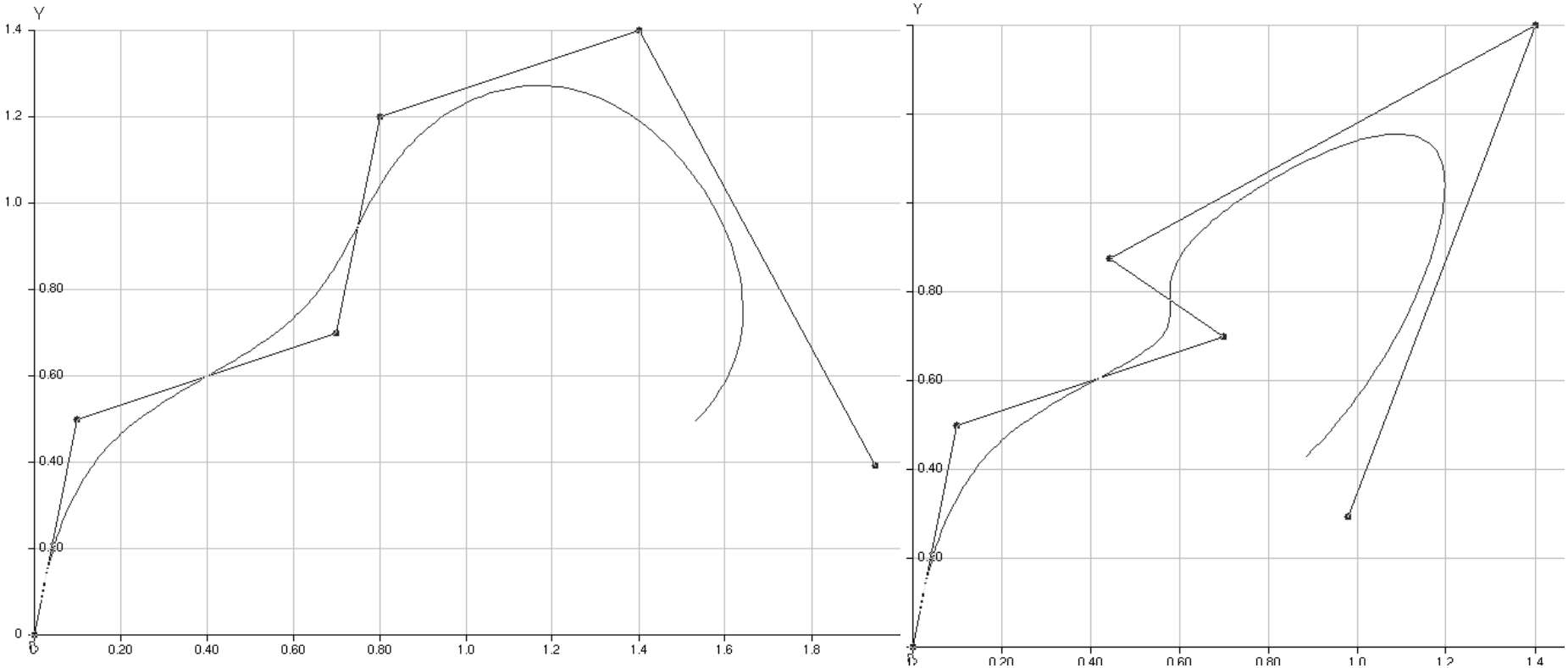
Sur 3 intervalles + contrainte de normalisations $\rightarrow$ les coeffs des polynomes sont déterminés.

splines de degré 2:

$$B(t) = \begin{cases} 0 \ si \ t < 0 \\ 1/2t^2 \ si \ t < 1 \\ -1/2 + t - (t-1)^2 \ si \ t < 2 \\ 5/2 - t + 1/2 * (t-2)^2 \ si \ t < 3 \\ 0 \ si \ 3 <= t \end{cases}$$



Exemples d'approximation par des fonctions spline



Quelques exemples de courbes définies par B splines

Les problèmes majeurs à résoudre pour modéliser linéairement

- trouver la représentation d'une forme dans une base donnée: résoudre des systèmes $AX = B$ où $A(m \times n)$ est de taille quelconque. Si $m > n$ trouver la meilleure solution aux moindres carrés.
- trouver la meilleure représentation linéaire d'un ensemble de formes (analyse en composantes principales, analyse en composantes indépendantes)
- s'assurer que la représentation linéaire est bien adaptée au problème!!!

Part I

Rappels d'algèbre linéaire

Ce qui doit être connu...

Noyau, image, base, pseudo-inverse, projection, valeurs propres et vecteurs propres. soit A une matrice $m \times n$

$$Ker(A) = \{x \in R^n | Ax = 0\}$$

$$Im(A) = \{y \in R^m | \exists x \in R^n y = Ax\}$$

$$rang(A) = dim(Im(A))$$

$$dim(Ker(A)) + rang(A) = n$$

Résolution de systèmes linéaires $AX = b$

- si $Ker(A) \neq 0$, il n'y a pas de solution unique.
- Si X_0 est solution alors $X_0 + Ker(A)$ l'est aussi.
- si b n'appartient pas à $Im(A)$, il n'y a pas de solutions.

Rq: dans le cas de données numériques, c'est nettement moins simple:

- on peut avoir $Ker(A) = 0$ avec pourtant des directions tq $||Au||$ est petite.
  En théorie, $Ax = 0$ n'a pas de solutions non triviales alors qu'en pratique il existe des valeurs tq Ax est très proche de 0 qui risquent de perturber le calcul de la solution.
- Un système $Ax = b$ de taille $m \times n$ avec $m > n$ n'a pas de solutions en général (sauf équations redondantes). En pratique pourtant, on aimerait satisfaire ce système au mieux (voir chapitre sur les moindres carrés)

Part II

Résolution aux moindres carrés

Le problème

objectifs: étant donné un ensemble de données supposées suivre un modèle linéaire, trouver les paramètres du modèle correspondant:

- l'ajustement par une droite: Soit un ensemble de n points (x_i, y_i) que l'on veut approximer par une droite d'équation $y = ax + b$. Le modèle est défini par le paramètre $p = \begin{bmatrix} a \\ b \end{bmatrix}$.
- l'approximation d'une courbe définie par un ensemble de points par des fonctions splines: Soient $(t_i, x_i)_{i \leq n}$ l'ensemble des points à approximer sur la base de splines
  On cherche α_j

$$f(t) = \sum_{j=1}^{j=m} \alpha_j B_j(t)$$

  tel que $f(t)$ approxime au mieux $(t_i, x_i)_{i \leq n}$

Moindres carrés: le problème

Dans la plupart des cas, le nombre de mesures est supérieur au nombre d'inconnues ($m >> n$). Le système $Ap = z$ n'a donc pas de solution ...
Sachant que les mesures sont entachées d'incertitudes, on cherche à trouver p satisfaisant au mieux le système $Ap = z$.
Dans le cas de la droite, une mesure naturelle de l'adéquation des mesures au modèle est $C = \sum_{i=1}^{i=n}(y_i - ax_i - b)^2$, c'est à dire $||z - Ap||$.

Résolution de $Ap = z$ aux moindres carrés:

Chercher $\hat{p}$ tel que

$$||Z - A\hat{p}||^2 = min_p||Z - Ap||^2$$

Solution des moindres carrés

Résolution de $Ap = z$ aux moindres carrés:

$$\hat{p} = (A^t A)^{-1} A^t Z$$

preuve:
$\hat{p}$ est minimum de $f(p) = ||Z - Ap||^2$ donc la dérivée de f doit s'annuler.

$$\begin{aligned} f(p+h) - f(p) &= < Z - A(p+h), Z - A(p+h) > - < Z - Ap, Z - Ap > \\ &= -2 < Z - Ap, Ah > + < Ah, Ah > \\ &= -2(A^t(Z - Ap), h > + < Ah, Ah > \end{aligned}$$

La dérivée de $f(X)$ s'annule donc si $A^t(Z - Ap) = 0$

$$A^t Ap = A^t$$

si $A^t A$ est inversible (vrai si rang(A)=r) alors $\hat{p} = (A^t A)^{-1} A^t Z$.

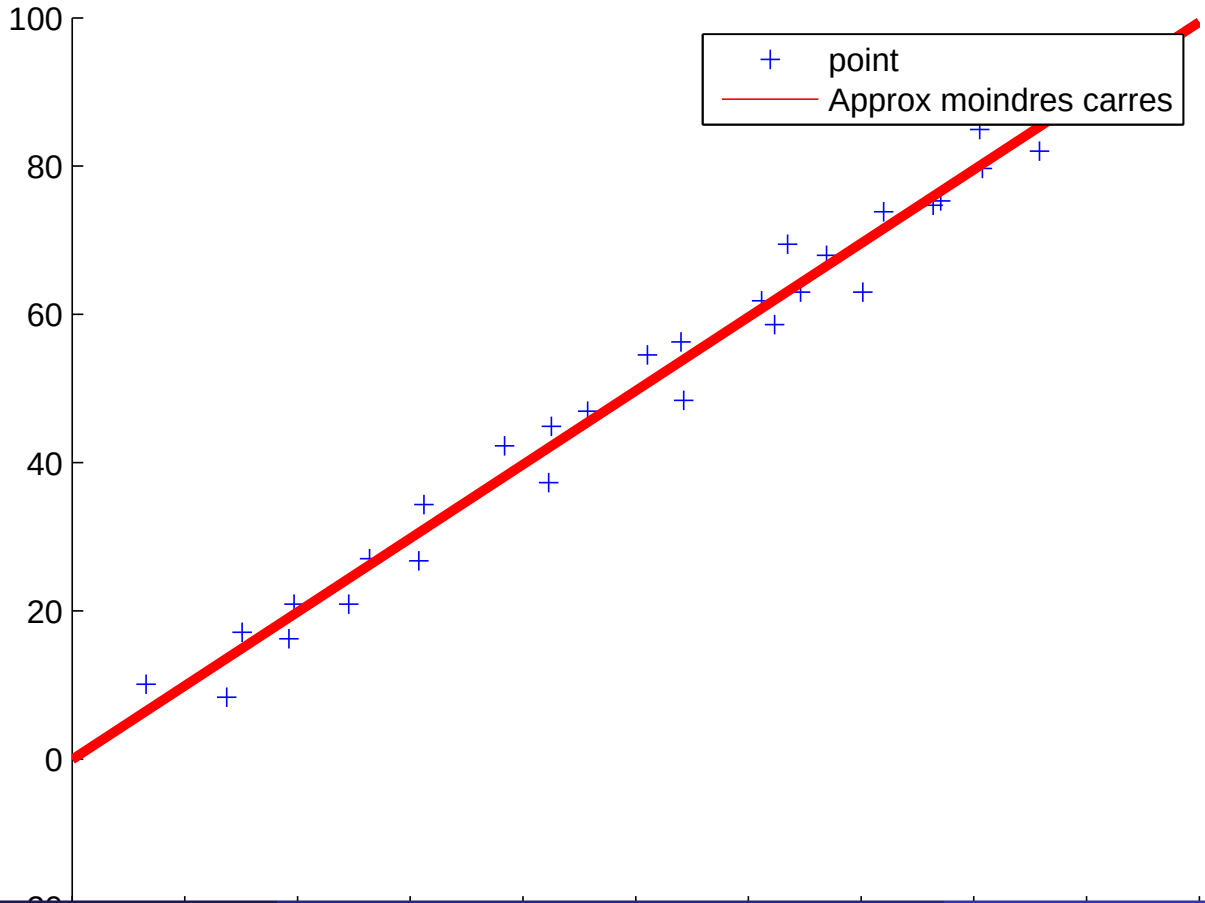
Solution des moindres carrés: compléments

- on peut choisir une norme significative du problème, par exemple $||Z||^2 = \sum \frac{1}{\sigma^2} z_i^2$ ou plus généralement $||Z||^2 = Z^t \Lambda^{-1} Z$ où Λ est la matrice de covariance des mesures.
  On a l'estimation
  $$\hat{p} = (A^t \Lambda^{-1} A)^{-1} A^t \Lambda^{-1} Z$$
- l'erreur d'estimation $cov(p - \hat{p})$ est $(A^t \Lambda^{-1} A)^{-1}$.

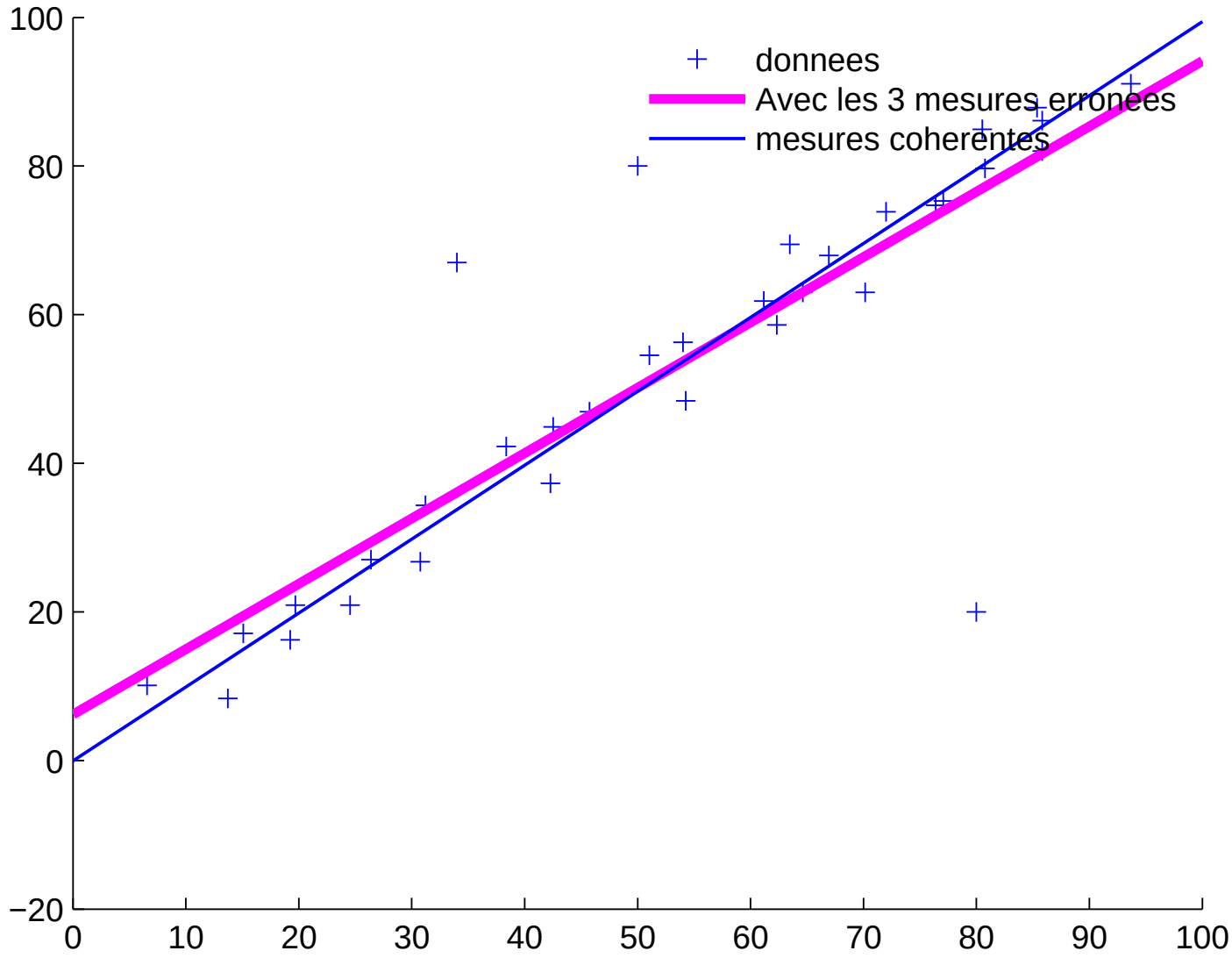
Exemple:Approximation d'une droite

Equation: $y = ax + b$

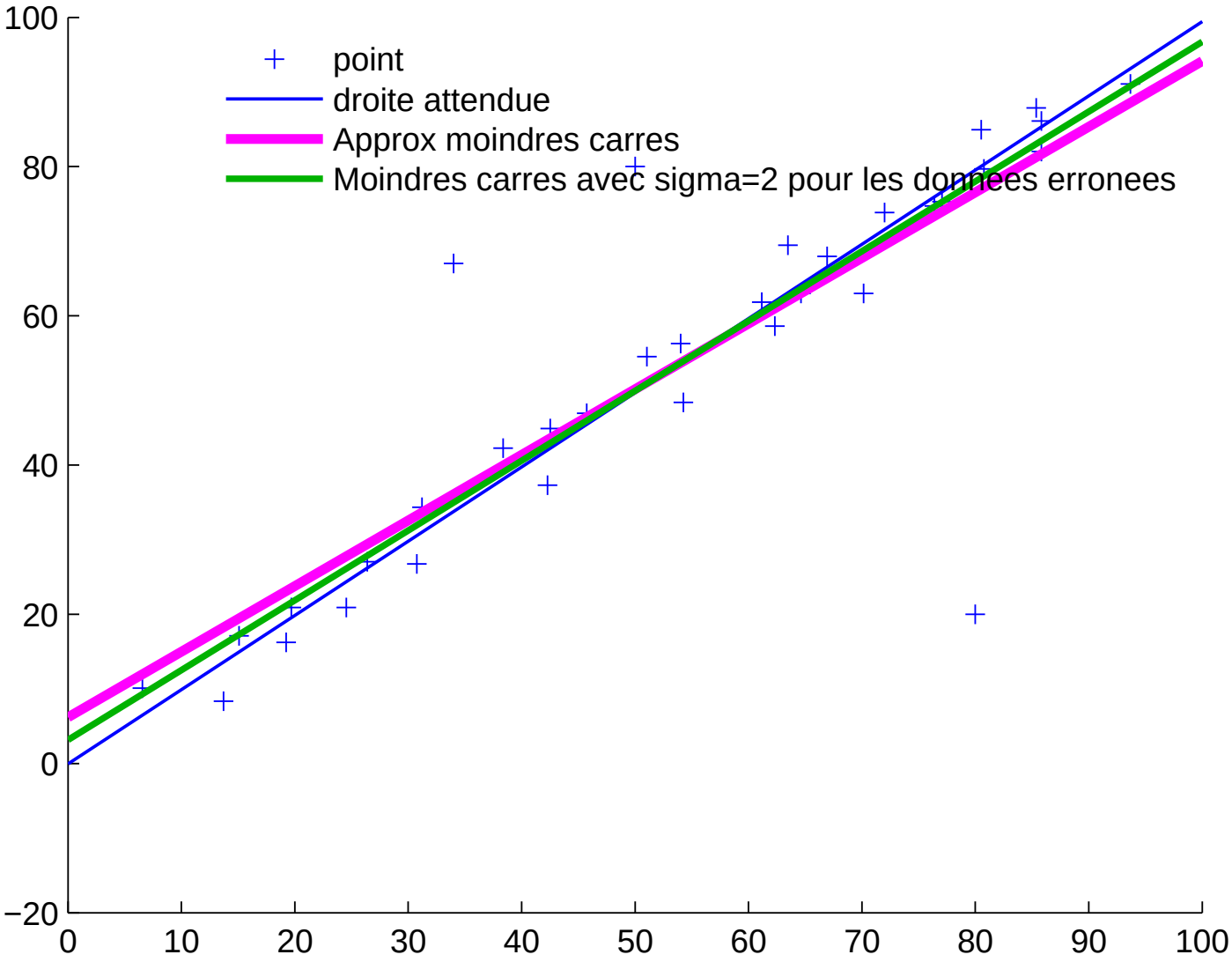
$$p = [a, b]^t, z = [y_1, .., y_n]^t, \; A = \begin{bmatrix} x_1 & 1 \\ .. & .. \\ x_n & 1 \end{bmatrix}$$



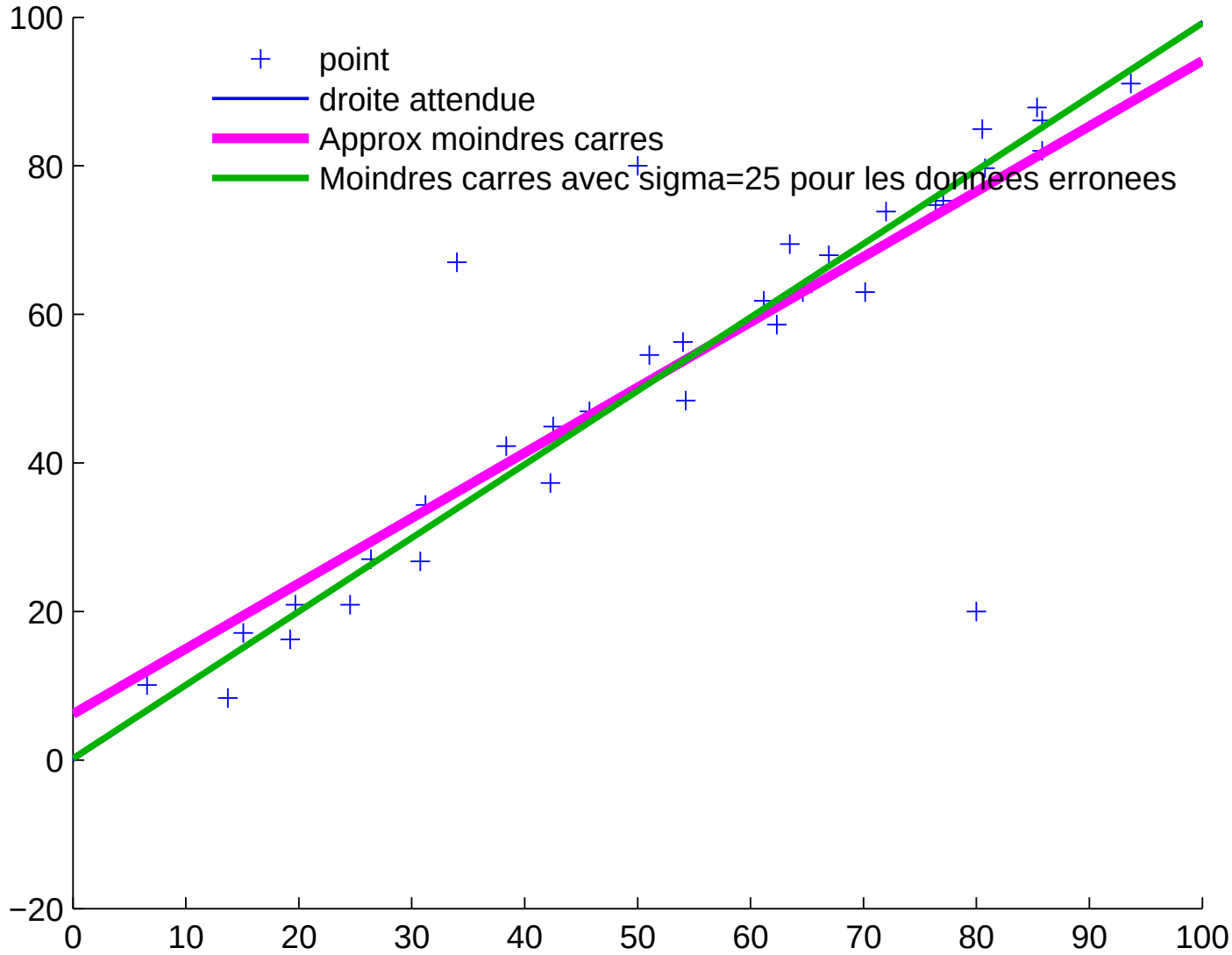
Avec des données incertaines



Prise en compte des incertitudes



Prise en compte des incertitudes



Exemple:Approximation B splines

Soient $(t_i, x_i)_{i \leq n}$ l'ensemble des points à approximer sur la base de splines
On cherche α_j

$$f(t) = \sum_{j=1}^{j=m} \alpha_j B_j(t)$$

tel que $f(t)$ approxime au mieux $(t_i, x_i)_{i \leq n}$
On cherche donc $\alpha_{j\, j \leq m}$ minimisant

$$\sum_{i=1}^{i=n} (\sum_{j=1}^{j=m} \alpha_j B_j(t_i) - x_i)^2$$

Ici $p = \begin{bmatrix} \alpha_1 \\ \dots \\ \alpha_n \end{bmatrix}$, Estimation classique aux moindres carrés avec

$$A = \begin{bmatrix} B_1(t_1) & \dots & B_m(t_1) \\ \dots & \dots & \dots \\ B_1(t_n) & \dots & B_m(t_n) \end{bmatrix} \quad Z = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Cette matrice est creuse car $B_j(x)$ a un support fini (nulle en dehors de ce support)

Moindres carrés et contraintes

Ce paragraphe s'inspire du rapport de recherche [zhang 95]. On considère le problème d'approximer au mieux une conique. Problème: approximer au mieux un ensemble de points 2D $m_i = (x_i, y_i)$ par une conique d'équation:

$$Q(x, y) = Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$$

(ellipse $\rightarrow$ $B^2 - AC < 0$). Si les données sont bruitées le système $Q(x_i, y_i) = 0$, $i = 1..n$ n'a pas de solution. $\rightarrow$ minimiser une fonction d'objectif:

$$r = \sum_{i=1}^{n} Q^2(x_i, y_i)^2$$

Existence d'une solution triviale $A = B = C = D = E = F = 0$. (représentation non minimale) $\rightarrow$ normaliser le problème

Normaliser avec $A + C=1$

Pourquoi?
plusieurs choix possibles, par exemple F=1...
mais cela suppose que la conique ne passe pas par l'origine
Pour une ellipse, A et C sont positifs et non nuls, donc poser $A + C = 1$ n'exclut aucun cas de figure.
alors $p = [A, B, D, E, F]^t$. le problème s'écrit

$$[x_i^2 - y_i^2, 2x_i y_i, 2x_i, 2y_i, 1]p = -y_i^2$$

Equation matricielle

$$Ap = b$$

Minimiser

$$r(p) = (Ap - b)(Ap - b)^t$$

dérivée $r'(p) = 2A^t(Ap - B) \rightarrow p = (A^t A)^{-1} A^t b$

Cas général: Normaliser avec $||p|| = 1$

$$p = [A, B, C, D, E, F]^t \quad avec \quad A^2 + B^2 + C^2 + D^2 + E^2 + F^2 = 1$$

Equation matricielle

$$Ap = 0 \ avec \ ||p|| = 1$$

et

$$A = \begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2x_1 & 2y_1 & 1 \\ . & . & . & . & . & . \\ x_n^2 & 2x_ny_n & y_n^2 & 2x_n & 2y_n & 1 \end{pmatrix}$$

Minimiser $(Ap)^t(Ab) = p^tA^tAp$ sous la contrainte $||p|| = 1$.
solution: le vecteur propre de A^tA correspondant à la plus petite valeur propre.
Remarque: Efficace pour les problèmes pour lesquels le choix de la normalisation n'est pas intuitif.

Le critère à optimiser...

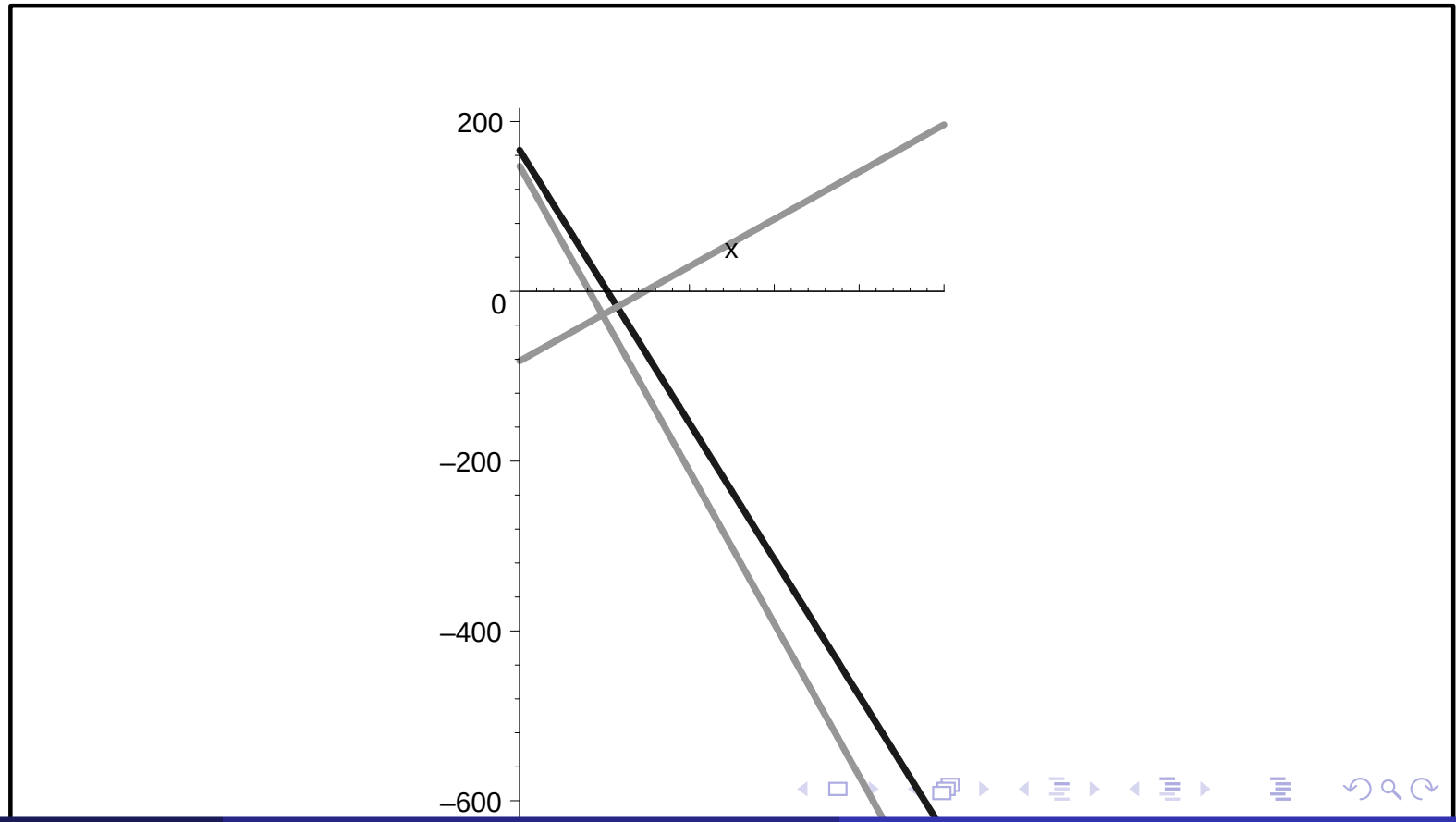
Dans le cas de l'ajustement, plusieurs mesures de l'adéquation modèle/mesure peut être envisagées

- le résidu algébrique $(y_i - ax_i - b)^2$: mais cette mesure dépend du repère choisi dans le plan (résultats non invariants aux rotations des données)
- La valeur absolue du résidu $|y_i - ax_i - b|$... mais non dérivable $\rightarrow$ le calcul analytique du minimum est impossible
- la distance **géométrique** point/droite $\frac{|ax_i+by_i+c|}{\sqrt{a^2+b^2}}$... Mais cette formulation n'est plus linéaire en $p \rightarrow$ minimisation itérative. Par contre elle est invariante aux rotations des données.

Adopter un compromis permettant d'approximer au mieux la distance géométrique tout en gardant une estimation directe du résultat

Invariance aux transformations des coordonnées images

Souhait: si on fait subir une transformation T aux données, on aimerait obtenir comme résultat $T(\mathcal{C})$.
Avec les distances algébriques, cette propriété d'invariance n'est en général pas vérifiée: exemple de l'approximation par une droite y=ax+b



Moindres carrés pondérés

Utiliser une autre fonction de mesure $f_i' = f_i/|\nabla f_i|$ où $f_i = f(x_i, y_i)$ est le résidu. C'est une approximation assez bonne de la distance géométrique permettant une implémentation de type moindres carrés.
formule exacte pour la droite:

$$f_i/|\nabla f_i = |ax + by + c|/\sqrt{a^2 + b^2}$$

minimiser

$$\sum_i f_i^2/|\nabla f_i|^2 = (Ap - b)^t \underbrace{\begin{bmatrix} |\nabla f_1|^2 & 0 & .. & 0 \\ . & & . & . \\ 0 & & .. & .. & |\nabla f_n|^2 \end{bmatrix}^{-1}}_{W} (Ap - b)$$

Moindres carrés pondérés

minimiser

$$\sum_i f_i^2/|\nabla f_i|^2 = (Ap-b)^t \underbrace{\begin{bmatrix} |\nabla f_1|^2 & 0 & .. & 0 \\ . & & . & . \\ 0 & & .. & .. & |\nabla f_n|^2 \end{bmatrix}^{-1}}_{W} (Ap-b)$$

Ici W est une fonction de $p \rightarrow$ pas de solution explicite en général. Algorithme numérique (minimisation d'une fonction non convexe) ou utiliser un moindre carré itératif calculant une estimation p_k qui converge vers p

- Déterminer une estimée initiale p_0 de p en utilisant un *moindre carré classique*.
- Tant que $|p_k - p_{k-1}| > seuil$:
  calculer $W(k)$ en utilisant l'estimation courante de p_k
  $p_{k+1} = (A^t W_k^{-1} A)^{-1} A^t W_k^{-1} b$

Part III

L'analyse en composantes principales (ACP)

Objectif de l'ACP

Objectif: résumer l'information apportée par un grand nombre de variables par un nombre restreint de **nouvelles variables**.

- Cette analyse se distingue de l'analyse factorielle où on souhaite seulement identifier parmi les variables initiales celles qui contribuent à expliquer le phénomène.
- l'ACP est très utilisée en RF pour modéliser un phénomène à partir d'une base d'exemple à l'aide d'un petit nombre de variables qui sont combinaison linéaire des variables initiales (et qui n'ont donc pas forcément de sémantique)

Méthode de l'ACP

soit un tableau X de n individus caractérisé par p variables. x_i^j est la valeur de la j^{ieme} variable sur le i^{ieme} individu. Chaque individu est pondéré par un poids p_i, $\sum p_i = 1$.
but déterminer de nouvelles variables décrivant les individus en perdant le moins possible d'information $\rightarrow$ déterminer un tableau Y de taille $n \times q$ avec $q < p$.
Technique: projeter le nuage sur un espace affine de dimension q de façon à ce que le nuage soit déformé le moins possible en projection.
remarque: d'autres critères (avec d'autres hypothèses a priori peuvent être envisagés, voir l'ACI)

Critère d'optimisation

$$\mathbf{x_i} = \begin{pmatrix} x_i^1 \\ . \\ . \\ x_i^p \end{pmatrix} \qquad \mathbf{x^j} = \begin{pmatrix} x_1^j \\ . \\ . \\ x_n^j \end{pmatrix}$$

On cherche un espace affine W de dimension q de base orthonormée $\mathbf{u_1}, .., \mathbf{u_q}$ d'origine $\underline{a}$
la projection orthogonale du vecteur $\mathbf{x_i}$ est donnée par

$$\hat{\mathbf{x}}_\mathbf{i} = \underline{a} + \sum_{k=1}^{k=q} y_i^k \mathbf{u_k}$$

Déformation:

$$I_W = \sum_{i=1}^{i=n} p_i d^2(\mathbf{x^i}, \hat{\mathbf{x}}_\mathbf{i})$$

Problème: déterminer $W = \underline{a} + [\mathbf{u_1}, .., \mathbf{u_q}]$ tel que I_W soit minimal.

Choix de l'origine $\underline{a}$

Prop: $\underline{a}$ est le centre de gravité du nuage des individus.
preuve: soit W_a et W_g deux espaces affine parallèles passant respectivement par a et g, centre de gravité du nuage. Alors, d'après le théorème de Huyghens,

$$I_{W_a} = I_{W_g} + d^2(g, W_a)$$

$\rightarrow$ l'inertie est minimale pour un sous espace contenant le centre de gravité du nuage.
Conséquence: on peut supposer les données centrées. On suppose dans la suite $g = 0$.

Détermination des u_i

Prop:

$$I_{W_O} = \sum_{i=1}^{i=n} ||p_i \mathbf{x}_i||^2 - \sum_{k=1}^{k=q} u_k^t V u_k \text{ ou } V = \sum_i p_i \mathbf{x}_i \mathbf{x}_i^t$$

preuve:

$$I_{W0} = \sum_{i=1}^{i=n} p_i ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2$$

comme $\hat{\mathbf{x}}_i$ et $\mathbf{x}_i - \hat{\mathbf{x}}_i$ sont orthogonaux, $||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 = ||\mathbf{x}_i||^2 - ||\hat{\mathbf{x}}_i||^2$ or $\hat{\mathbf{x}}_i = \sum_{k=1}^{k=q} < \mathbf{x}_i, u_k > u_k$ donc,

$$\sum p_i ||\hat{\mathbf{x}}_i||^2 = \sum_{i=1}^{i=n} p_i < \mathbf{x}_i, u_k >^2 = \sum_{i,k} p_i u_k^t \mathbf{x}_i \mathbf{x}_i^t u_k = \sum_{k=1}^{k=q} u_k^t V u_k$$

avec $V = \sum p_i \mathbf{x}_i \mathbf{x}_i^t$ Le problème revient donc à maximiser $\sum_{k=1}^{k=q} u_k^t V u_k$.

Lemme
$\sum_{k=1}^{k=q} u_k^t V u_k$ est maximal lorsque les $u_{j1 \leq j \leq k}$ sont les k vecteurs propres de V associés aux k plus grandes valeurs propres.

Theorem

le sous espace W cherché est le sous espace affine passant par le centre de gravité du nuage engendré par les q vecteurs propres de la matrice V associés aux q plus grandes valeurs propres (axes principaux d'inertie).

on appelle k^{ieme} composante principale le vecteur $\mathbf{y}^k$ dont les composantes sont les projections des points du nuage sur le k^{ieme} axe d'inertie.
$Y_i^k =< \mathbf{x}_i, u_k >$.

Propriétés des composantes principales

1. La moyenne de chaque composante est nulle.

$$\sum p_i < x_i, u_k >=< \sum p_i x_i, u_k >=< g, u_k >= 0$$

car on a supposé les données centrées.

2. la variance de la k^{ieme} composante principale est égale à λ_k. *preuve*:

$$Variance(y^k) = \sum_{i=1}^{i=n} p_i (y_i^k)^2 = \sum p_i < \mathbf{x}_i, u_k >^2 = \sum_{i=1}^{i=n} p_i u_k^t \mathbf{x}_i \mathbf{x}_i^t u_k = u_k^t V u_k = \lambda_k$$

$\rightarrow$ la part d'inertie expliquée par la k^{ieme} composante principale est $\frac{\lambda_k}{(\lambda_1 + ... + \lambda_p)}$.

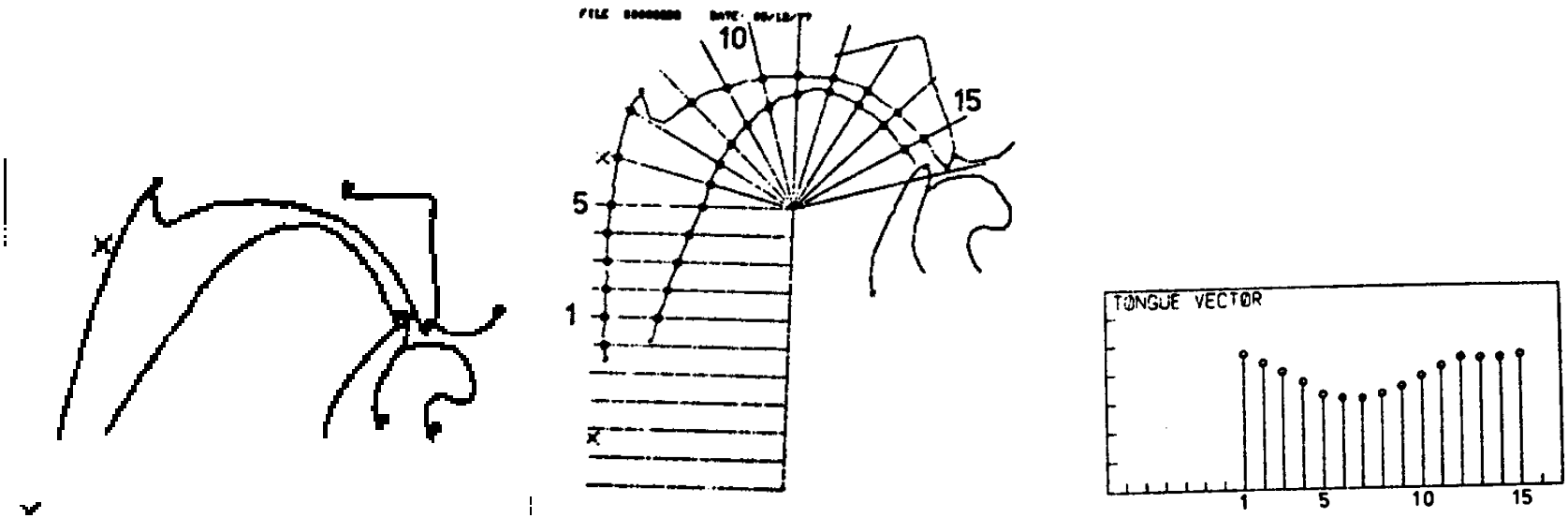
3. les composantes principales ne sont pas corellées entre elles.
comme les composantes principales sont d'espérance nulle,

$$\begin{aligned} cov(y^k, y^{k'}) &= E(y^k, y^{k'}) = \sum p_i y_i^k y_i^{k'} = \sum p_i < x_i, u_k >< x_i, u_{k'} > \\ &= \sum p_i u_k^t x_i x_i^t u_{k'} = u_k^t V u_{k'} = u_k^t \lambda' u_{k'} = \lambda u_k^t u_{k'} = 0 \end{aligned}$$

car la base est orthonormée.

Exemple: Analyse du conduit vocal [Mae79]

On souhaite élaborer un modèle du conduit vocal pour pouvoir synthétiser de la parole. Pour un locuteur donné, on dispose de la représentation des coupes de la langue pour 12 voyelles.
Objectif: compacter l'information pour obtenir un représentation paramétrique du conduit vocal.
Chaque tracé de langue X^j est décrit par 15 paramètres.
$X^j = [x_1^j, ..., x_{15}^j]^t$. on forme la matrice $V = \sum X^j (X^j)^t$



Conduit vocal

3 composantes principales suffisent à expliquer 98% de la variance. Soient V_1, V_2, V_3 ($V_i \in R^{18}$) les vecteurs propres correspondants.
Un modèle paramétrique du conduit vocal est donc $\sum_{i=1}^{i=3} \alpha_i V_i$, $\alpha_i \in R$.
Contribution de chacune des composantes principales au mouvement de la langue.

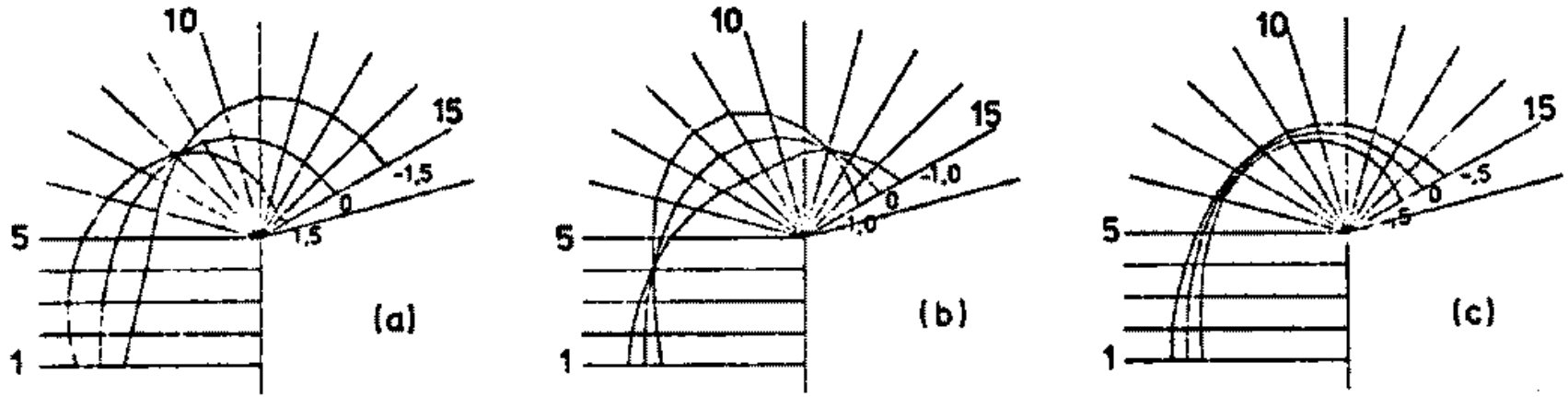


Fig. 5 Déplacement du contour de la langue à partir de sa position moyenne, qui est marquée par "0", dû à chacune des 3 premières composantes. Le mouvement dû à la 1-ière composante est représenté dans (a), à la 2-ième dans (b), et à la 3-ième dans (c).

.../...

Modéliser la variabilité des formes [CTCG95, CPJT96, CT99]

objectif: Construire des modèles de forme à partir d'exemples qui permettent

- d'analyser des formes nouvelles (appartiennent elles à l'ensemble?)
- de simuler de nouvelles formes appartenant à l'ensemble

idée: l'analyse des variations des formes dans l'ensemble d'apprentissage permet de construire un modèle qui intègre ces variations.

Contexte Les *formes* sont représentées par un ensemble de n marqueurs en dimension d (les exemples seront montrés en dimension 2). Ces marqueurs sont censés se correspondre entre les exemples (points spécifiques) $\rightarrow$ utilisation fréquente de points à forte courbure.

Un forme est représentée par un vecteur

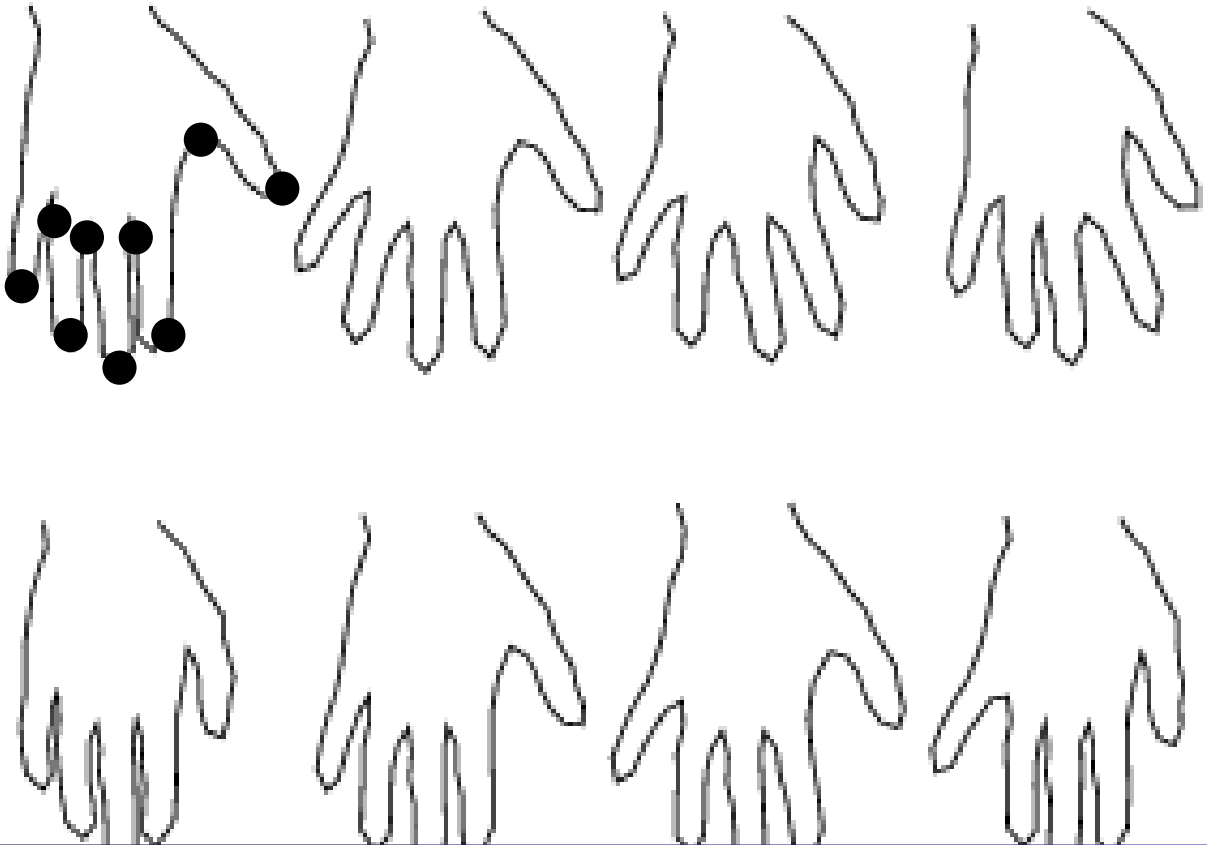
$$X = (x_1, ..., x_n, y_1, ..., y_n)$$

Modéliser la variabilité

Alignement des exemples

Afin d'enlever les variations qui sont attribuables à un mouvement global, les formes sont alignées dans un repère commun.

Exemples de la base d'apprentissage:



Modélisation de la variabilité par une ACP

Courbe moyenne :

$$\bar{X} = \frac{1}{s}\sum X^i$$

où X^i est la ieme courbe.
Covariance des données:

$$S = \frac{1}{s-1}\sum (X^i - \bar{X})(X^i - \bar{X})^t$$

Calcul des vecteurs propres correspondant aux q plus grandes valeurs propres de S.
Représentation de l'ensemble des données comme la variance de la i^{ieme} composante est λ_i, l'ensemble

$$\bar{X} + \sum \alpha_i u_i, \quad \alpha_i \in [-3\sqrt{\lambda_i}, 3\sqrt{\lambda_i}]$$

est en général une assez bonne description de la base de données.

Modélisation de la variabilité par une ACP

RQ: on suppose en général que la distribution des α_i est gaussienne. Ce n'est pas toujours compatible avec certaines variations non linéaires de forme $\rightarrow$ générer les α selon des distributions plus complexes.

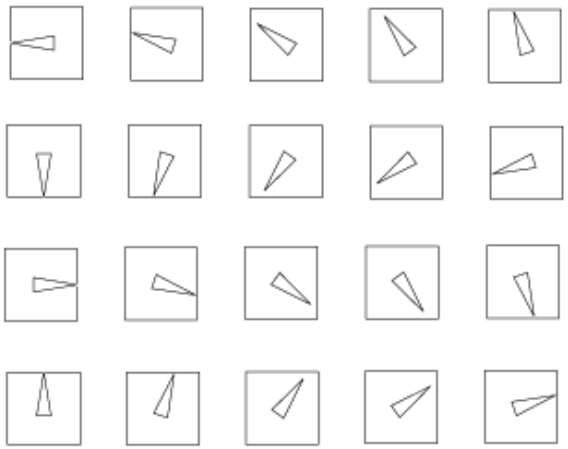


Figure 4.9: Examples from training set of synthetic shapes

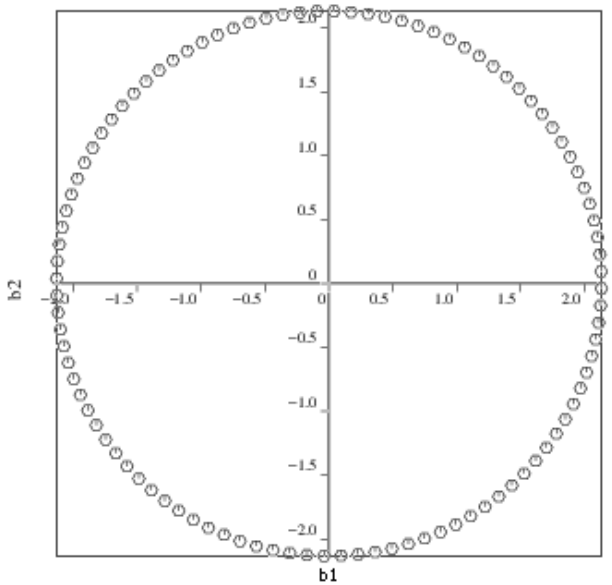


Figure 4.10: Distribution of **b**for 100 synthetic shapes

Choix du nombre de modes

- soit par la variance expliquée (doit etre suffisante)
- soit en s'assurant que les éléments de la base sont approximés avec une précision suffisante

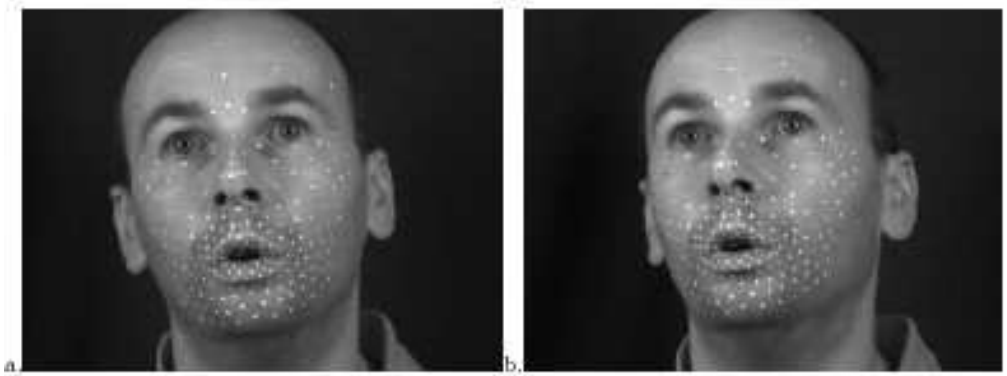
Mode 1

Mode 2

Mode 3

Exemple de la tête parlante

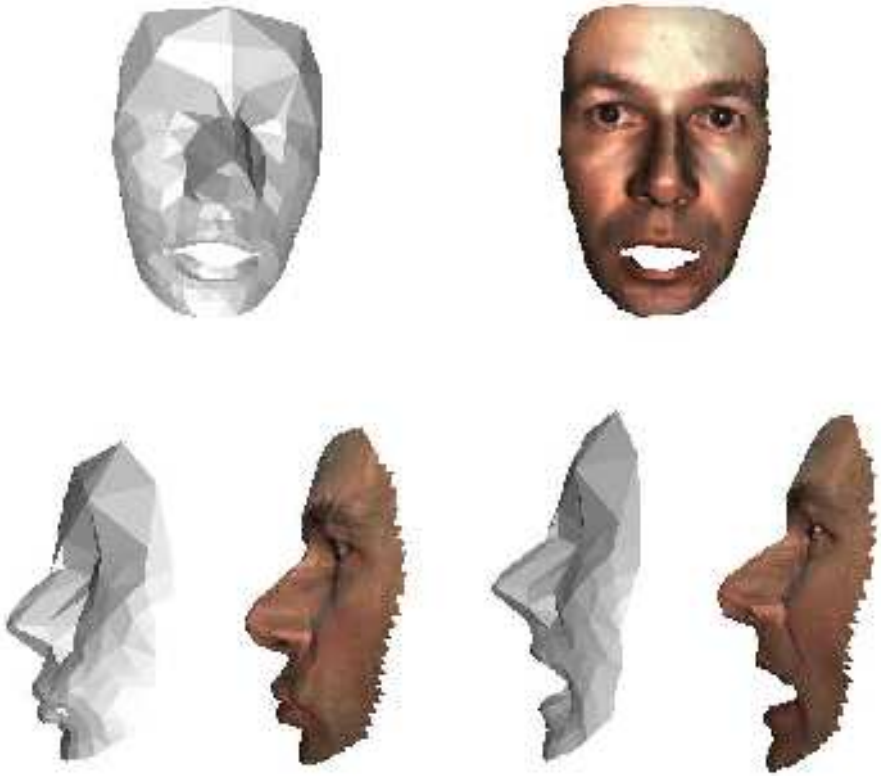
- Objectif: apprendre à commander une tête avec un petit nombre de paramètres
- apprentissage de la dynamique de la tête sur un corpus de 20 mn par un système stéréo
  permettant de recueillir un maillage de la tête à une cadence de 200 fps.



- Effectuer une ACP sur ces maillage (200 points par maillage)

Exemple de la tête parlante

Inertie associée aux modes: 51.1, 24, 6.45, 1.85,1.72,1.37,1.17,0.81,0.71,0.53,0.45



Reconnaissance

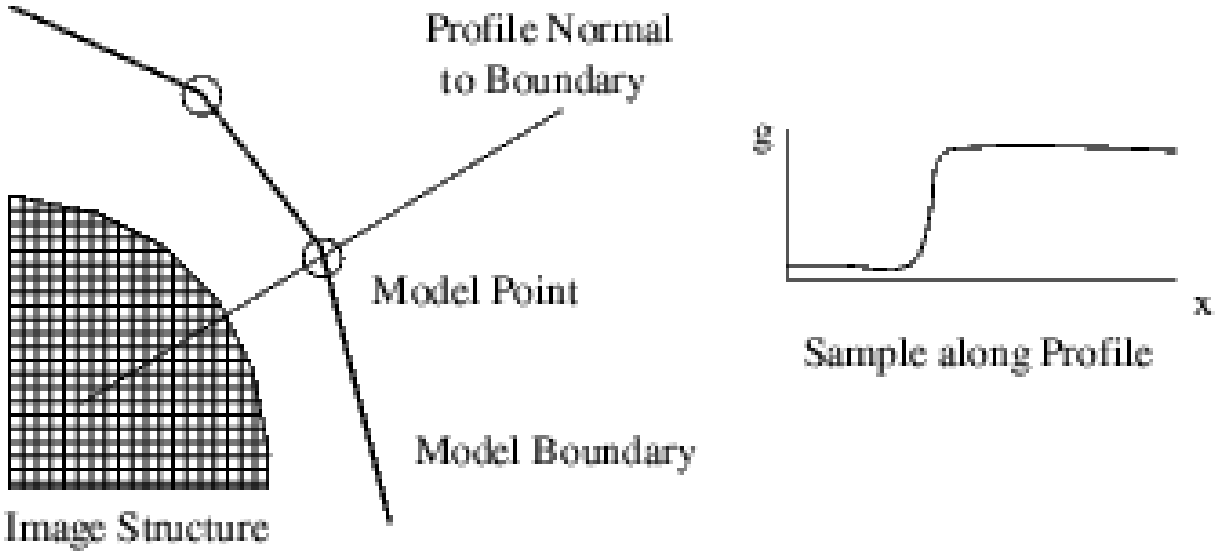
Tester l'appartenance d'une nouvelle forme à la base de données
Deux résidus sont en général pris en compte:

- l'adéquation de la nouvelle forme avec les composantes principales
- le résidu de la mesure vis à vis des résidus de la base d'apprentissage.

Identifier grâce au modèle trouvé([cootes])

But: trouver la localisation d'un visage grâce au modèle de forme.

1. **Trouver des correspondances modèle/image**



Identifier grâce au modèle trouvé([cootes])(2)

2. Calculer une transformation rigide (pour l'orientation) et les paramètres du modèle superposant au mieux le modèle generé et les points trouvés dans l'image

Figure 7.5: Search using Active Shape Model of a face

minimiser sur T et α_i

$$d(PointsDetectes, T(\sum_{1}^{q} \alpha_i u_i))$$

La fonction de coût est non linéaire, donc minimisation itérative à partir d'une position initiale.

Pour poursuivre sur l'ACP

- Rendre l'ACP robuste à la présence de données erronées:**[torre01]** Fernando De la Torre and Michael Black. Robust principal component analysis for computer vision. In *Proceedings of 8th International Conference on Computer Vision, Vancouver (Canada)*, pages 362–369, 2001.
- Permettre l'ACP dans le cas de données manquantes: **[shum95]** H.Y. Shum, K.Ikeuchi, and R.Reddy, Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on PAMI*, 17(9):854–867, 1995.

Part IV

L'analyse en composantes indépendantes

Quelques remarques sur l'ACP

- l'ACP cherche à extraire des composantes orthogonales, ce qui n'a pas de sens physique a priori
- l'ACP extrait des composantes décoréllées, ce qui ne signifie pas qu'elles soient indépendantes, sauf dans le cas gaussien

L'analyse en composantes indépendantes cherche à extraire des composantes qui soient statistiquement indépendantes et non gaussiennes.

Indépendance et corrélation

Indépendance Deux v.a sont indépendantes si pour tout A et B

$$P(X = A \text{ et } Y = B) = P(X = A) * P(Y = B)$$

i.e la réalisation de A n'a aucune influence sur la réalisation de B.
covariance de deux v.a: $cov(X, Y) = E(X - E(X)) * (Y - E(Y))$
X et Y indépendantes implique cov(X,Y)=0.**L'inverse n'est pas vrai.**
Exemple: soit la variable aléatoire à deux dimensions (X,Y), de densité constante dans le cercle $x^2 + y^2 \leq 1$ et nulle ailleurs.
Par raison de symétrie $cov(X, Y) = 0$. donc X et Y ne sont pas corrélées. Mais ces variables ne sont pas indépendantes. Si elles l'étaient, on aurait

$$Pr[(X > 1/\sqrt{(2)}) \wedge (Y > 1/\sqrt{(2)})] = Pr(X > 1/\sqrt{(2)}) \times Pr(X > 1/\sqrt{(2)})$$

or le premier membre est nul, alors que le second est strictement positif.

Contexte de l'ACI

Au départ, un problème de séparation de sources: extraire d'observations vectorielles des composantes linéaires qui soient aussi *indépendantes que possible*.

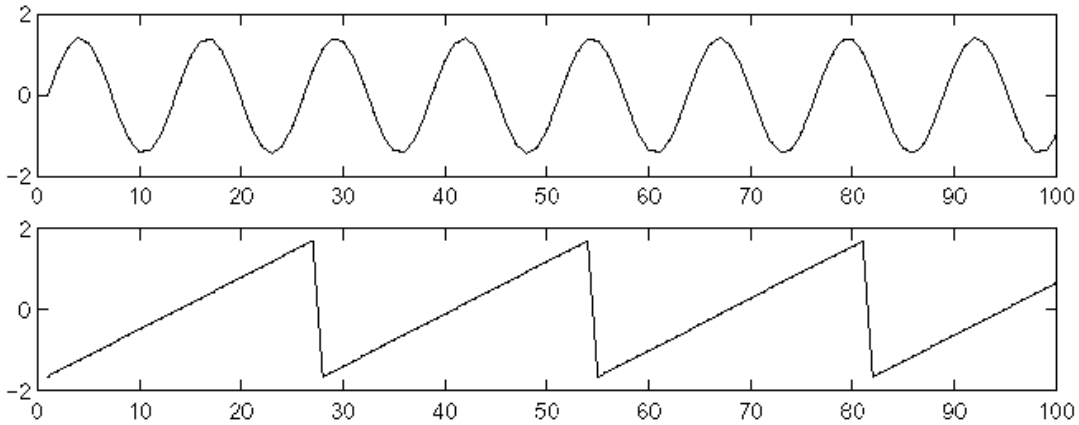
Exemple: soient deux personnes parlant simultanément et deux micros enregistrant la conversation à deux endroits de la salle. On note $x_1(t)$ et $x_2(t)$ les signaux enregistrés et $s_1(t)$ et $s_2(t)$ les signaux inconnus émis par les locuteurs. On a

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

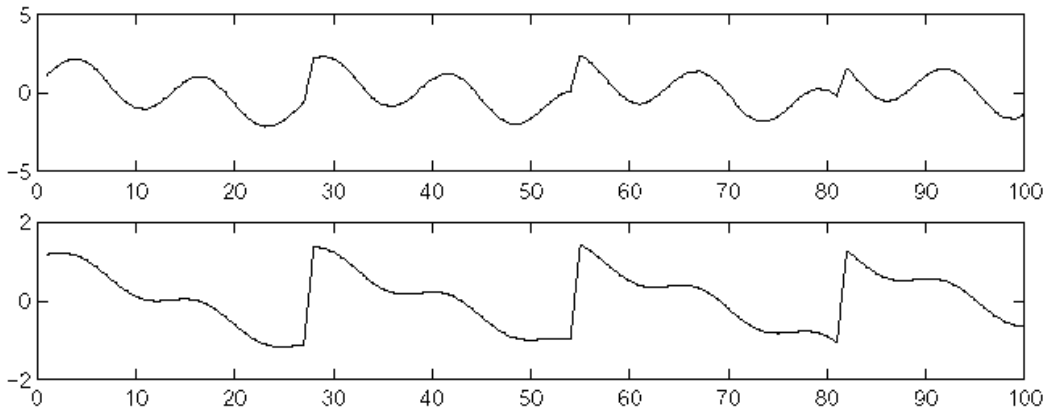
Problème: trouver les signaux initiaux s_1 et s_2 sachant que les a_{ij} sont aussi inconnues. $X = AS$

Remarque: les s_i ne peuvent être retrouvés qu'à un facteur près (on fixe donc $E(s_i^2) = 0$ et sans notion de hiérarchie (contrairement à l'ACP)

Exemple (tiré de Hyvarinen 00)

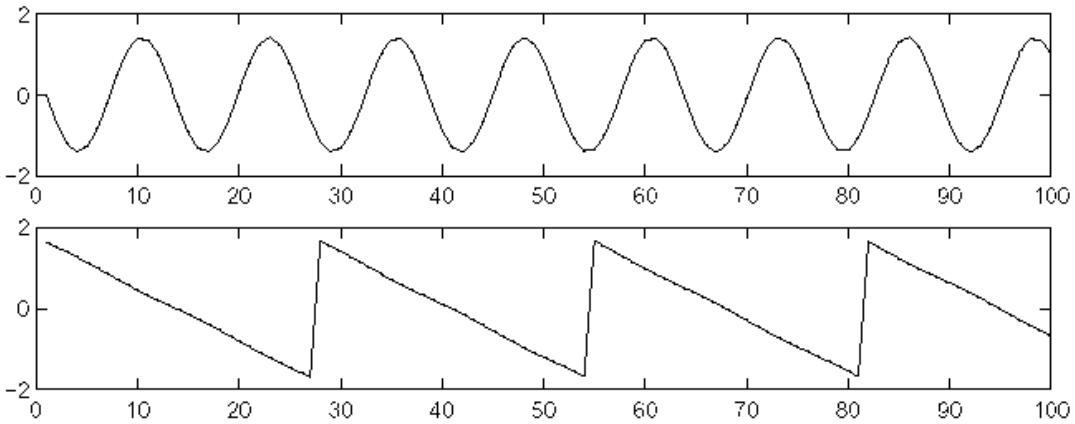


Les signaux initiaux

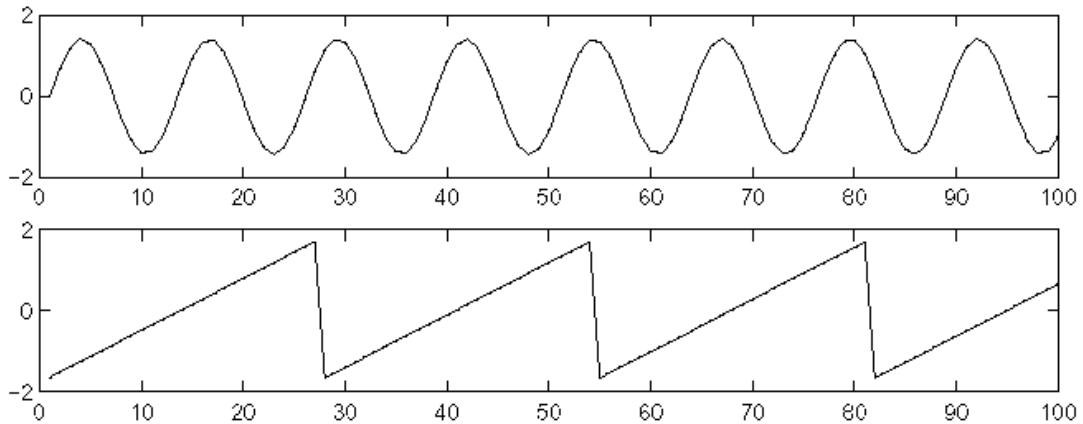


Les signaux mélangés: données d'entrée pour l'ACI

Exemple



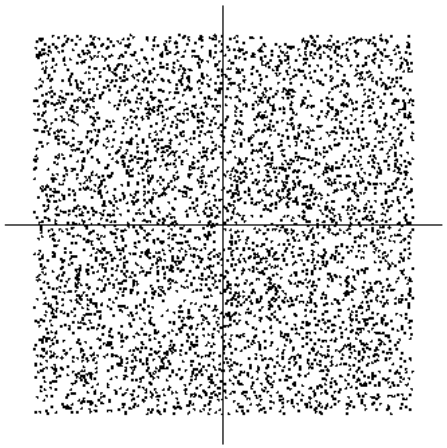
Resultats de l'ACI: les signaux sont retrouvés à facteur multiplicatif près.



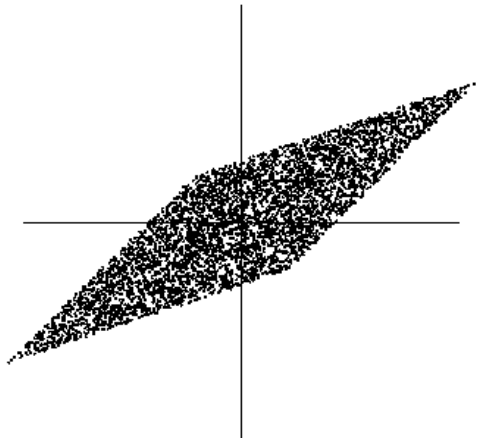
Vérité terrain: Les signaux initiaux

Une idée intuitive de l'utilisation de l'indépendance

Soien deux v.a s_1, s_2 indépendantes suivant une loi uniforme sur $[-\sqrt{3}, \sqrt{3}]$. On considère $X = AS$ avec $A = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$.



Densite de s_1, s_2

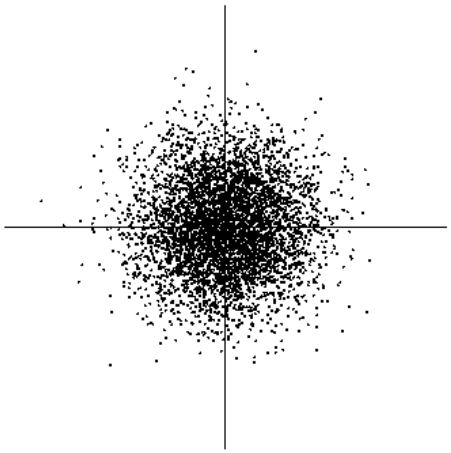


Densite de x_1, x_2

La loi de mélange donne les directions des colonnes de A (cotés du parallélogramme) $\rightarrow$ on peut donc résoudre intuitivement le problème, au moins dans le cas uniforme)

Pourquoi le cas gaussien est interdit

si (s_1, s_2) sont indépendants et gaussiens ($cov(s_1, s_2) = I$). Soit A une matrice orthogonale. Alors:
As est gaussien de densité $p = \frac{1}{2\pi} exp(-(x_1^2 + x_2^2)/2)$. puisque
$cov(As) = Acov(s)A^t = AA^t = I$
$\Rightarrow$ La densité du mélange ne contient aucune information sur A donc on ne peut pas identifier s_1 et s_2



La densité d'un couple de v.a gaussiennes indépendantes

Caractérisation de l'indépendance: maximiser la non gaussianité

théorème central limite: la somme de N variable indépendantes tend vers loi gaussienne quand $N \to \infty$
Interprétation: $\to$ la somme de variables non gaussiennes est plus proche d'une gaussienne que chacune des variables initiales.
Or $A^{-1}X$ doivent être indépendants.

Principe: On détermine les combinaisons linéaires $\sum b_i x_i$ qui sont des maxima locaux de **non gaussianité**
(si la C.L ne fournit pas l'un des s_i, il fournit une C.L des s_i dont quelque chose de plus gaussien d'après le théorème central limite).
Chaque maximum local fournit une composante indépendante.

Mesurer la non gaussianité

Kurtosis:

$$kurt(x) = E(x^4) - 3 * (E(x^2))^2$$

Si on suppose que $var(x) = 1$, $kurt(x) = E(x^4) - 3$
Pour une variable gaussienne, $E(y^4) = 3 * E(y^2)$. Donc $kurt(x) = 0$ pour une gaussienne.
La non gaussianité est souvent mesurée par $|kurt(x)|$ ou $kurt(x)^2$ en raison de sa simplicité et de ses bonnes propriétés ($kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2)$ et $kurt(\alpha x) = \alpha^4 kurt(x)$. Mais il existe d'autres mesures...
Des méthodes itératives d'optimisation sont ensuite utilisées pour maximiser $kurt(\sum b_i x_i)$.

Pour poursuivre sur l'ACI

- l'ACP décorelle les données alors que l'ICA les rend indépendantes
- les composantes indépendantes doivent être non gaussiennes pour que l'ACI fonctionne
- l'ICA utilise des moments d'ordres supérieurs à ceux utilisés pour l'ACP (statistiques du second ordre)
- Il n'existe pas UN algorithme pour l'ACI (au contraire de l'ACP) car il existe diverses façons d'exprimer l'indépendance et de la mesurer. Il y a ensuite diverses façons d'optimiser le critère d'indépendance
- l'objectif implicite de l'ACI est souvent de trouver des composantes physiquement significatives, ce qui est plus compliqué que l'ACP qui a des objectifs plus modestes mais plus faciles à atteindre

- pour des comparatifs ICA, PCA en terme de reconnaissance voir par exemple Draper 03 (Recognizing faces with PCA and ICA, CVIU 2003),

Haut: vecteurs propres de la PCA. Bas: base pour l'ICA

- un site consacré à l'ACI
  `http://www.cnl.salk.edu/~tony/ica.html`

Part V

Exemples

Reconnaissance de visages par ACP: eigenfaces [pentland]

Objectif: faire de la reconnaissance de visages à partir d'une base de données.
Le concept d'eigenfaces : Pentland, Turk, Darrel

- A partir de données normalisées (alignées, même échelle), faire une ACP sur la base d'images. Représenter la forme par l'ensemble des modes significatifs
- Reconnaissance: projeter la forme candidate sur la base de l'ACP. Regarder si les coeffcients sur la base sont statistiquement compatibles avec la variablité issues de l'ACP ($\alpha \in [-3\sqrt{\lambda_i}, 3\sqrt{\lambda_i}]$).

Reconnaissance de visages par ACP: eigenfaces [pentland]

Reconstruction d'un visage en utilisant 1, 2, 3 .... modes

Reconnaissance de visages par ACP

- Test sur une base de 5700 visages pour environ 3000 personnes
- ACP construite à partir de 128 visages
- 20 modes conservés

Reconnaissance de visages par ACP

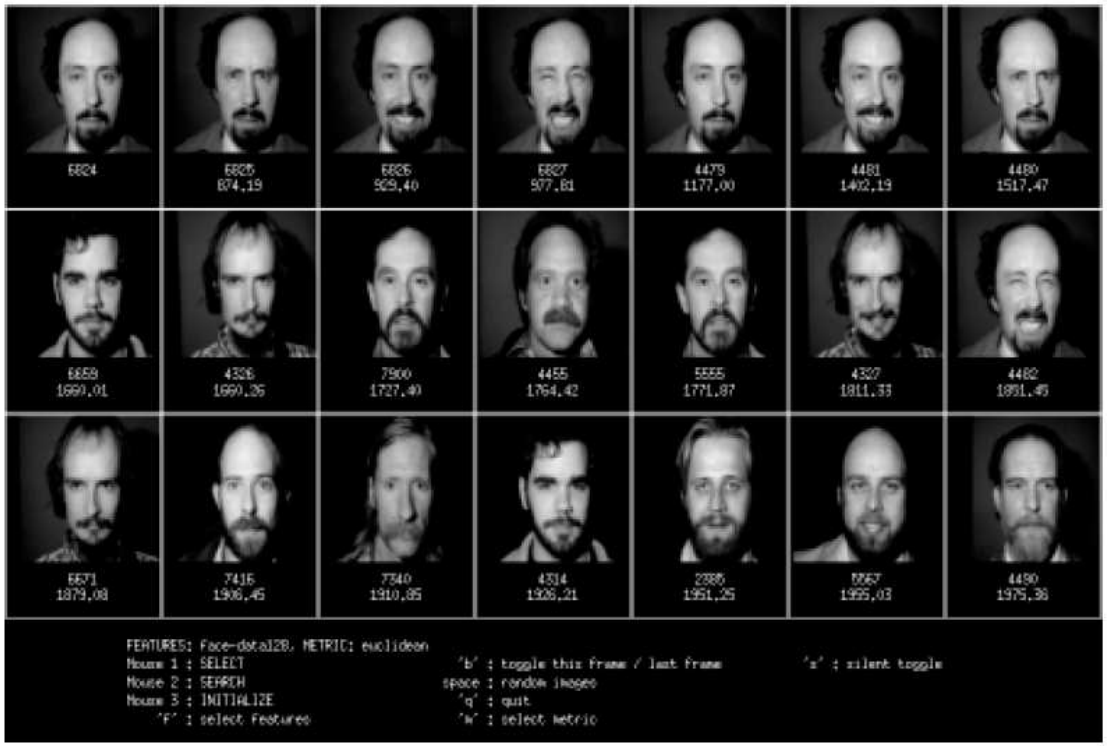


Figure 1: The face at the upper left was selected by the user; the remainder of the faces are the 20 most-similar faces found from among the entire 7,562 individuals in the database. Similarity decreases left to right, top to bottom. Note the ability to match an individual despite wide variations in expression.

Taux de reconnaissance de 95% pour 200 visages pris au hasard dans la base (reco si le visage le plus approchant est celui de la personne)

Reconnaissance de visages par ACP: améliorations

objectifs: rendre la reconnaissance plus robuste à des variations de positions et identifier des expressions

- Construire
  plusieurs modèles correspondant à des orientations différentes du visage

Reconnaissance de visages par ACP: améliorations

- Taux de reconnaissance de 90% pour les modèles utilisant les points de vues différents et de 88% pour le modèle classique.
- test d'extrapolation: apprentissage pour des vues de -90 a + 45 degrés et tests pour des vues de 68 et 90 degrés. Pour 68 degrés, reconnaissance de 83 et 78 %. Pour desvues à 90 degrés, taux de 50% et 43 %.

Reconnaissance temps réel (Viola & Jones)

Robust real time object detection, 2001, Viola & Jones
Un système temps réel de reconnaissance d'objets et illustré sur la reconnaissance de visages.

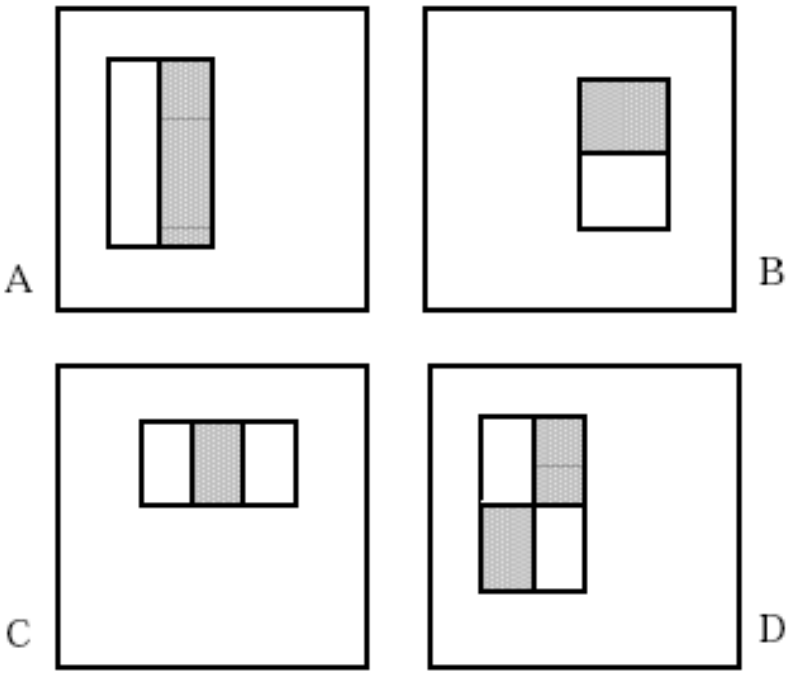
- des indices peu structurés (rectangles)
- utilisation de l'algorithme ADABOOST pour construire un classifieur selectionnant un petit nombre d'indices
- Une cascade de filtres améliorant l'efficacité de la détection en focalisant sur les régions prometteuses de l'images.

Apprentissage du classifieur

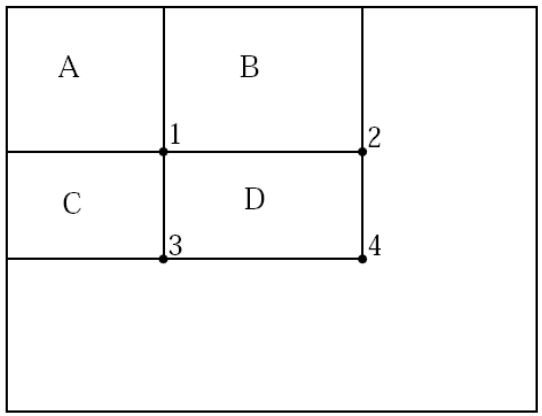
Données: des images de visage 24x24 (5000 exemples exemples positifs étiquetés 1) et des exemples négatifs (10000 exemples étiquetés 0).

Les données sont normalisées: visages de face à la même échelle.

Les filtres



la réponse des filtres est calculée facilement: 4+1 -(2+3)



les zones grisées sont comptées négativement.

$$reponse = \int_{regions\ \ grises} I(x,y)dxdy - \int_{regions\ \ blanches} I(x,y)dxdy$$

Pour une fenetre 1024x1024, il y a environ 180000 filtres possibles (en variant position, forme et taille).

Utilisation des filtres rectangulaires comme détecteur de visages

Soit f_j un rectangle. Le classifieur associé h_j est défini par f_j et un indice de parité p_j indiquant le sens de l'inégalité.

$$h_j(x) = \begin{cases} 1 \ si \ p_j f_j(x) < p_j \theta_j \\ 0 \ sinon \end{cases}$$

- Il est nécessaire de choisir un seuil et une parité pour chaque classifieur
- le classifieur choisi est celui minimisant l'erreur de classification

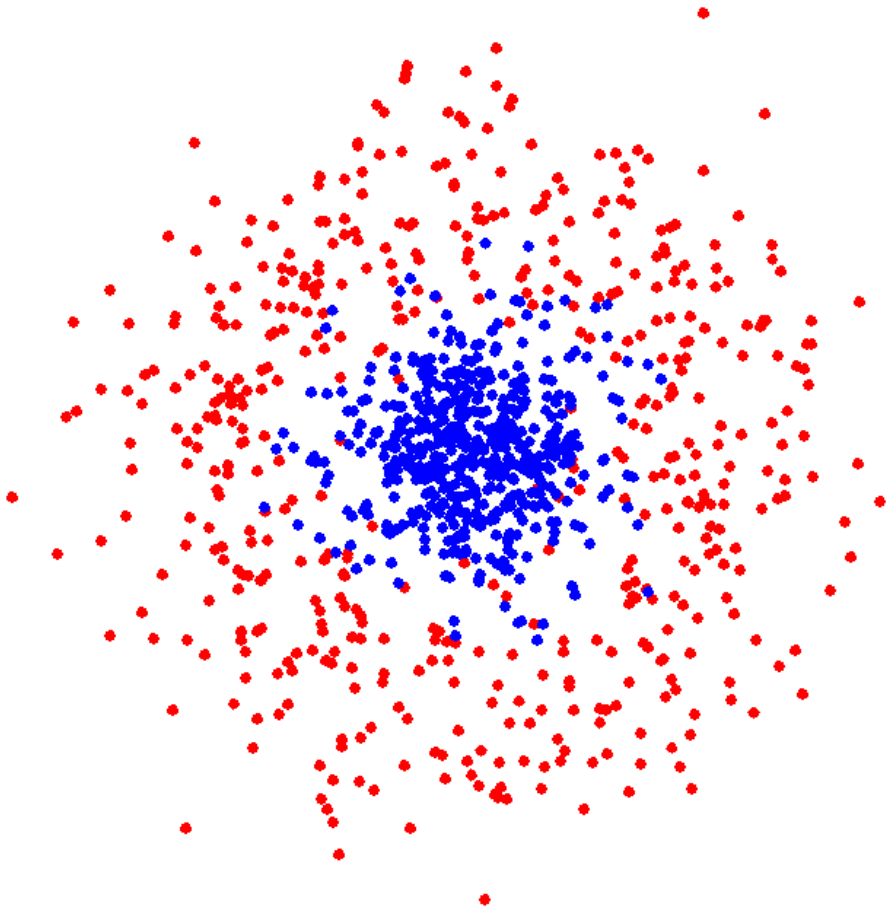
$$\sum_i \omega_i |h_j(x_i) - y_i|$$

Apprentissage du classifieur: Adaboost

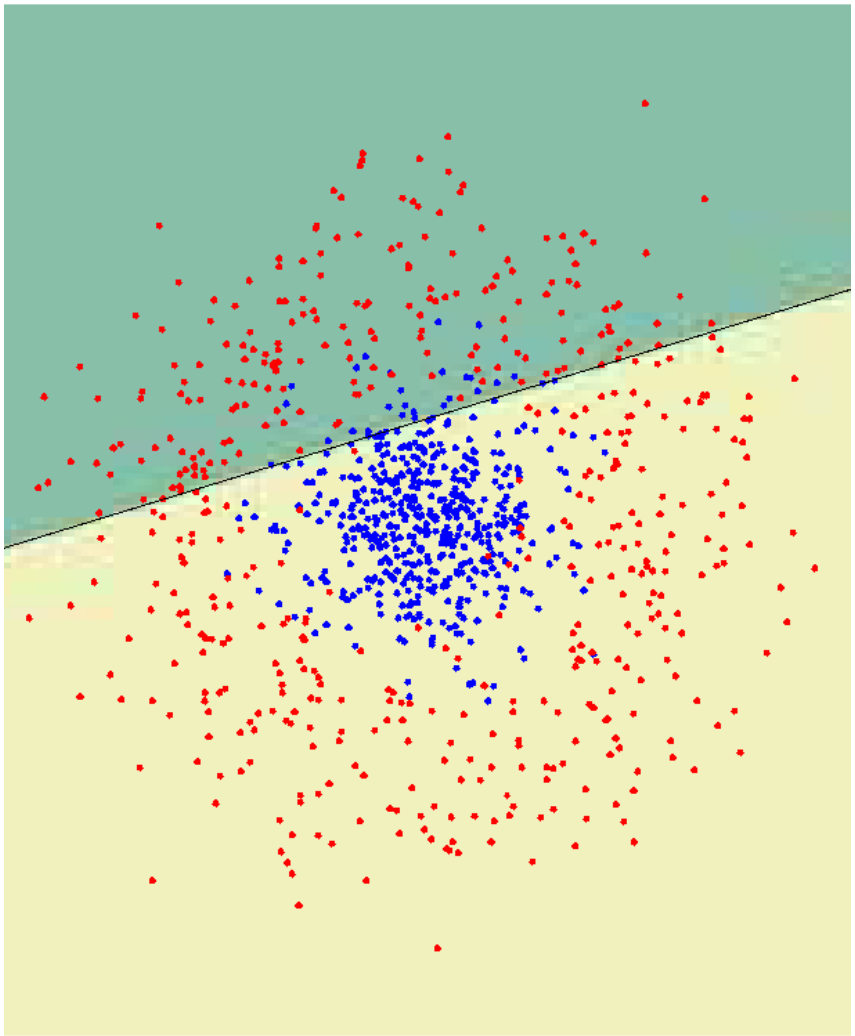
Principe: combinaison de classifieurs élémentaires qui vont fournir un classifieur fort

- Détermination d'un premier classifieur séparant au mieux les données (visages/ non visages)
- les exemples mal classifiés sont pondérés plus fortement et on poursuit la classification
- Classifieur final $h(x) = \sum \alpha_t h_t(x)$

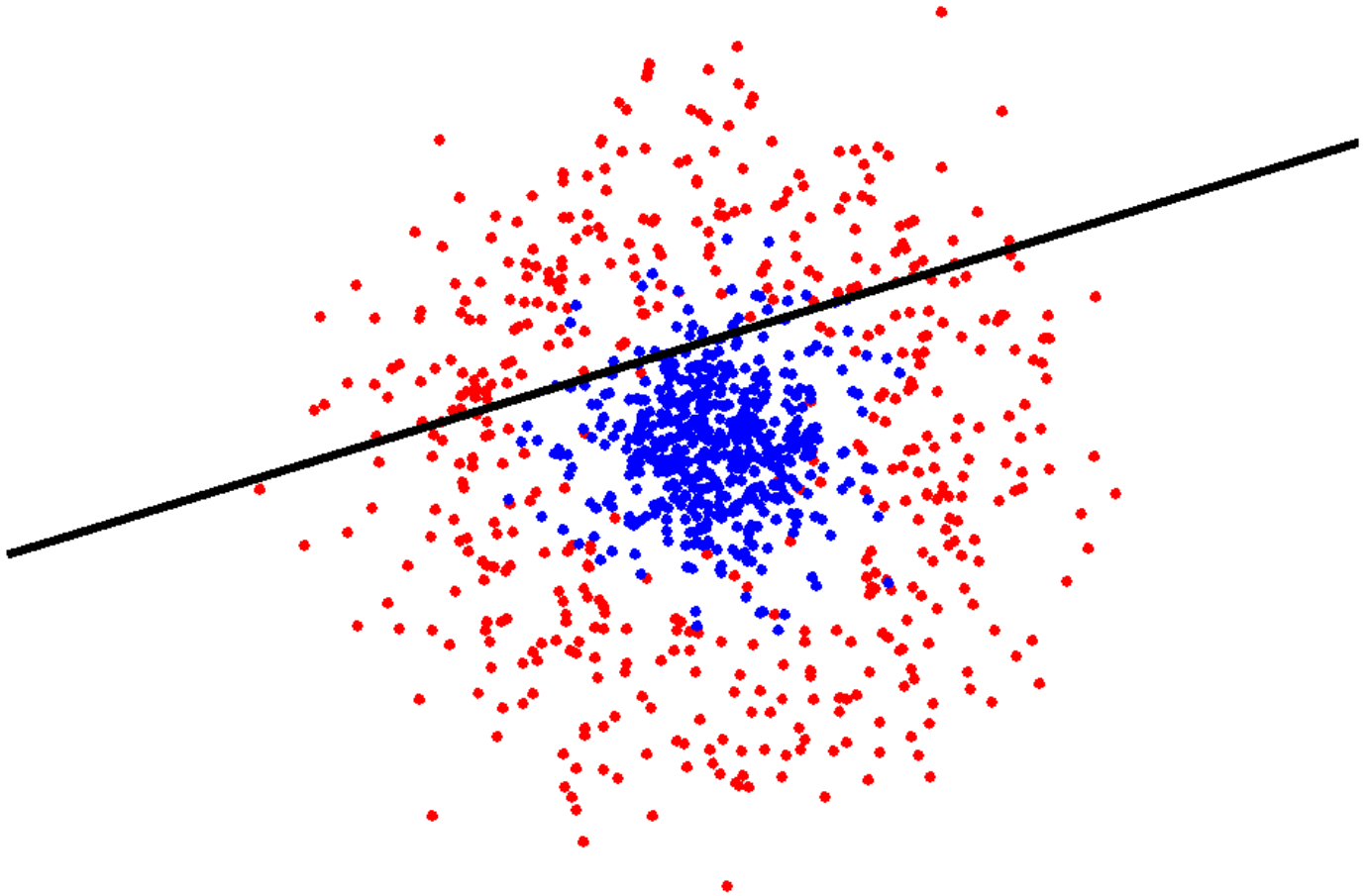
Adaboost



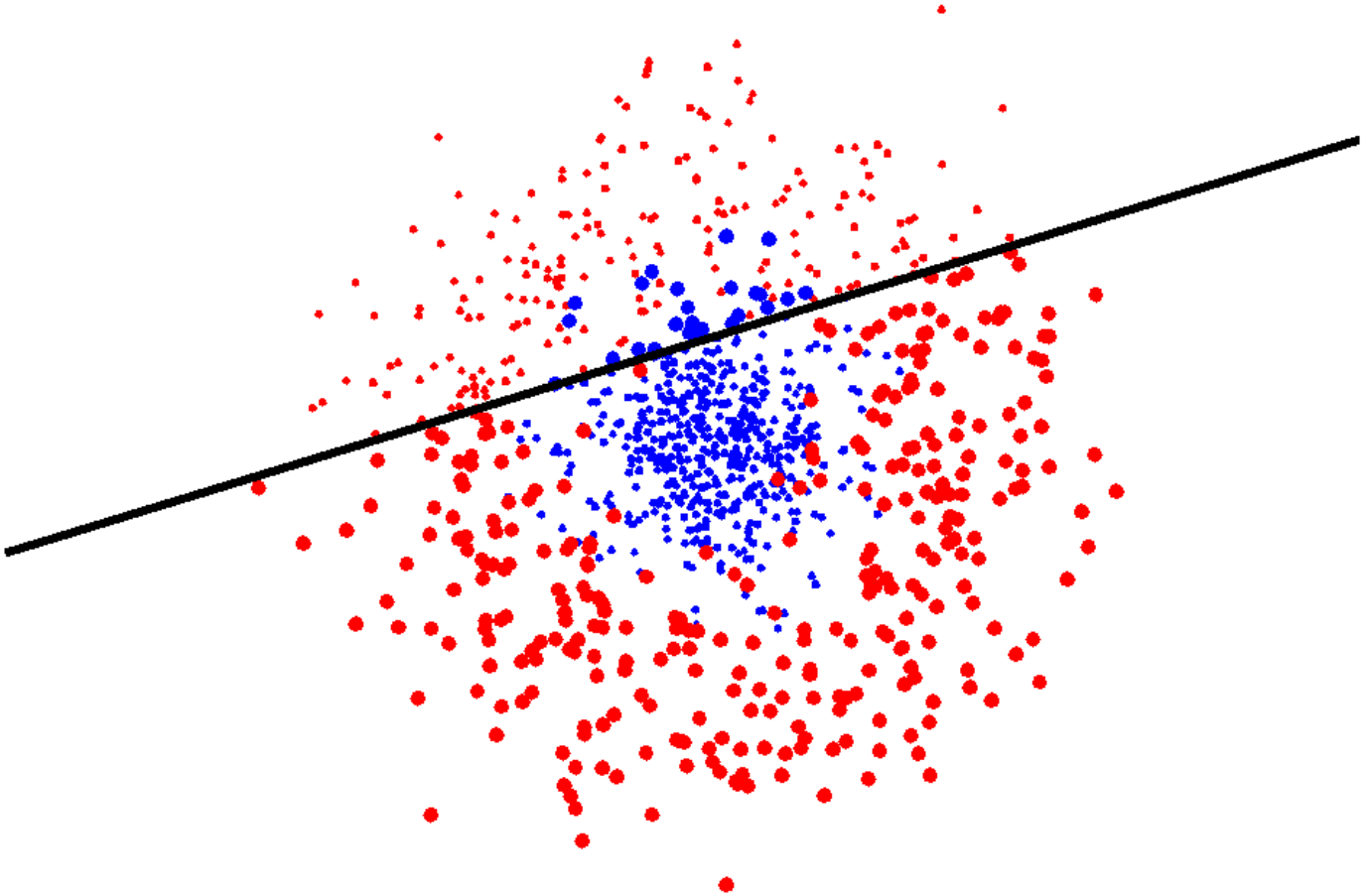
Adaboost



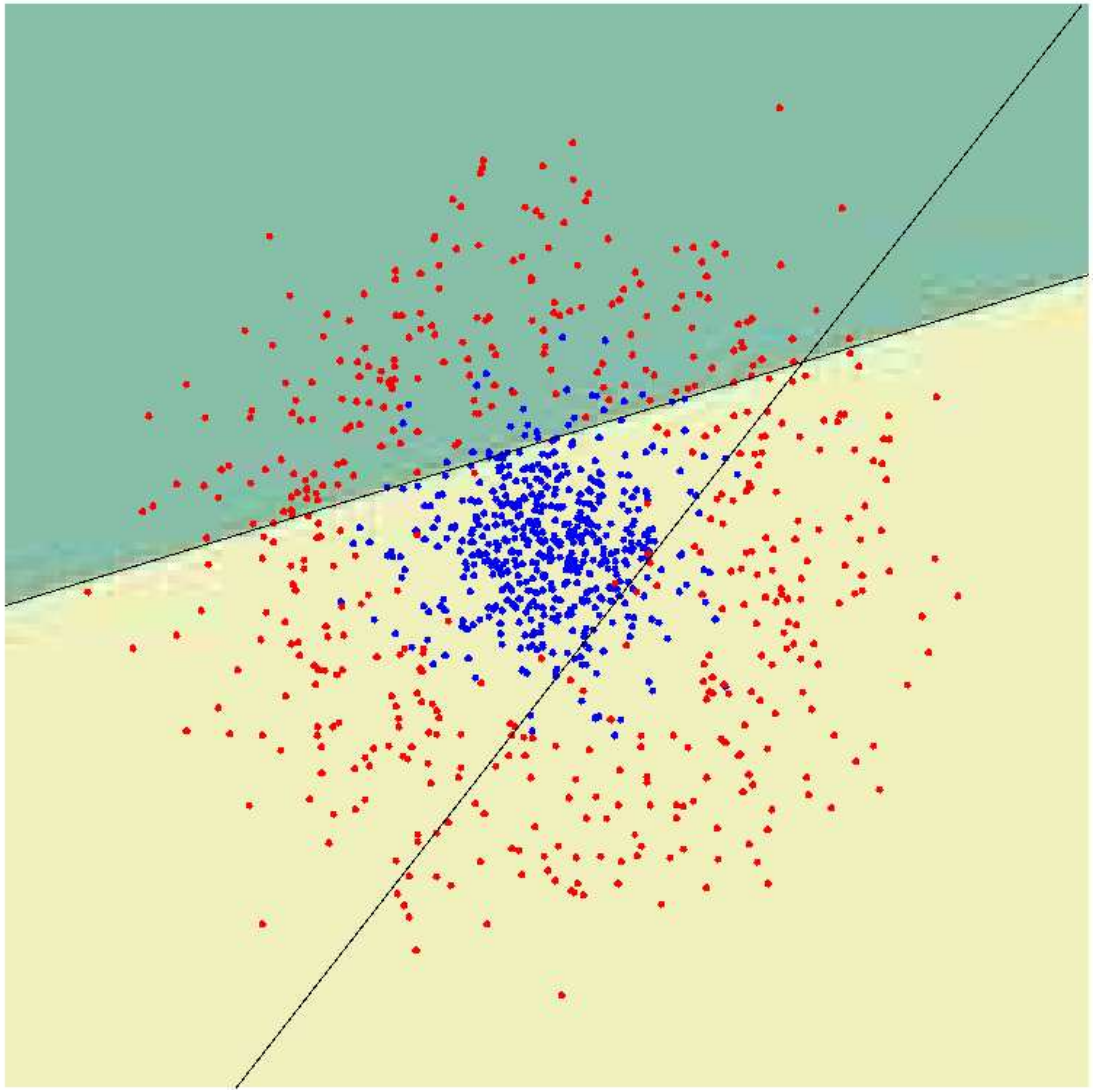
Adaboost



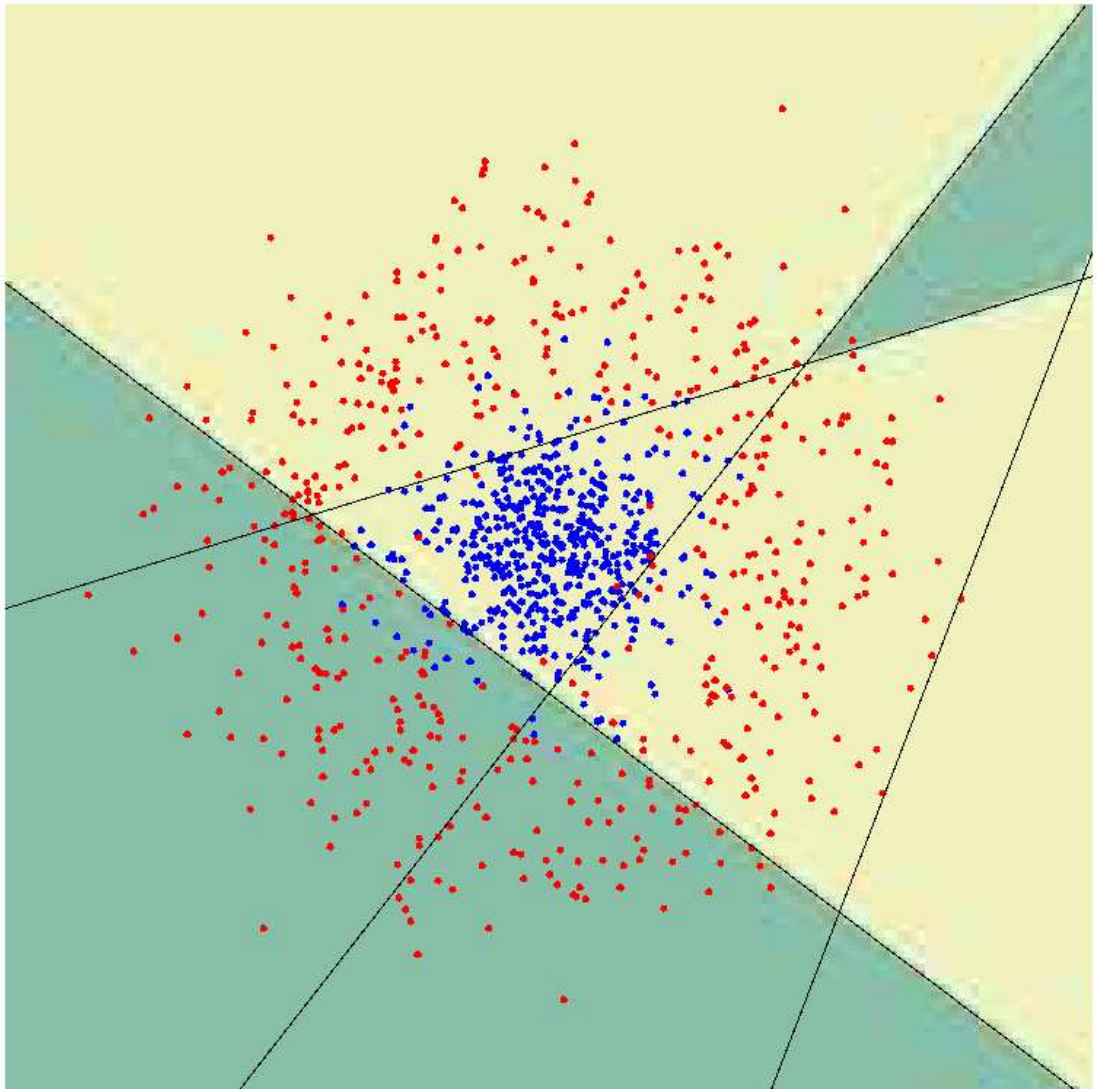
Adaboost



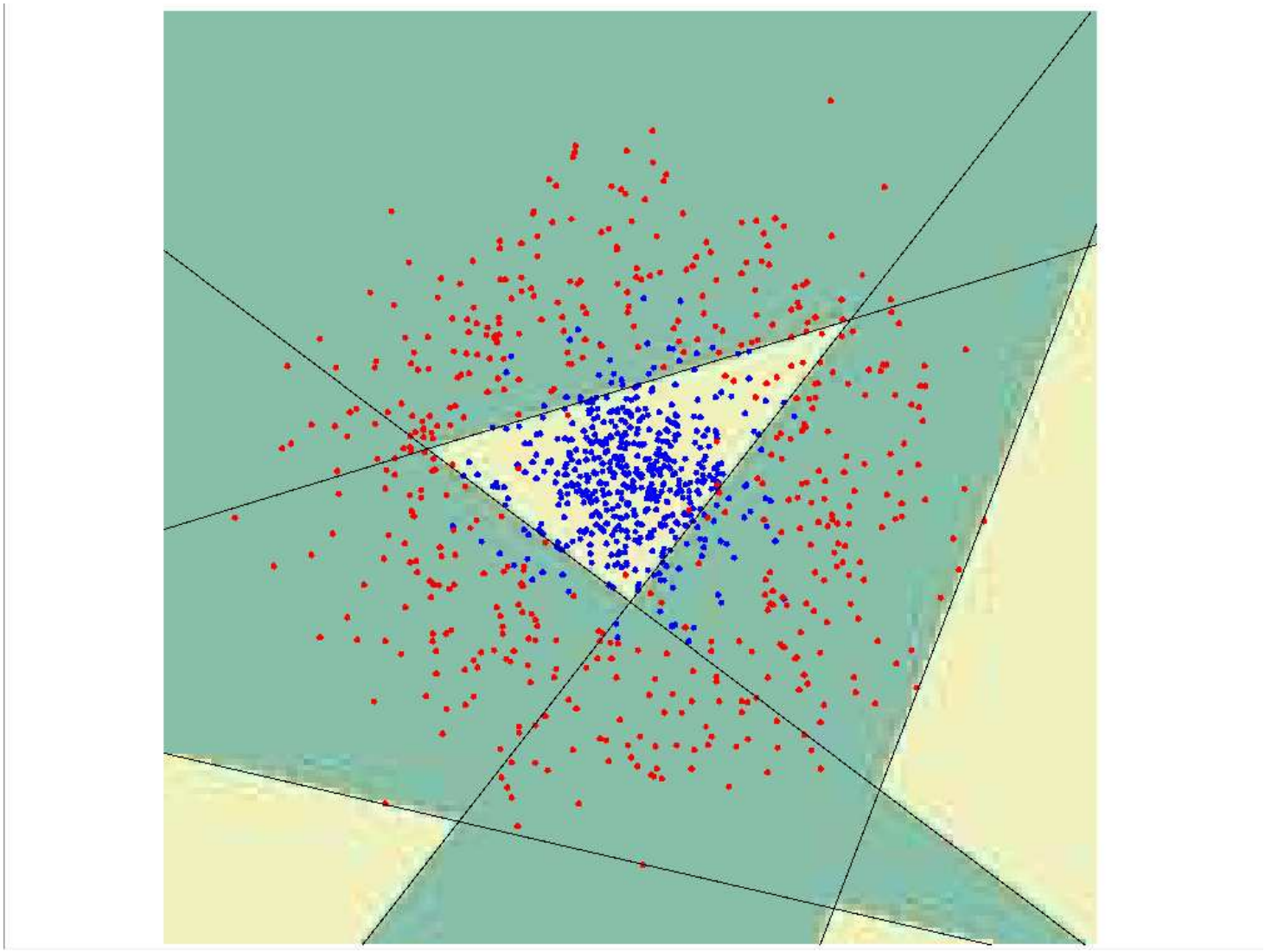
Adaboost



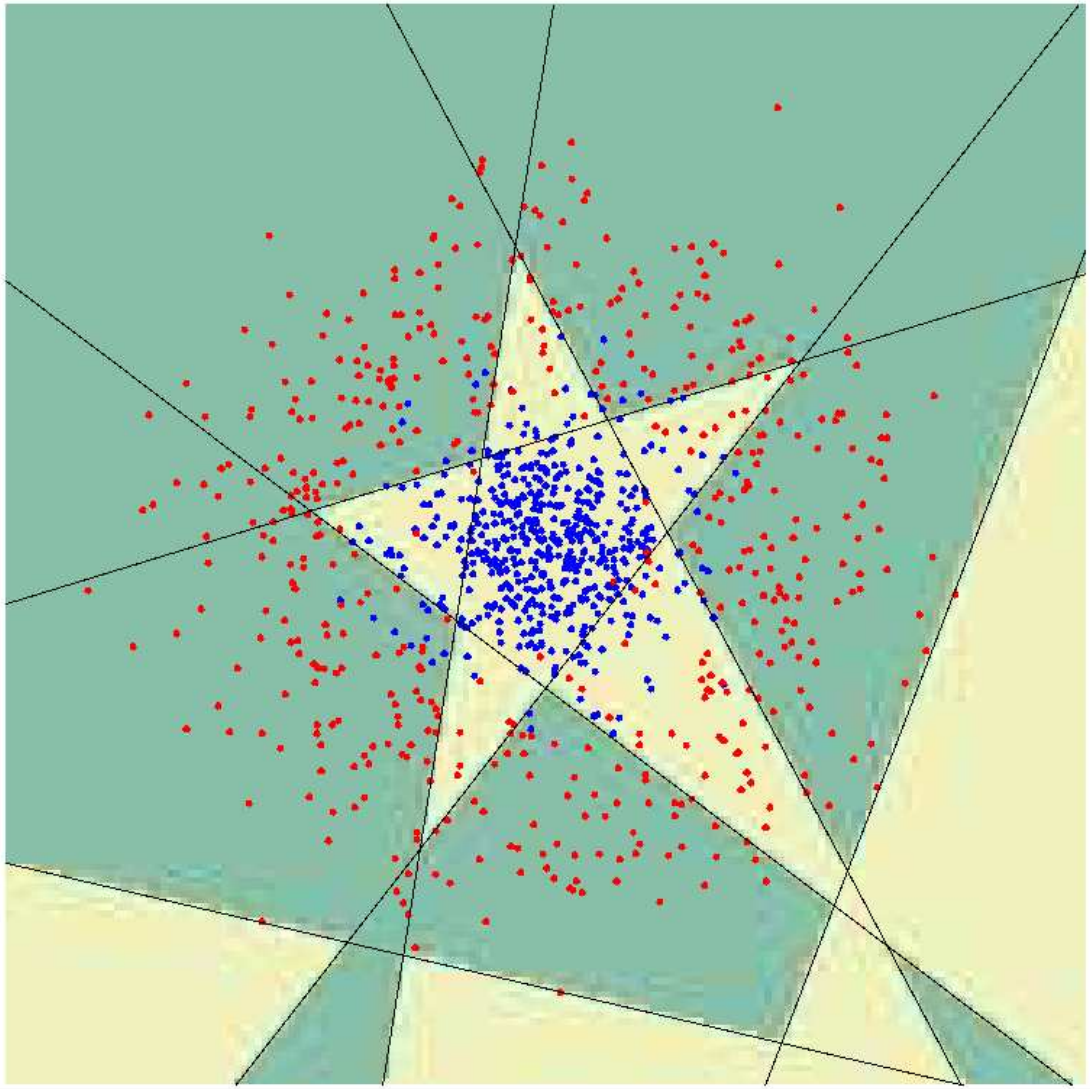
Adaboost



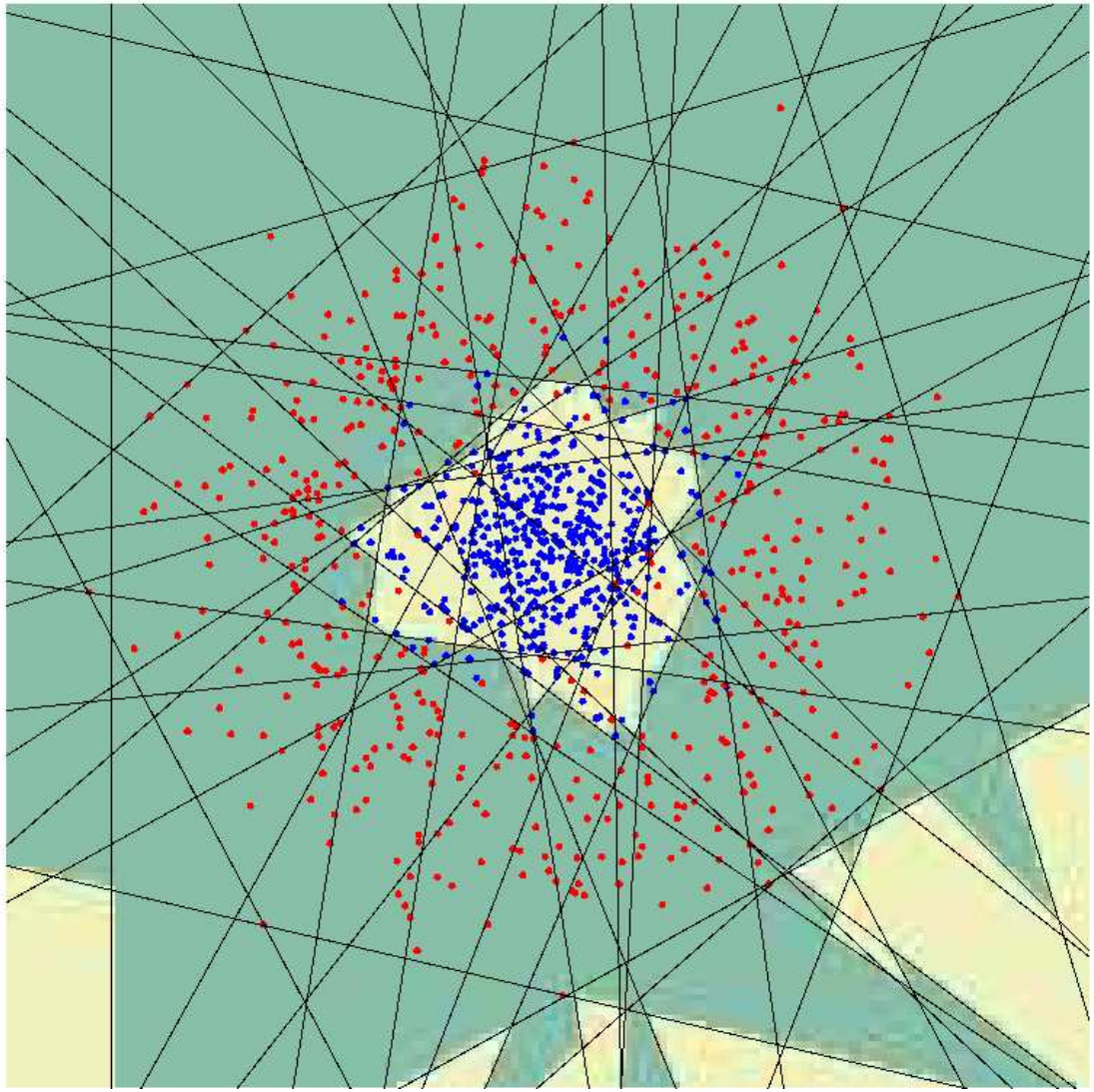
Adaboost



Adaboost



Adaboost



Adaboost

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Table 1: The boosting algorithm for learning a query online. T hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors.

Les filtres trouvés

Figure 5: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

Pour un taux de détection correcte de visages de 95%, on obtient un classifieur à 200 éléments donnant un seul faux positif pour une base de données de 14000 éléments.

Utilisation de filtres en cascade

- Utilisation en reconnaissance: le filtre est passé en chaque point de l'image. Très coûteux
- En pratique: de nombreux points peuvent être identifiés comme non visage à l'aide de classifieurs de plus petites tailles
  $\rightarrow$ commencer par un classifieur de petite taille pour élaguer les points possibles et se focaliser sur ces données avec des filtres plus complexes.

Utilisation de filtres en cascade

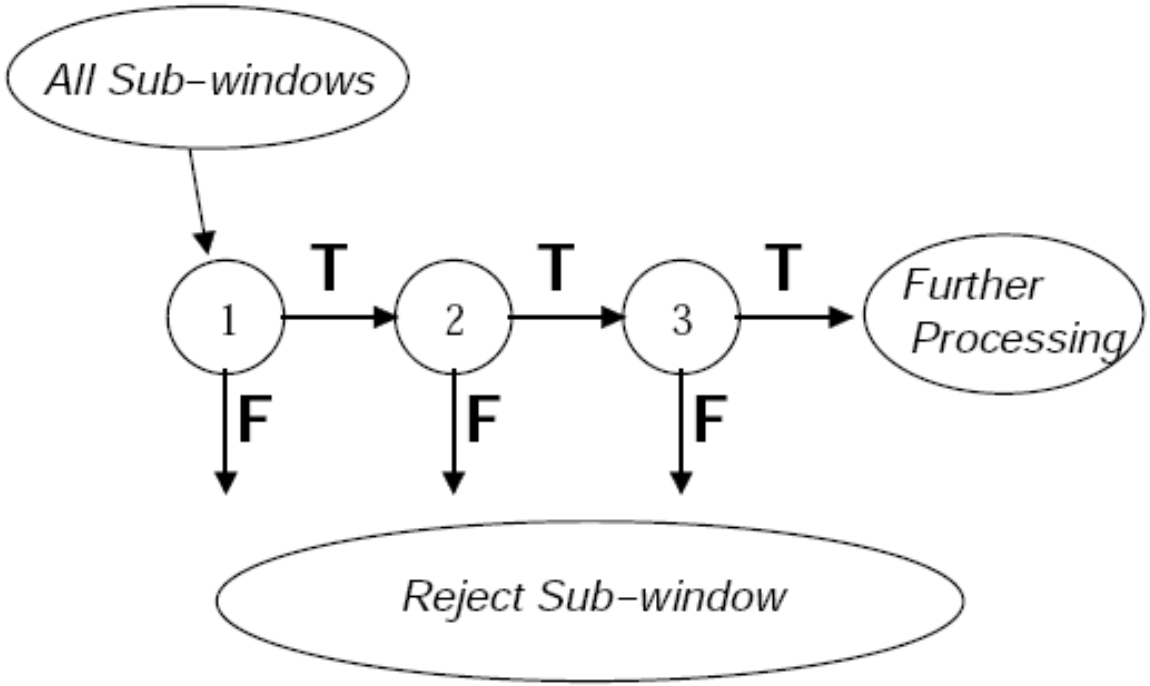


Figure 6: Schematic depiction of a the detection cascade. A series of classifiers are applied to every sub-window. The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the number of sub-windows have been reduced radically. Further processing can take any form such as additional stages of the cascade (as in our detection system) or an alternative detection system.

Résultats

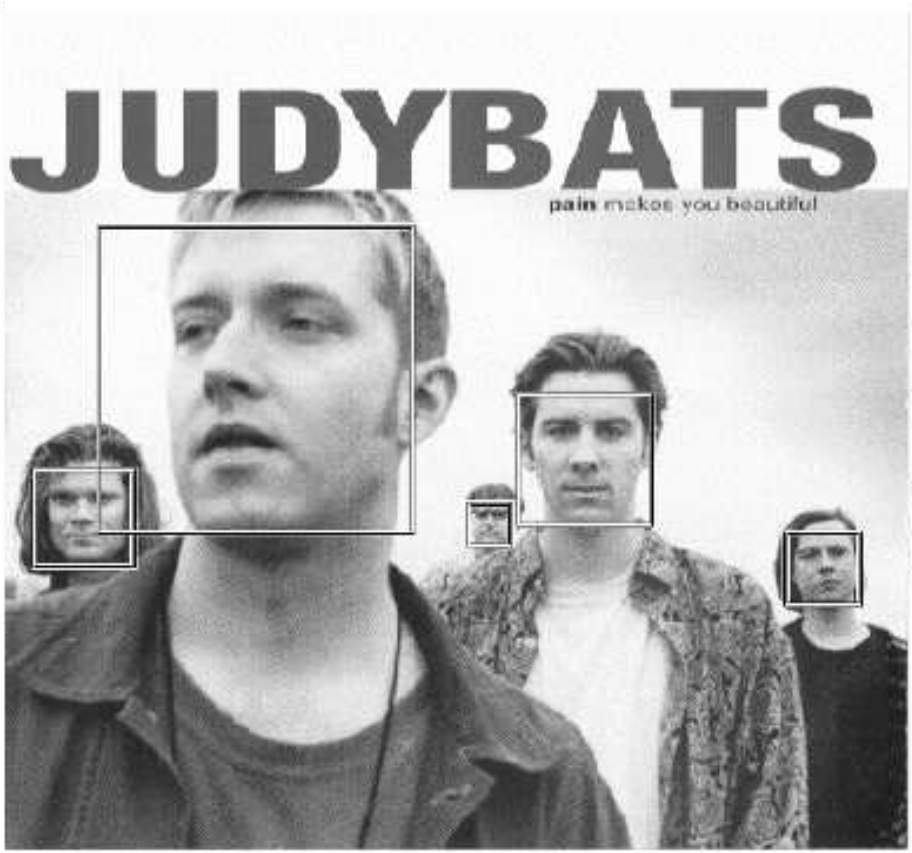
Données: 4916 faces de résolution 24x24 et 10000 non faces. Le détecteur final est formé de 32 cascades et utilise 4297 indices.

- premier classifieur: 2 éléments. Détection de 100% des faces. Rejet de 60% des non faces.
- classifier suivant: détection de 100% des faces, 80% des non faces rejetés
- puis trois classifieurs à 20 indices suivis de 5 classifieurs à 100 éléments et enfin 20 classifieurs à 200 éléments.

Note: détermination du nombre d'indices par couche: le nombre d'indices n'est augmenté que s'il conduit à une réduction significative du taux de faux positifs.
En moyenne, 8 indices sont testés par fenêtre (à comparer 4297 indices du classifieur): cela est dû au fait que de nombreux points sont rejetés par les toutes premières couches du filtre.

Résultats

prise en compte de changement d'échelle par scaling du filtre lui même (et pas de l'image)



Résultats

Figure 10: Output of our face detector on a number of test images from the MIT+CMU test set.

Bibliographie


S. Agarwal and D. Roth.
Learning a sparse representation for object detection.
In *Proceedings of 7th European Conference on Computer Vision, Copenhagen (Denmark)*, 2002.


J. P Benzecri.
L'analyse de données.
Dunod, Paris, 1971.


T.F. Cootes, C.J. Page, C.B. Jackson, and C.J. Taylor.
Statistical grey-level models for object location and identification.
Image and Vision Computing, 14:533–540, 1996.


T.F. Cootes and C.J. Taylor.
A mixture model for representing shape variation.
Image and Vision Computing, 17(8):567–574, 1999.


T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham.
Active shape models -their training and application.

Bibliographie


R. O. Duda and P. E. Hart.
Pattern Classification and Scene Analysis.
Wiley-InterScience, 1973.


E. Diday, J. Lemaire, J. Ponget, and F. Testu.
Eléments d'analyse de données.
Dunod, 1982.


A. K. Jain, R. P. W Duin, and J. Mao.
Statistical Pattern Recognition: A Review.
IEEE Transactions on PAMI, 22(1):4–37, January 2000.


L. Lebart and J. P. Fenelon.
Informatique et statistique appliquées.
Dunod, Paris, 1979.


Yann LeCun, Fu-Jie Huang, and Leon Bottou.
Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting.

Bibliographie

S. Maeda.
Un modèle articulatoire de la langue avec des composantes linéaires.
In *Actes 10èmes Journées d'Etude sur la Parole*, pages 152–162, Grenoble, May 1979.

L. Rabiner.
A tutorial on hidden markov models and selected applications in speech recognition.
Proc. IEEE, 77:257–286, 1989.

H. Y. Shum, K. Ikeuchi, and R. Reddy.
Principal component analysis with missing data and its application to polyhedral object modeling.
IEEE Transactions on PAMI, 17(9):854–867, 1995.

A. Webb, editor.
Statistical Pattern Recognition.
wiley, 2002.