

Master 2 Recherche

Apprentissage Statistique et Optimisation

François Yvon — Michèle Sebag
Alexandre Allauzen — Marc Schoenauer

<http://www.limsi.fr/Individu/allauzen/wiki/index.php/TSI09> — <http://tao.lri.fr/tiki-index.php>

21 octobre 2010: Représentation

Représentation pour l'apprentissage

- ▶ Sélection d'attributs
- ▶ Changements de représentation linéaires
- ▶ Changements de représentation non linéaires

Au début sont les données...

Patient	AGE x1	SEX x2	BMI x3	BP x4	... x5	Serum x6	Measurements x7	... x8	x9	x10	Response y
1	59	2	32.1	101	157	93.2	38	4	4.9	87	151
2	48	1	21.6	87	183	103.2	70	3	3.9	69	75
3	72	2	30.5	93	156	93.6	41	4	4.7	85	141
4	24	1	25.3	84	198	131.4	40	5	4.9	89	206
5	50	1	23.0	101	192	125.4	52	4	4.3	80	135
6	23	1	22.6	89	139	64.8	61	2	4.2	68	97
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
441	36	1	30.0	95	201	125.2	42	5	5.1	85	220
442	36	1	19.6	71	250	133.2	97	3	4.6	92	57

Motivations : Trouver et élaguer des descripteurs

Avant l'apprentissage : décrire les données.

- ▶ Une description trop pauvre \Rightarrow on ne peut rien faire
- ▶ Une description trop riche \Rightarrow on doit élaguer les descripteurs

Pourquoi ?

- ▶ L'apprentissage n'est pas un problème bien posé
- ▶ \implies Rajouter de l'information inutile (l'âge du vélo de ma grand-mère) peut dégrader les hypothèses obtenues.

Feature Selection, Position du problème

Contexte

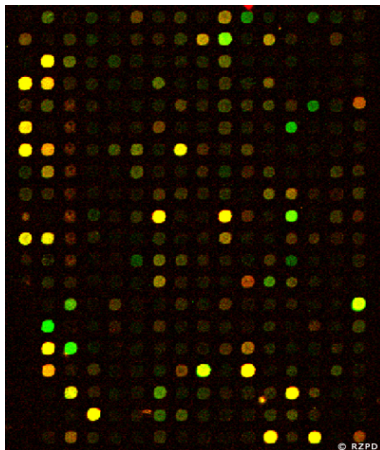
- ▶ Trop d'attributs % nombre exemples
 - ▶ En enlever Feature Selection
 - ▶ En construire d'autres Feature Construction
 - ▶ En construire moins Dimensionality Reduction
- ▶ Cas logique du 1er ordre : Propositionalisation

Le but caché : sélectionner ou construire des descripteurs ?

- ▶ Feature Construction : construire les bons descripteurs
- ▶ A partir desquels il sera facile d'apprendre
- ▶ Les meilleurs descripteurs = les bonnes hypothèses...

Quand l'apprentissage c'est la sélection d'attributs

Bio-informatique



- ▶ 30 000 gènes
- ▶ peu d'exemples (chers)
- ▶ but : trouver les gènes pertinents

Il est facile de faire n'importe quoi

Un exemple d'aventure fort désagréable...

<http://www-stat.stanford.edu/~hastie/TALKS/barossa.pdf>

(Rappel) Définition de p-value

Contexte : observation

le rouge est sorti 14 fois sur 20

Question : est-ce le hasard ?

deux hypothèses

▶ H_0 : le casino est honnête

$\Pr(\text{rouge}) = 1/2$

▶ ... ou non

p-value : Proba (observations | H_0)
probabilité d'observer ça sous l'hypothèse H_0

Nb de rouges sur N tirages $\sim \mathcal{B}(N, 1/2)$

$\Pr(\# \text{ rouges} \leq 14) = .057$

... On rejette l'hypothèse H_0 à 5% de niveau de confiance

Position du problème

Buts

- Sélection : trouver un sous-ensemble d'attributs
- Ordre/Ranking : ordonner les attributs

Formulation

Soient les attributs $\mathcal{A} = \{a_1, ..a_d\}$. Soit la fonction :

$$\mathcal{F} : \mathcal{P}(\mathcal{A}) \mapsto \mathbb{R}$$

$$A \subset \mathcal{A} \mapsto \text{Err}(A) = \text{erreur min. des hypothèses fondées sur } A$$

Trouver $\text{Argmin}(\mathcal{F})$

Difficultés

- Un problème d'optimisation combinatoire (2^d)
- D'une fonction \mathcal{F} inconnue...

Selection de features: approche filtre

Méthode univariée

Définir $score(a_i)$; ajouter itérativement les attributs maximisant $score$

ou retirer itérativement les attributs minimisant $score$

- + simple et pas cher
- optima très locaux

Backtrack possible

- ▶ Etat courant \mathcal{A}
- ▶ Ajouter a_i à \mathcal{A}
- ▶ Peut être ajouter a_i rend $a_j \in \mathcal{A}$ inutile ?
- ▶ Essayer d'enlever les features de \mathcal{A}

Backtrack = moins glouton; meilleures solutions ; beaucoup plus cher.

Selection de features: approche wrapping

Méthode multivariée

Mesurer la qualité d'un ensemble d'attributs :
estimer $\mathcal{F}(a_{i1}, \dots, a_{ik})$

Contre

Beaucoup plus cher : une estimation = un pb d'apprentissage.

Pour

Optima meilleurs

Selection de features: approche embarquée (embedded)

Principe – online

On rajoute à l'apprentissage un critère qui favorise les hypothèses à peu d'attributs.

Par exemple : trouver w , $h(x) = \langle w, x \rangle$, qui minimise

$$\sum_i (h(x_i) - y_i)^2 + \|w\|$$

Premier terme : coller aux données

Deuxième terme : favoriser w avec beaucoup de coordonnées nulles

Principe – offline

On a trouvé

$$h(x) = \langle w, x \rangle = \sum_{j=1}^d w_j x_j$$

Si $|w_j|$ petit, l'attribut j n'est pas important... Les enlever et recommencer.

Approches filtre, 1

Notations

Base d'apprentissage : $\mathcal{E} = \{(x_i, y_i), i = 1..n, y_i \in \{-1, 1\}\}$
 $a(x_i) =$ valeur attribut a pour exemple (x_i)

Corrélation

$$\text{corr}(a) = \frac{\sum_i a(x_i) \cdot y_i}{\sqrt{\sum_i (a(x_i))^2 \times \sum_i y_i^2}} \propto \sum_i a(x_i) \cdot y_i = \langle a, y \rangle$$

Limites

Attributs corrélés entre eux

Dépendance non linéaire

Corrélation et projection

Stoppiglia et al. 2003

Repeat

- ▶ a^* = attribut le plus corrélé à la classe

$$a^* = \operatorname{argmax} \left\{ \sum_i a(x_i) y_i, a \in \mathcal{A} \right\}$$

- ▶ Projeter les autres attributs sur l'espace orthogonal à a^*

$$\forall b \in \mathcal{A} \quad b \rightarrow b - \frac{\langle a^*, b \rangle}{\langle a^*, a^* \rangle} a^*$$

$$b(x_i) \rightarrow b(x_i) - \frac{\sum_j a^*(x_j) b(x_j)}{\sqrt{\sum_j a^*(x_j)^2} \sqrt{\sum_j b(x_j)^2}} a^*(x_i)$$

Corrélation et projection, suite

- ▶ Projeter y sur l'espace orthogonal à a^*

$$y \rightarrow y - \frac{\langle a^*, y \rangle}{\langle a^*, a^* \rangle} a^*$$
$$y_i \rightarrow y_i - \frac{\sum_j a^*(x_j) y_j}{\sum_j a^*(x_j)^2} a^*(x_i)$$

- ▶ Until Critère d'arrêt

- ▶ Rajouter des attributs aléatoires ($r(x_i) = \pm 1$) *probe*
- ▶ Quand le critère de corrélation sélectionne des attributs aléatoires, s'arrêter.

Limitations

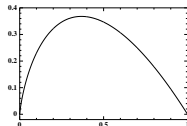
quand il y a plus de 6-7 attributs pertinents, ne marche pas bien.

Approches filtre, 3

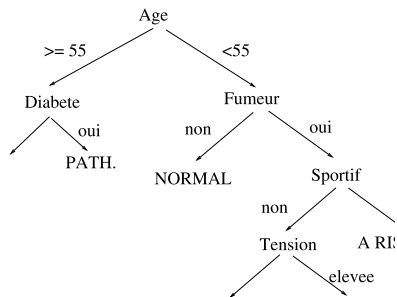
Gain d'information

arbres de décision

$$p([a = v]) = Pr(y = 1 | a(x_i) = v)$$
$$QI([a = v]) = -p([a = v]) \log p([a = v])$$
$$QI(a) = \sum_v Pr(a(x_i) = v) QI([a = v])$$



Gain d'information, suite



Limitations

Les mêmes que celles des arbres de décision

Problème de XOR.

Quelques scores

Notations : c_i une classe *en fouille de textes, contexte supervisé*
 a_k un mot (ou terme)

Critères

1. Fréquence conditionnelle

$$P(c_i|a_k)$$

2. Information mutuelle

$$P(c_i, a_k) \text{Log} \left(\frac{P(c_i, a_k)}{P(c_i)P(a_k)} \right)$$

3. Gain d'information

$$\sum_{c_i, \neg c_i} \sum_{a_k, \neg a_k} P(c, a) \text{Log} \frac{p(a, c)}{P(a)P(c)}$$

4. Chi-2

$$\frac{(P(t, c)P(\neg t, \neg c) - P(t, \neg c)P(\neg t, c))^2}{P(t)P(\neg t)P(c)P(\neg c)}$$

5. Pertinence

$$\text{Log} \frac{P(t, c) + d}{P(\neg t, \neg c) + d}$$

Approches wrapper

Principe générer/tester

Etant donné une liste de candidats $\mathcal{L} = \{A_1, \dots, A_p\}$

- Générer un candidat A
- Calculer $\mathcal{F}(A)$
 - apprendre h_A à partir de $\mathcal{E}|_A$
 - tester h_A sur un ensemble de test
- Mettre à jour \mathcal{L} .

$$= \hat{\mathcal{F}}(A)$$

Algorithmes

- hill-climbing / multiple restart
- algorithmes génétiques Vafaie-DeJong, IJCAI 95
- (*) programmation génétique & feature construction.

Krawiec, GPEH 01

Approches a posteriori

Principe

- Construire des hypothèses
- En déduire les attributs importants
- Eliminer les autres
- Recommencer

Algorithme : SVM Recursive Feature Elimination Guyon et al. 03

- SVM linéaire $\rightarrow h(x) = \text{sign}(\sum w_i \cdot a_i(x) + b)$
- Si $|w_i|$ est petit, a_i n'est pas important
- Eliminer les k attributs ayant un poids min.
- Recommencer.

Limites

Hypothèses linéaires

- Un poids par attribut.

Quantité des exemples

- Les poids des attributs sont liés.
- La dimension du système est liée au nombre d'exemples.

Or le pb de FS se pose souvent quand il n'y a pas assez d'exemples

Représentation pour l'apprentissage

- ▶ Sélection d'attributs
- ▶ **Changements de représentation linéaires**
- ▶ Changements de représentation non linéaires

Partie 2. Changements de représentation lineaires

- ▶ Réduction de dimensionalité
- ▶ Analyse en composantes principales
- ▶ Projections aléatoires
- ▶ Analyse sémantique latente

Dimensionality Reduction – Intuition

Degrees of freedom

- ▶ Image: 4096 pixels; but not independent
- ▶ Robotics: ($\#$ camera pixels + $\#$ infra-red) \times time; but not independent

Goal

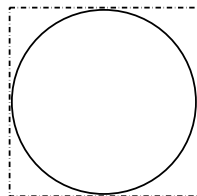
Find the (low-dimensional) structure of the data:

- ▶ Images
- ▶ Robotics
- ▶ Genes

Dimensionality Reduction

In high dimensions

- ▶ Everybody lives in the corners of the space
- ▶ Volume of Sphere $V_n = \frac{2\pi r^2}{n} V_{n-2}$
- ▶ All points are far from each other



Approaches

- ▶ Linear dimensionality reduction
 - ▶ Principal Component Analysis
 - ▶ Random Projection
- ▶ Non-linear dimensionality reduction

Criteria

- ▶ Complexity/Size
- ▶ Prior knowledge

e.g., relevant distance

Linear Dimensionality Reduction

Training set

unsupervised

$$\mathcal{E} = \{(\mathbf{x}_k), \mathbf{x}_k \in \mathbb{R}^D, k = 1 \dots N\}$$

Projection from \mathbb{R}^D onto \mathbb{R}^d

$$\mathbf{x} \in \mathbb{R}^D \rightarrow \begin{aligned} h(\mathbf{x}) &\in \mathbb{R}^d, \quad d \ll D \\ h(\mathbf{x}) &= A\mathbf{x} \end{aligned}$$

$$\text{s.t. minimize} \quad \sum_{k=1}^N \|\mathbf{x}_k - h(\mathbf{x}_k)\|^2$$

Principal Component Analysis

Covariance matrix S

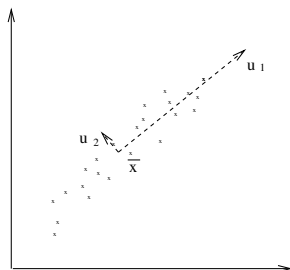
Mean

$$\mu_i = \frac{1}{N} \sum_{k=1}^N X_i(\mathbf{x}_k)$$

$$S_{ij} = \frac{1}{N} \sum_{k=1}^N (X_i(\mathbf{x}_k) - \mu_i)(X_j(\mathbf{x}_k) - \mu_j)$$

symmetric \Rightarrow can be diagonalized

$$S = U\Delta U' \quad \Delta = \text{Diag}(\lambda_1, \dots, \lambda_D)$$



Thm: Optimal projection in dimension d

projection on the first d eigenvectors of S

Let u_i the eigenvector associated to eigenvalue λ_i $\lambda_i > \lambda_{i+1}$

$$h : \mathbb{R}^D \mapsto \mathbb{R}^d, h(\mathbf{x}) = \langle \mathbf{x}, u_1 \rangle u_1 + \dots + \langle \mathbf{x}, u_d \rangle u_d$$

Sketch of the proof

1. Maximize the variance of $h(\mathbf{x}) = A\mathbf{x}$

$$\sum_k \|\mathbf{x}_k - h(\mathbf{x}_k)\|^2 = \sum_k \|\mathbf{x}_k\|^2 - \sum_k \|h(\mathbf{x}_k)\|^2$$

$$\text{Minimize } \sum_k \|\mathbf{x}_k - h(\mathbf{x}_k)\|^2 \Rightarrow \text{Maximize } \sum_k \|h(\mathbf{x}_k)\|^2$$

$$\text{Var}(h(\mathbf{x})) = \frac{1}{N} \left(\sum_k \|h(\mathbf{x}_k)\|^2 - \left\| \sum_k h(\mathbf{x}_k) \right\|^2 \right)$$

As

$$\left\| \sum_k h(\mathbf{x}_k) \right\|^2 = \left\| A \sum_k \mathbf{x}_k \right\|^2 = N^2 \|A\mu\|^2$$

where $\mu = (\mu_1, \dots, \mu_D)$.

Assuming that \mathbf{x}_k are centered ($\mu_i = 0$) gives the result.

Sketch of the proof, 2

2. Projection on eigenvectors u_i of S

Assume $h(\mathbf{x}) = \mathbf{Ax} = \sum_{i=1}^d \langle \mathbf{x}, v_i \rangle v_i$ and show $v_i = u_i$.

$$\text{Var}(AX) = (AX)(AX)' = A(XX')A' = ASA' = A(U\Delta U')A'$$

Consider $d = 1$, $v_1 = \sum w_i u_i$

$$\sum w_i^2 = 1$$

remind $\lambda_i > \lambda_{i+1}$

$$\text{Var}(AX) = \sum \lambda_i w_i^2$$

maximized for $w_1 = 1, w_2 = \dots = w_N = 0$

that is, $v_1 = u_1$.

Principal Component Analysis, Practicalities

Data preparation

- ▶ Mean centering the dataset

$$\begin{aligned}\mu_i &= \frac{1}{N} \sum_{k=1}^N X_i(\mathbf{x}_k) \\ \sigma_i &= \sqrt{\frac{1}{N} \sum_{k=1}^N X_i(\mathbf{x}_k)^2 - \mu_i^2} \\ z_k &= \left(\frac{1}{\sigma_i} (X_i(\mathbf{x}_k) - \mu_i) \right)_{i=1}^D\end{aligned}$$

Matrix operations

- ▶ Computing the covariance matrix

$$S_{ij} = \frac{1}{N} \sum_{k=1}^N X_i(z_k) X_j(z_k)$$

- ▶ Diagonalizing $S = U' \Delta U$
might be not affordable...

Complexity $\mathcal{O}(D^3)$

Random projection

Random matrix

$$A : \mathbb{R}^D \mapsto \mathbb{R}^d \quad A[d, D] \quad A_{i,j} \sim \mathcal{N}(0, 1)$$

define

$$h(\mathbf{x}) = \frac{1}{\sqrt{d}} A \mathbf{x}$$

Property: h preserves the norm in expectation

$$E[\|h(\mathbf{x})\|^2] = \|\mathbf{x}\|^2$$

With high probability

$$1 - 2\exp\{-(\varepsilon^2 - \varepsilon^3)\frac{d}{4}\}$$

$$(1 - \varepsilon)\|\mathbf{x}\|^2 \leq \|h(\mathbf{x})\|^2 \leq (1 + \varepsilon)\|\mathbf{x}\|^2$$

Random projection

Proof

$$h(\mathbf{x}) = \frac{1}{\sqrt{d}} A \mathbf{x}$$

$$\begin{aligned} E(\|h(\mathbf{x})\|^2) &= \frac{1}{d} E \left[\sum_{i=1}^d \left(\sum_{j=1}^D A_{i,j} X_j(\mathbf{x}) \right)^2 \right] \\ &= \frac{1}{d} \sum_{i=1}^d E \left[\left(\sum_{j=1}^D A_{i,j} X_j(\mathbf{x}) \right)^2 \right] \\ &= \frac{1}{d} \sum_{i=1}^d \sum_{j=1}^D E[A_{i,j}^2] E[X_j(\mathbf{x})^2] \\ &= \frac{1}{d} \sum_{i=1}^d \sum_{j=1}^D \frac{\|\mathbf{x}\|^2}{D} \\ &= \|\mathbf{x}\|^2 \end{aligned}$$

Random projection, 2

Johnson Lindenstrauss Lemma

For $d > \frac{9 \ln N}{\varepsilon^2 - \varepsilon^3}$, with high probability

$$(1 - \varepsilon) \|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|h(\mathbf{x}_i) - h(\mathbf{x}_j)\|^2 \leq (1 + \varepsilon) \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

More:

<http://www.cs.yale.edu/clique/resources/RandomProjectionMethod.pdf>

Analyse Sémantique Latente - LSA

1. Motivation
2. Algorithme
3. Discussion

Example

- c1: Human machine interface for ABC computer applications
- c2: A survey of user opinion of computer system response time
- c3: The EPS user interface management system
- c4: System and human system engineering testing of EPS
- c5: Relation of user perceived response time to error measurement

- m1: The generation of random, binary, ordered trees
- m2: The intersection graph of paths in trees
- m3: Graph minors IV: Widths of trees and well-quasi-ordering
- m4: Graph minors: A survey

Exemple, suite

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

LSA, 2

Motivations

- ▶ Contexte : représentation sac de mots
- ▶ Malédiction de la dimensionalité
- ▶ Synonymie / Polysémie

\mathbb{R}^D

Objectifs

- ▶ Réduire la dimension
- ▶ Avoir une “bonne topologie”

\mathbb{R}^d

une bonne distance

Remarque

- ▶ une similarité évidente : le cosinus
- ▶ pourquoi ce n'est pas bon ?

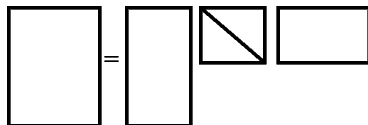
Plus d'info

<http://lsa.colorado.edu>

LSA, 3

Input

Matrice $X = \text{mots} \times \text{documents}$



Principe

1. Changement de base des mots, documents aux concepts
2. Réduction de dimension

Différence Analyse en composantes principales

LSA \equiv Singular Value Decomposition

Input

X matrice mots \times documents

$m \times d$

$$X = U' S V$$

avec

- U : changement de base mots $m \times r$
- V : changement de base des documents $r \times d$
- S : matrice diagonale $r \times r$

Réduction de dimension

- S Ordonner par valeur propre décroissante
- $S' = S$ avec annulation de toutes les vp, sauf les (300) premières.

$$X' = U' S' V$$

Intuition

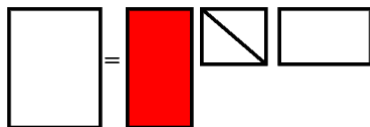
$$X = \begin{pmatrix} & m_1 & m_2 & m_3 & m_4 \\ d_1 & 0 & 1 & 1 & 1 \\ d_2 & 1 & 1 & 1 & 0 \end{pmatrix}$$

m_1 et m_4 ne sont pas “physiquement” ensemble dans les mêmes documents ; mais ils sont avec les mêmes mots ; “donc” ils sont un peu “voisins”...

Après SVD + Réduction,

$$X = \begin{pmatrix} & m_1 & m_2 & m_3 & m_4 \\ d_1 & \epsilon & 1 & 1 & 1 \\ d_2 & 1 & 1 & 1 & \epsilon \end{pmatrix}$$

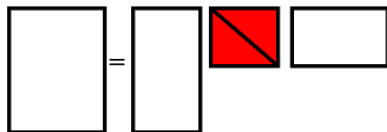
Algorithme



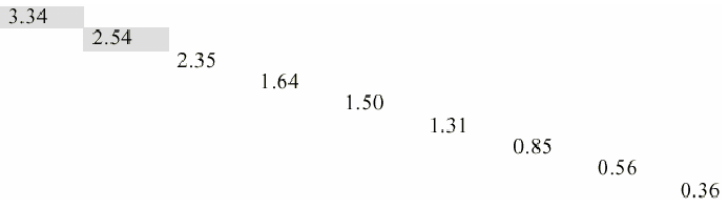
Singular value
Decomposition of the
words by contexts matrix

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

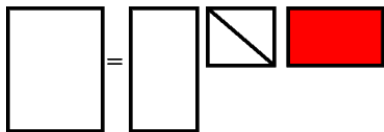
Algorithme, 2



Singular value
Decomposition of the
words by contexts matrix



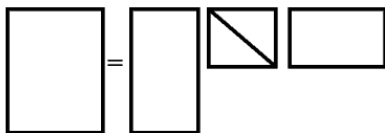
Algorithme. 3



Singular value
Decomposition of the
words by contexts matrix

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Algorithme, 4

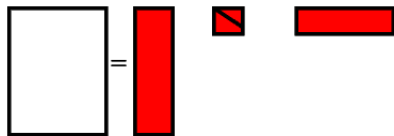


Singular value
Decomposition of the
words by contexts matrix

3.34

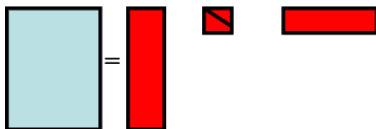
2.54

Algorithme, 5



Singular value
Decomposition of the
words by contexts matrix

Algorithm, 6



Singular value
Decomposition of the
words by contexts matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

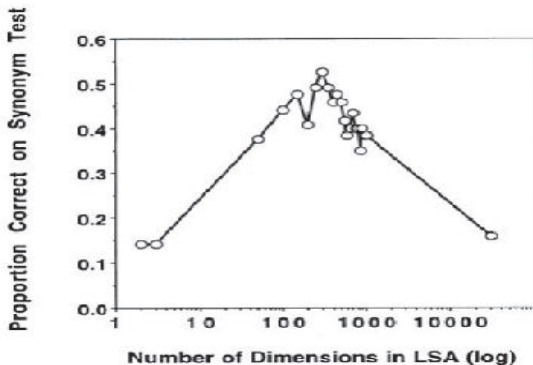
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Discussion

Une application

Test de synonymie

TOEFL



Déterminer le nb de dimensions/vp

Expérimentalement...

Quelques remarques

et la négation ?

battu par: nb de hits sur le Web

aucune importance (!)

P. Turney

Quelques applications

- ▶ Educational Text Selection
Permet de sélectionner automatiquement des textes permettant d'accroître les connaissances de l'utilisateur.
- ▶ Essay Scoring
Permet de noter la qualité d'une rédaction d'étudiant
- ▶ Summary Scoring & Revision
Apprendre à l'utilisateur à faire un résumé
- ▶ Cross Language Retrieval
permet de soumettre un texte dans une langue et d'obtenir un texte équivalent dans une autre langue

LSA – Analyse en composantes principales

Ressemblances

- ▶ Prendre une matrice
- ▶ La mettre sous forme diagonale
- ▶ Annuler toutes les valeurs propres sauf les plus grandes
- ▶ Projeter sur l'espace obtenu

Différences

	ACP	LSA
Matrice	covariance attributs	mots \times documents
d	2-3	100-300

Probabilistic LSA

T. Hoffman, UAI 99

Principe : supposons

- des “groupes” de documents \equiv variables cachées z
- “peu” de var. cachées (comparé aux paires mots \times documents)

$$P(\text{documents}|\text{mots}) = P(\text{documents}|z) \times P(z|\text{mots})$$

Alors : Contraindre la décomposition

$$X_p = U_p S_p V_p^t$$

- $U_p : p(\text{documents}|z)$
- $S_p : p(z)$
- $V_p = p(\text{mots}|z)$

Comment ? Expectation Maximization

Expectation Maximization

Principe de l'apprentissage génératif

- Input : éléments g_1, \dots, g_N
- Output : modèles $\mathcal{G}_1, \dots, \mathcal{G}_n$

Algorithmes itératifs

A chaque itération

Pour tout g_i

Trouver \mathcal{G}_j tq

EXPECTATION

$$p(g_i | \mathcal{G}_j) = \max\{p(g_i | \mathcal{G}_k), k = 1..n\}$$

Pour tout \mathcal{G}_j

Soit $E_j = \{g_i \text{ affecté à } \mathcal{G}_j\}$

Mettre à jour \mathcal{G}_j pour maximiser

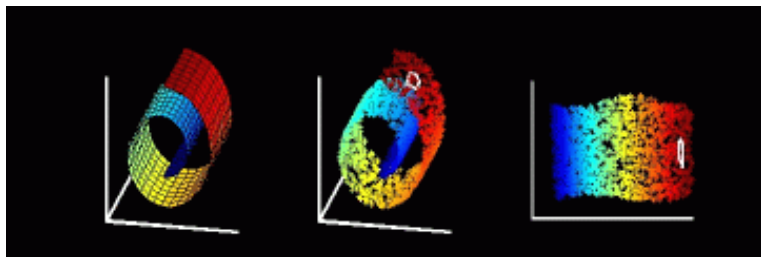
MAXIMISATION

$$\sum_{g \in E_j} p(g | \mathcal{G}_j)$$

Représentation pour l'apprentissage

- ▶ Sélection d'attributs
- ▶ Changements de représentation linéaires
- ▶ **Changements de représentation non linéaires**

Non-Linear Dimensionality Reduction



Conjecture

Examples live in a manifold of dimension $d \ll D$

Goal: consistent projection of the dataset onto \mathbb{R}^d

Consistency:

- ▶ Preserve the structure of the data
- ▶ e.g. preserve the distances between points

Multi-Dimensional Scaling

Position of the problem

- ▶ Given $\{\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{x}_i \in \mathbb{R}^D\}$
- ▶ Given $sim(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}^+$
- ▶ Find projection Φ onto \mathbb{R}^d

$$\begin{aligned}x \in \mathbb{R}^D &\rightarrow \Phi(x) \in \mathbb{R}^d \\sim(\mathbf{x}_i, \mathbf{x}_j) &\sim sim(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))\end{aligned}$$

Optimisation

Define X , $X_{i,j} = sim(\mathbf{x}_i, \mathbf{x}_j)$; X^Φ , $X_{i,j}^\Phi = sim(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$

Find Φ minimizing $\|X - X^\Phi\|$

Rq : Linear Φ = Principal Component Analysis

But linear MDS does not work: preserves all distances, while

only *local* distances are meaningful

Non-linear projections

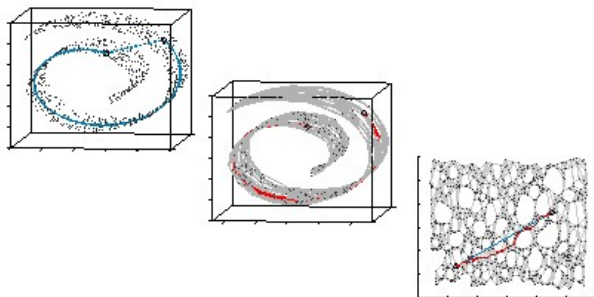
Approaches

- ▶ Reconstruct global structures from local ones and find global projection
- ▶ Only consider local structures

Isomap

LLE

Intuition: locally, points live in \mathbb{R}^d



Isomap

Tenenbaum, da Silva, Langford 2000
<http://isomap.stanford.edu>

Estimate $d(x_i, x_j)$

- ▶ Known if \mathbf{x}_i and \mathbf{x}_j are close
- ▶ Otherwise, compute the shortest path between \mathbf{x}_i and \mathbf{x}_j
geodesic distance (dynamic programming)

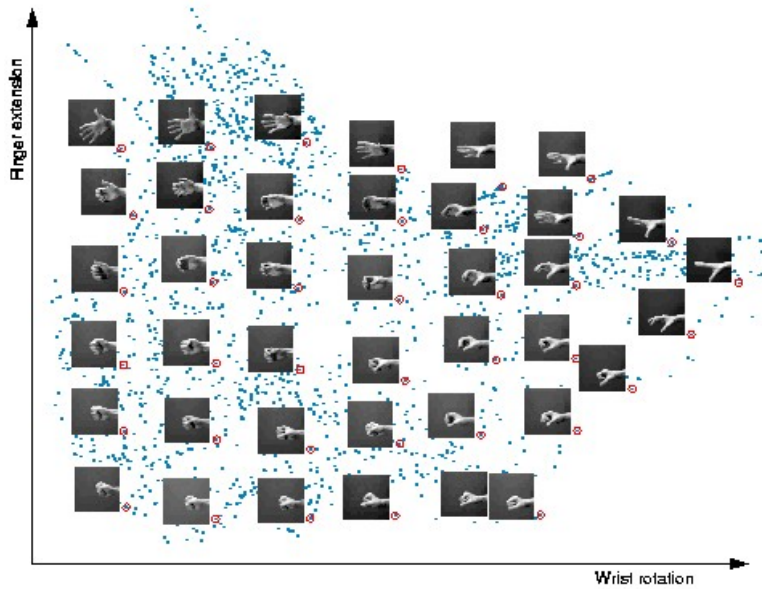
Requisite

If data points sampled in a convex subset of \mathbb{R}^d ,
then geodesic distance \sim Euclidean distance on \mathbb{R}^d .

General case

- ▶ Given $d(\mathbf{x}_i, \mathbf{x}_j)$, estimate $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$
- ▶ Project points in \mathbb{R}^d

Isomap, 2



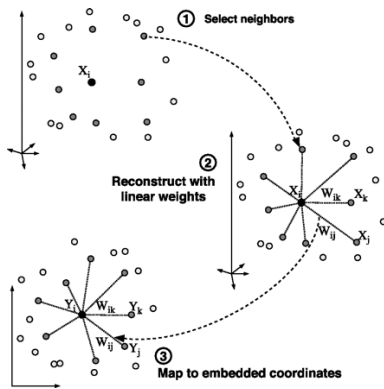
Locally Linear Embedding

Roweis and Saul, 2000

<http://www.cs.toronto.edu/~roweis/lle/>

Principle

- Find local description for each point: depending on its neighbors



Local Linear Embedding, 2

Find neighbors

For each \mathbf{x}_i , find its nearest neighbors $\mathcal{N}(i)$

Parameter: number of neighbors

Change of representation

Goal Characterize \mathbf{x}_i wrt its neighbors:

$$\mathbf{x}_i = \sum_{j \in \mathcal{N}(i)} w_{i,j} \mathbf{x}_j \quad \text{with} \quad \sum_{j \in \mathcal{N}(i)} w_{ij} = 1$$

Property: invariance by translation, rotation, homothety

How Compute the local covariance matrix:

$$C_{j,k} = \langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{x}_k - \mathbf{x}_i \rangle$$

Find vector w_i s.t. $Cw_i = 1$

Local Linear Embedding, 3

Algorithm

Local description: Matrix W such that

$$\sum_j w_{i,j} = 1$$

$$W = \underset{W}{\operatorname{argmin}} \left\{ \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_j w_{i,j} \mathbf{x}_j \right\|^2 \right\}$$

Projection: Find $\{z_1, \dots, z_n\}$ in \mathbb{R}^d minimizing

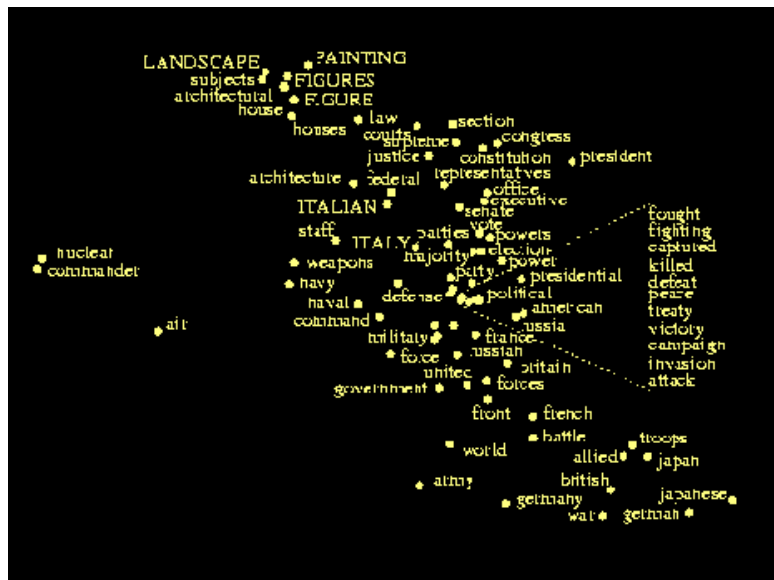
$$\sum_{i=1}^N \left\| z_i - \sum_j w_{i,j} z_j \right\|^2$$

Minimize $((I - W)Z)'((I - W)Z) = Z'(I - W)'(I - W)Z$

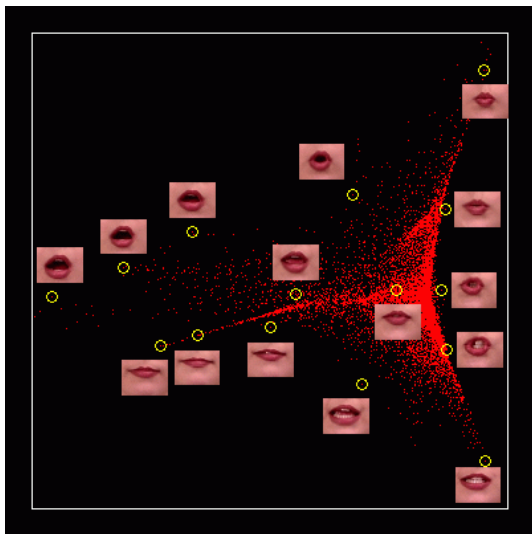
Solutions: vectors z_i are eigenvectors of $(I - W)'(I - W)$

- ▶ Keeping the d eigenvectors with lowest eigenvalues > 0

Example, Texts



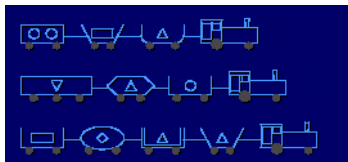
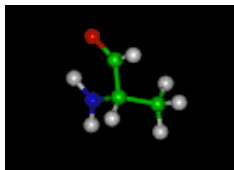
Example, Images



LLE

Propositionalization

Relational domains



Relational learning

PROS

Use domain knowledge

CONS

Covering test \equiv subgraph matching

Inductive Logic Programming

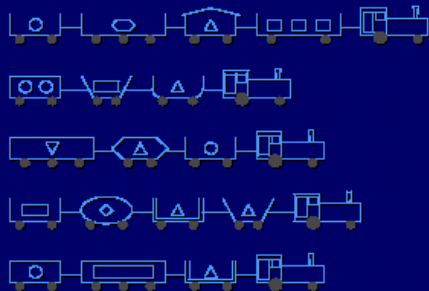
Data Mining
exponential complexity

Getting back to propositional representation: **propositionalization**

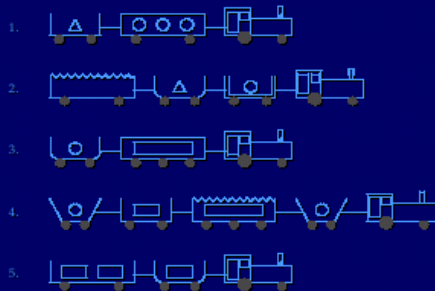
West - East trains

Michalski 1983

1. TRAINS GOING EAST



2. TRAINS GOING WEST



Propositionalization

Linus (ancestor)

Lavrac et al, 94

$West(a) \leftarrow Engine(a, b), first_wagon(a, c), roof(c), load(c, square, 3)...$
 $West(a') \leftarrow Engine(a', b'), first_wagon(a', c'), load(c', circle, 1)...$

West	Engine(X)	First Wagon(X,Y)	Roof(Y)	Load ₁ (Y)	Load ₂ (Y)
a	b	c	yes	square	3
a'	b'	c'	no	circle	1

Each column: a role predicate, where the predicate is determinate linked to former predicates (left columns) with a single instantiation in every example

Propositionalization

Stochastic propositionalization

Kramer, 98

Construct random formulas \equiv boolean features

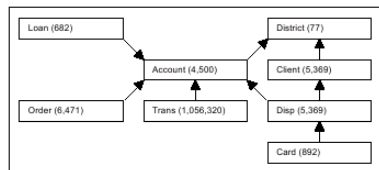
SINUS – RDS

<http://www.cs.bris.ac.uk/home/rawles/sinus>

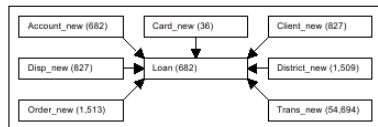
<http://labe.felk.cvut.cz/~zelezny/rsd>

- ▶ Use modes (user-declared) `modeb(2,hasCar(+train,-car))`
- ▶ Thresholds on number of variables, depth of predicates...
- ▶ Pre-processing (feature selection)

Propositionalization



DB Schema



Propositionalization

RELAGGS

Database aggregates

- ▶ average, min, max, of numerical attributes
- ▶ number of values of categorical attributes

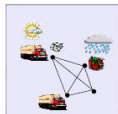
Apprentissage par Renforcement Relationnel

Real Time Strategy Games



- Many objects of various types in complex interactions
- Good players can generalize across situations involving distinct object configurations

The Logistics Domain



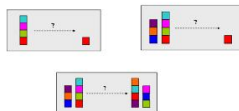
- Move many objects around with many other objects
- Identities and numbers of objects always changing

Robot Soccer



- Reasoning about relationship between objects (players and ball) key to good play

and of course Blockworld



- Would like a policy that is independent of number of objects/blocks

Propositionalisation

Contexte variable

- ▶ Nombre de robots, position des robots
- ▶ Nombre de camions, lieu des secours

Besoin: Abstraire et Generaliser

Attributs

- ▶ Nombre d'amis/d'ennemis
- ▶ Distance du plus proche robot ami
- ▶ Distance du plus proche ennemi