Advanced Machine Learning: from Theory to Practice
Part II - Lecture 1
Unsupervised and Semi-supervised learning

F. d'Alché-Buc and E. Le Pennec

Fall 2015 - Winter 2016

Outline : period 2

1. Today (Dec 4) : Semi-supervised learning (F.dAB)
2. Dec 11 : Collaborative Filtering (Erwan Le Pennec)
3. Dec 18 : Neural Networks (Erwan Le Pennec)
4. Jan 8 : Ranking (Stephan Clemencon)
5. Jan 15 : Deep learning (Nicolas Leroux)
6. Jan 22 : Deep learning (Nicolas Leroux)

Unlabeled data

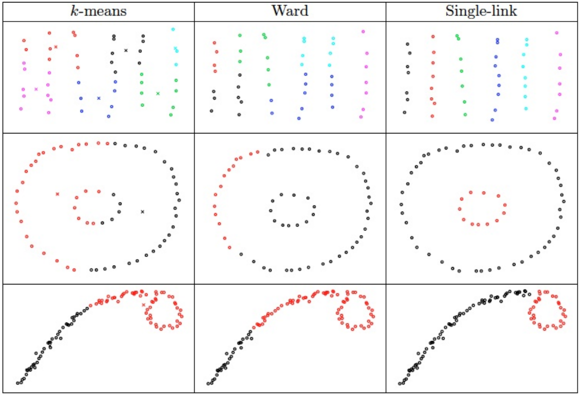
- Available data are unlabeled : documents, webpages, clients database ...
- Labeling data is expensive and requires some expertise

Learning from unlabeled data

- Modeling probability distribution $\rightarrow$ graphical models
- Dimension reduction $\rightarrow$ pre-processing for pattern recognition
- **Clustering** : group data into homogeneous clusters $\rightarrow$ organize your data, make easier access to them, pre and post processing, application in segmentation, document retrieval, bioinformatics ...

Operator-valued kernels for multiregression

Different clusterings



Semi-supervised learning

- Benefit from the availability of huge sets of unlabeled data
- Unlabeled data inform us about the probability distribution of the data $p(x)$
- Can we use it ? does it improve the performance of the resulting regressors/classifiers ?

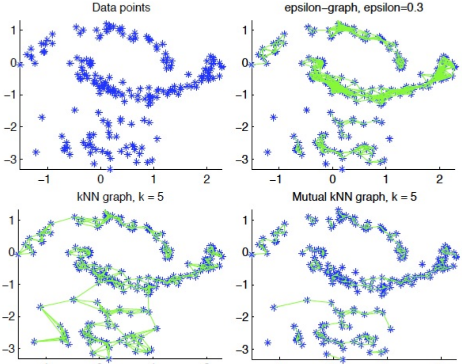
Goal

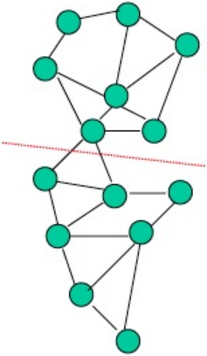
- Labeled data : $\mathcal{S}_\ell = \{(x_1, y_1), \ldots, (x_\ell, y\ell)\}$
- Unlabeled data : $\mathcal{X}_u = \{x_{\ell+1}, \ldots, x_{\ell+u}\}$, $n = \ell + u$: available during training !
- Usually $\ell << u$
- Test data : $\mathcal{X}_{test} = \{x_{n+1}, \ldots, x_{n+m}\}$: not available during training
- **Learn a function f : $\mathcal{X} \rightarrow \mathcal{Y}$ (regression/classification) that behaves well on test data**

- Data $x_1, \ldots, x_n$ with their similarity values $s_{ij} \geq 0$ or with heir distance d_{ij} values
- Build a graph $G = (V, E)$
- vertex v_i corresponds to data x_i
- An edge w_{ij} is defined according the ε-graph method or the k-nn method

Several ways to construct it :

- ε-graph : connect all points whose pairwise distance is at most ε (alt. whose pairwise similarity is at least ε
- k-nearest-neighbour-graph : connect v_i and v_j if x_i is among the k-nearest-neighbours of x_j OR x_i is among the k-nearest-neighbours of x_j





- $Cut(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} w_{ij}$

Definitions

- W matrix : adjacency matrix
- Degree matrix D : $d_{ii} = \sum_j w_{ij}$, if $i \neq j$, $d_{ij} = 0$
- Graph Laplacian : L = D - W

Eigenvalues/eigenvectors

- Eigenvector u : $Lu = \lambda u$
- We notice that (D-W) 1_n = D - W.1 = 0, then the smallest eigenvalue is $\lambda_1 = 0$

Connected components

- The multiplicity of the smallest eigenvalue (0) of L is the number of connected components in the graph

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}$$

Clustering with Laplacian graph

- $f_i, i = 1, \ldots, n$: membership of data i to cluster 1
- $f_i = 1$ if $x_i \in Cluster1(A)$, -1 otherwise Cluster 2 ($\bar{A}$)
- Find f that minimizes $J(f)$:

$$
\begin{aligned}
J(f) &= \frac{1}{4}\sum_{i,j} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{4}\sum_{i,j} w_{ij}(f_i^2 + f_j^2 - 2f_i f_j) \\
&= \frac{1}{2} f^T (D - W) f \\
&= 2|V| RatioCut(A, \bar{A})
\end{aligned}
$$

RatioCut

$$Ratiocut(A, \bar{A}) = \frac{cut(A, \bar{A})}{|A|} + \frac{cut(\bar{A}, A)}{|A|}$$

- Avoid trivial solution : $f \perp 1_n$
- Control the complexity of f (ℓ_2 regularization) : $\sum_i f_i^2 = n$

$\min_{f \in \mathbb{R}^n} f^T L f$
subject to : $f \perp 1$, $||f|| = \sqrt{n}$

- Solve the previous relaxed problem $\rightarrow$ the vector corresponding to the second smallest eigenvalue is solution
- Threshold the values of f to get discrete values1 and -1

Algorithm

- Solve the previous relaxed problem $\rightarrow$ take the k eigenvectors $(v_1, \ldots, v_k)$ corresponding to the k smallest positive eigenvalues
- Represent your data in the new space spanned by these k vectors : form the matrix V with the v_k's as column vectors
- each row of V represents an individual
- Apply k-means in the k-dimensional space

- Notations : A and B are two disjoint subsets of the nodes set V that form a partition
- $cut(A, B) = \sum_{t \in A, u \in B} w_{t,u}$
- $vol(A) = \sum_{t \in A, u \in V} w_{t,u}$
- Normalized cut (avoid isolated subset) : $Ncut(A, B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(B,A)}{vol(B)}$

$\min_{f \in \mathbb{R}^n} \frac{f^T L f}{f^T D f}$
subject to : $f^T D 1 = 0$

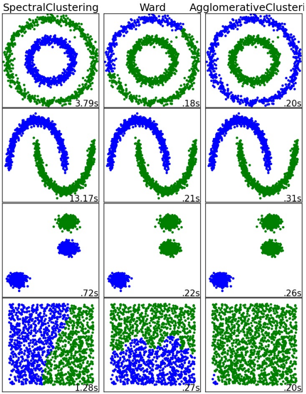
- Notations : A and B are two disjoint subsets of the nodes set V that form a partition
- $cut(A, B) = \sum_{t \in A, u \in B} w_{t,u}$
- $vol(A) = \sum_{t \in A, u \in V} w_{t,u}$
- Normalized cut (avoid isolated subset) : $Ncut(A, B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(B,A)}{vol(B)}$

$\min_{f \in \mathbb{R}^n} \frac{f^T L f}{f^T D f}$
subject to : $f^T D 1 = 0$

Solve the generalized eigenvalue problem :
$(D - W)f = \lambda D f$ which can be re-written as
$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} z = \lambda z$
with $z = D^{-\frac{1}{2}} f$.

- Importance of the initial graph : several ways to construct it (k-neighbours)
- Able to extract clusters on a manifold
- Stability
- Model selection : eigengap

Spectral clustering
Difficult clustering tasks



- Figure from scikitkearn :
- code : spectral $=$ cluster.SpectralClustering(n$_c$*lusters* $=$ 2, *eigen$_s$olver* $='$ *arpack'*, *affinity* $=$ " *nearest$_n$eighbors*")

Spectral clustering
Eigengap heuristic

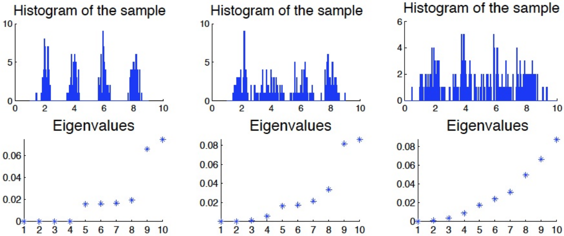


Figure 4: Three data sets, and the smallest 10 eigenvalues of L_{rw}.

- Source Tutorial U. Von Luxburg

- Learn f from $\mathcal{X}$ to $\mathcal{Y}$ using $\mathcal{S}_\ell = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ and $\mathcal{X}_u = \{x_{\ell+1}, \ldots, x_{\ell+u}\}$
- Methods
  - Self-training (including generative approaches)
  - Loss-based methods
    - Margin for unlabeled data
    - Smoothness penalty (graph-based semi-supervised learning)

- Any classifier : f

Principle

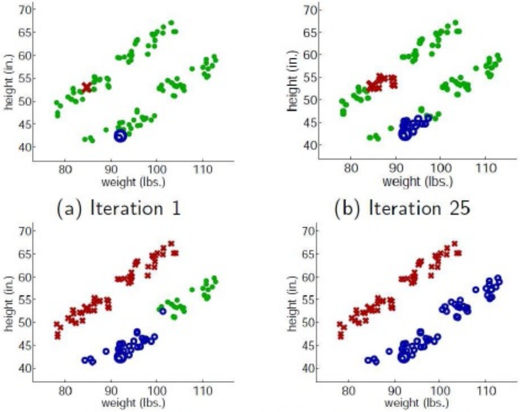
1. k=0
2. Learn f_k by training on $\mathcal{S}_k = \mathcal{S}$
3. Use f to label $\mathcal{X}_u$ and get $\mathcal{S}_{k+1}$ new set of $\ell + u$ labeled data
4. Learn f_{k+1} by training on $\mathcal{S}_{k+1}$
5. If $D(f_{k+1}, f_k)$ is small then STOP else GOTO 3

Semi-supervised learning

Self-training : example with k-NN (1)

- Two nice clusters without outliers [example Piyush Ray]

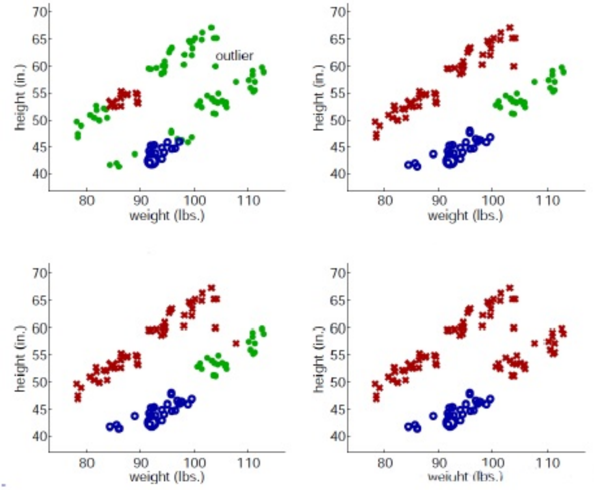
Base learner: KNN classifier



(a) Iteration 1 (b) Iteration 25

(c) Iteration 74 (d) Final labeling of all instances

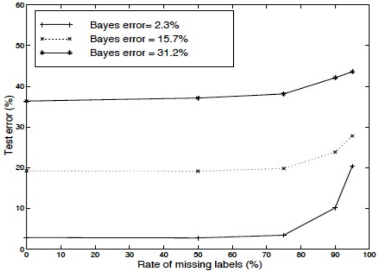
- Two clusters with outliers



- Margin : $\rho(x, y, h) = y.h(x)$
- Which margin for unlabeled data ?
- Reinforce the confidence of the classifier
  - $\rho_2(x, h) = h(x)^2$
  - $\rho_1(x, h) = |h(x)|$
  - **Implicit assumption** : cluster assumption : data in the same cluster share the same label
- Worked for SVM, MarginBoost, ...

- $h_t \in \mathcal{H}$: base classifier
- Boosting model : $H_T(x) = \sum_t \alpha_t h_t(x)$
- Loss function : $J(H_t) = \sum_{i=1}^{\ell} exp(-\rho(x_i, y_i, H_t)) + \lambda \sum_{j=\ell+1}^{n} exp(-\rho_u(x_j, H_t))$

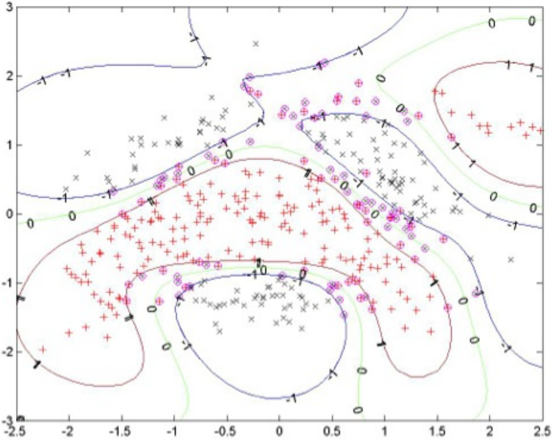
- Toys problems with different level of difficulty (we control Bayes error by mixing more or less the generative models)



[figure : NIPS 2001]

Semi-supervised learning

Data used in the previous sample



In transduction, one wants to predict the outputs of the test set $y_{\ell+1}, \dots, y_{\ell+u}$. Let us call $\mathbf{y}^* = [y_1^*, \dots, y_u^*]$ the prediction vector. Joachims proposed a Transductive SVM with a soft margin :

TSVM

$$\underset{\mathbf{w}, y^*, b}{\text{minimize}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell} \xi_i + C^* \sum_{j=1}^{u} \xi_i^*$$

under the constraints

$$y_i(\mathbf{w}^T \mathbf{x}_i + \mathrm{b}) \geq 1 - \xi_i,\ i = 1, \dots, n$$

$$y_j^*(\mathbf{w}^T \mathbf{x}_{\ell+j} + \mathrm{b}) \geq 1 - \xi_j^*,\ i = 1, \dots, n$$

$$y_j^* \in \{-1, +1\},\ j = 1, \dots, u$$

$$\xi_i \geq 0$$

$$\xi_j^* \geq 0$$

Ref : Joachims, 1999.

- Bennet and Demiriz 1999, 2001
- Bennet and Demiriz proposed $\rho_1(x, h) = |h(x)|$ and an implementation of S3VM based on Mangasarian's work.
- Robust Linear Programming

SVM formulation :

$$\min_{w,b,\eta} C \sum_{i=1}^{t} \eta_i + \frac{1}{2}\|w\|^2$$

$$s.t.\ y_i[wx_i - b] + \eta_i \geqslant 1$$

$$\eta_i \geqslant 0, i = 1, \ldots, l$$

Semi-supervised learning
Semi-supervised Support Vector Machine (S3VM) 2/2

S3VM formulation (Bennet and Demiriz) :

$$
\begin{array}{ll}
\min\limits_{\mathbf{w},b,\eta,\xi,z} & C\left[\sum\limits_{i=1}^{\ell}\eta_i + \sum\limits_{j=\ell+1}^{\ell+k} min(\xi_j, z_j)\right] + \|\,\mathbf{w}\,\| \\
subject\ to & y_i(\mathbf{w}\cdot x_i + b) + \eta_i \geq 1 \quad \eta_i \geq 0 \quad i = 1,\ldots,\ell \\
 & \mathbf{w}\cdot x_j - b + \xi_j \geq 1 \quad \xi_j \geq 0 \quad j = \ell+1,\ldots,\ell+k \\
 & -(\mathbf{w}\cdot x_j - b) + z_j \geq 1 \quad z_j \geq 0
\end{array}
$$

With integer variables $d_i = 0 or 1$ according it belongs to class 1 or class -1 (d has to be learned as well) :

$$
\begin{array}{ll}
\min\limits_{\mathbf{w},b,\eta,\xi,z,d} & C\left[\sum\limits_{i=1}^{\ell}\eta_i + \sum\limits_{j=\ell+1}^{\ell+k} (\xi_j + z_j)\right] + \|\,\mathbf{w}\,\| \\
subject\ to & y_i(\mathbf{w}\cdot x_i - b) + \eta_i \geq 1 \quad \eta_i \geq 0 \quad i = 1,\ldots,\ell \\
 & \mathbf{w}\cdot x_j - b + \xi_j + M(1-d_j) \geq 1 \quad \xi_j \geq 0 \quad j = \ell+1,\ldots,\ell+k \\
 & -(\mathbf{w}\cdot x_j - b) + z_j + Md_j \geq 1 \quad z_j \geq 0 \quad d_j = \{0,1\}
\end{array}
$$

Mixed integer programming.

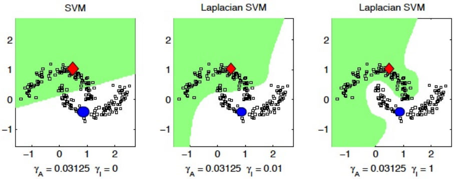
Let k be a positive definite kernel and $\mathcal{H}_k$ the unique RKHS induced by k.

Smoothness constraint / Manifold regularization 1/2

- Training data : $\mathcal{S}_\ell = \{(x_i, y_i, i =, \ldots \ell\}$ and $\mathcal{S}_u = \{x_{\ell+1}, \ldots, x_{\ell+u}\}$
- For $f \in \mathcal{H}_k$ and W a similarity matrix between data
- Impose an additional penalty that ensures smoothness of function f : for two close inputs, f takes close values
- Ref : Belkin, Nyogi and Sindwani (2006)

The key ideas :

- We assume that a better knowledge of the marginal distribution $P_x(x)$ will give us bette knowledge of $P(Y|x)$.
- If two points x_1 and x_2 are close in the intrinsic geometry of P_x then the conditional distribution $P(y|x_1)$ and $P(y|x_2)$ will be close.



- If $\mathcal{M}$, the support of P_x is a submanifold $\subset \mathbb{R}^p$, then we can try to minimize the penalty :

$$||f||_I^2 = \int_{\mathcal{M}} ||\nabla_{\mathcal{M}} f||^2 p(x) dx$$

- $\nabla_{\mathcal{M}} f$ is the gradient of f along the manifold $\mathcal{M}$
- Approximation of $||f||_I^2$:

$$||f||_I^2 \approx \sum_{ij} w_{ij} (f(x_i) - f(x_j))^2,$$

where W is the adjacency matrix of the data graph.

Let k be a positive definite kernel and $\mathcal{H}_k$ the unique RKHS induced by k.

Smoothness constraint / Manifold regularization

Minimize $J(f)$ in $\mathcal{H}_k$:

$$J(f) = \frac{1}{\ell}\sum_{i=1}^{\ell} V(x_i, y_i, f) + \lambda ||f||_k^2 + \lambda_u \sum_{ij} w_{ij}(f(x_i) - f(x_j))^2$$

Let k be a positive definite kernel and $\mathcal{H}_k$ the unique RKHS induced by k.

Smoothness constraint / Manifold regularization

Minimize $J(f)$ in $\mathcal{H}_k$:

$$\begin{aligned} J(f) &= \frac{1}{\ell}\sum_{i=1}^{\ell} V(x_i, y_i, f) + \lambda ||f||_k^2 + \lambda_u \sum_{ij} w_{ij}(f(x_i) - f(x_j))^2 \\ &= \frac{1}{\ell}\sum_{i=1}^{\ell} V(x_i, y_i, f) + \lambda ||f||_k^2 + \lambda_u f^T L f \end{aligned}$$

$$
\begin{aligned}
J(f) &= \frac{1}{\ell}\sum_{i=1}^{\ell} V(x_i, y_i, f) + \lambda ||f||_k^2 + \lambda_u \sum_{ij=1}^{\ell+u} w_{ij}(f(x_i) - f(x_j))^2 \\
&= \frac{1}{\ell}\sum_{i=1}^{\ell} V(x_i, y_i, f) + \lambda ||f||_k^2 + \lambda_u f^T L f
\end{aligned}
$$

Any minimizer of $J(f)$ admits a representation
$\hat{f}(\cdot) = \sum_{i=1}^{\ell+u} \alpha_i k(x_i, \cdot)$

- Closed-from solution : extension of ridge regression

$$
\begin{aligned}
V(x_i, y_i, f) &= (y_i - f(x_i))^2 \\
\lambda_L = \frac{\lambda_u}{u+\ell} & \\
\hat{\alpha} &= (JK + \lambda\ell Id + \frac{\lambda_u \ell}{(u+\ell)^2} LK)^{-1} Y
\end{aligned}
$$

K : Gram matrix for all data
J : $(\ell+u)\times(\ell+u)$ diagonal matrix with the first ℓ values equal to 1 and the remaining ones to 0.

We choose the hinge loss functions :

$$\min_{f \in \mathcal{H}_k} \frac{1}{\ell} \sum_{i=1}^{\ell} (1 - y_i f(x_i))_+ + \lambda ||f||_k^2 + \frac{\lambda_u}{u+\ell} f^T L f$$

We benefit from the representer theorem.

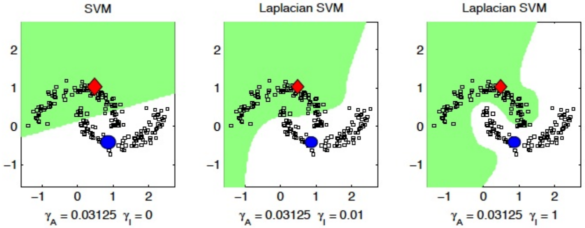
In practise, we solve :

$$\min_{\alpha\in\mathbb{R}^{l+u},\xi\in\mathbb{R}^{l}} \frac{1}{l}\sum_{i=1}^{l}\xi_i+\gamma_A\alpha^T K\alpha+\frac{\gamma_I}{(u+l)^2}\alpha^T KLK\alpha$$

$$\text{subject to: } y_i\Big(\sum_{j=1}^{l+u}\alpha_j K(x_i,x_j)+b\Big)\geq 1-\xi_i, \quad i=1,\ldots,l$$

$$\xi_i\geq 0 \quad i=1,\ldots,l.$$

Results : Belkin et al. 2006, JMLR.



- Code the Laplacian SVM or the Laplacian Kernel Ridge regressor
- Book : Semi-supervised learning, Chapelle, Scholpkoft, Zien,MIT