MDI341 - Advanced Machine Learning

From Unsupervised Learning to Semi-Supervised Learning

Florence d'Alché-Buc

Contact: florence.dalche@telecom-paris.fr,
Télécom Paris, Institut Polytechnique de Paris

Outline

Learning from unlabeled data

Unlabeled data

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

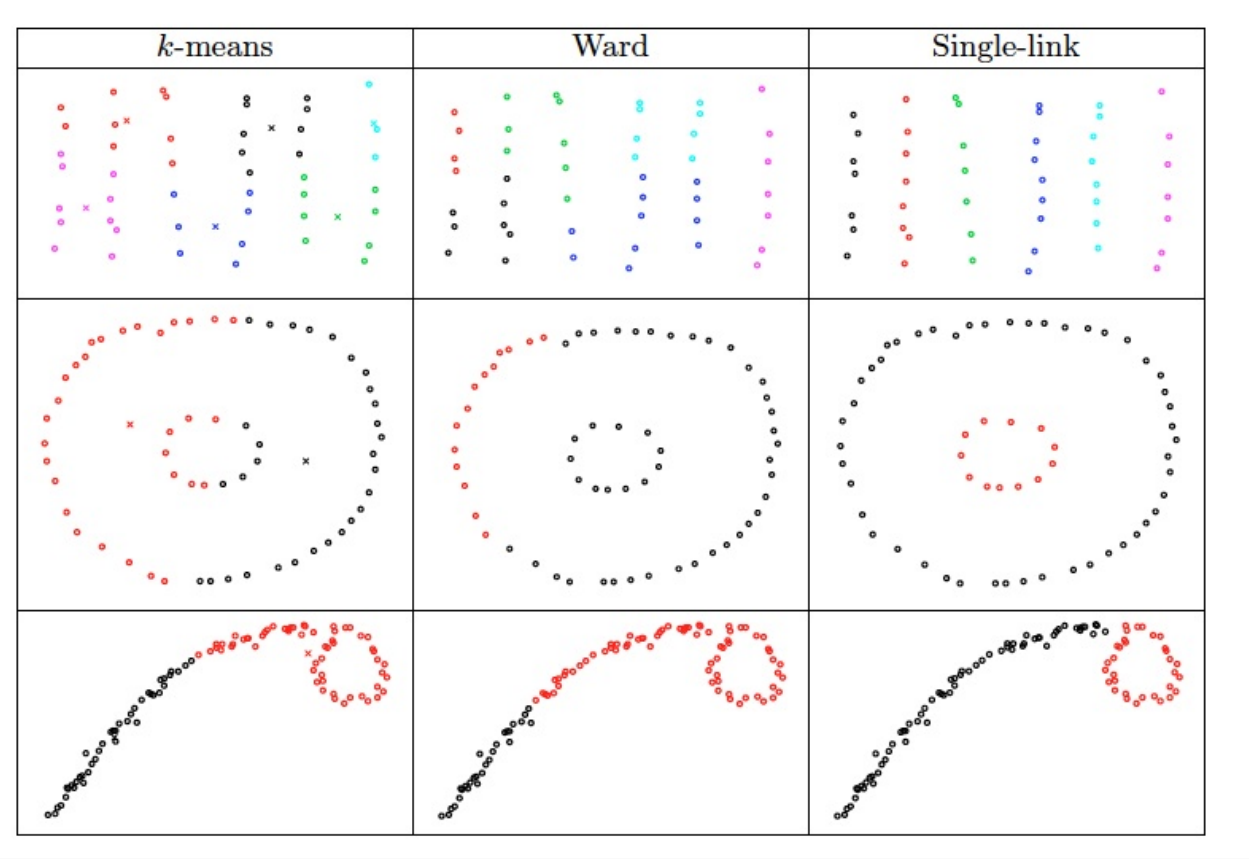
Learning from unlabeled data

- Modeling probability distributions with probabilistic graphical models $\rightarrow$ generative approaches
- Representation Learning, Dimensionality 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 ...

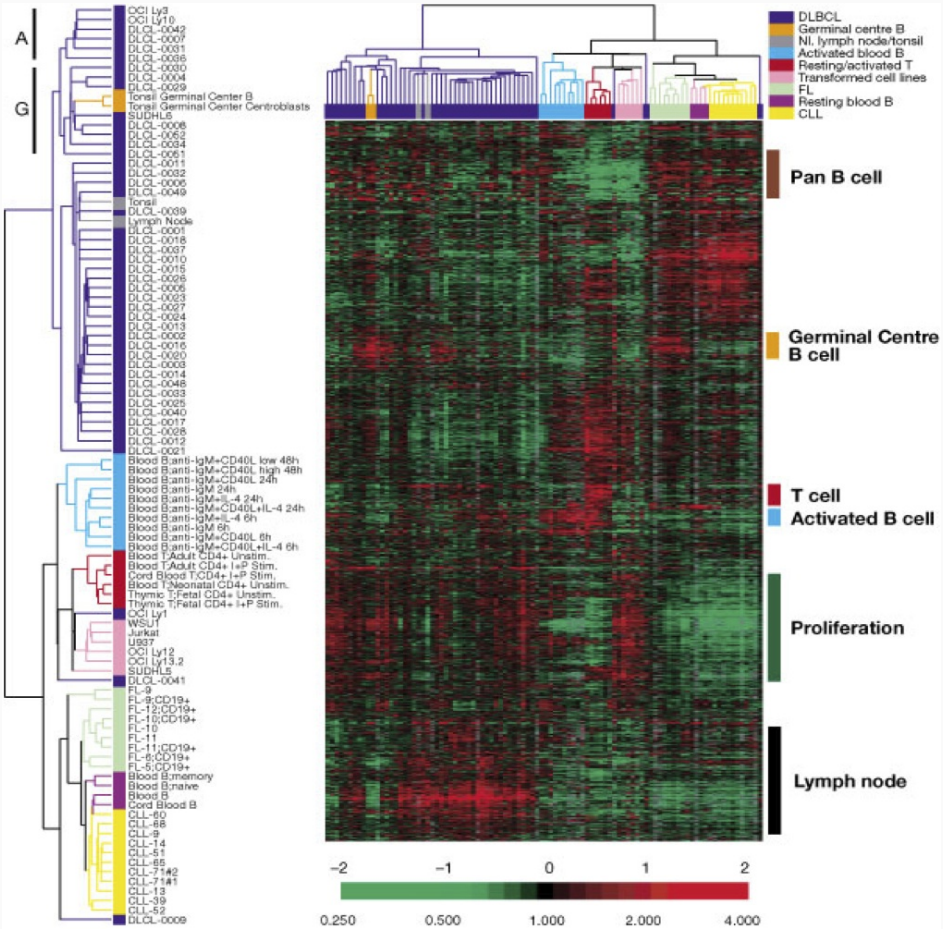
Motivation for Clustering

- **Marketing:** finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records
- **Insurance:** identifying groups of insurance policy holders
- **Computer vision:** segmentation
- **WWW:**document classification; clustering weblog data to discover groups of similar access patterns.

Different clusterings



Clustering genes according their expression



Nature Feb, 2000, Paper by Allzadeh. A et al.

Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling

Clustering for image segmentation

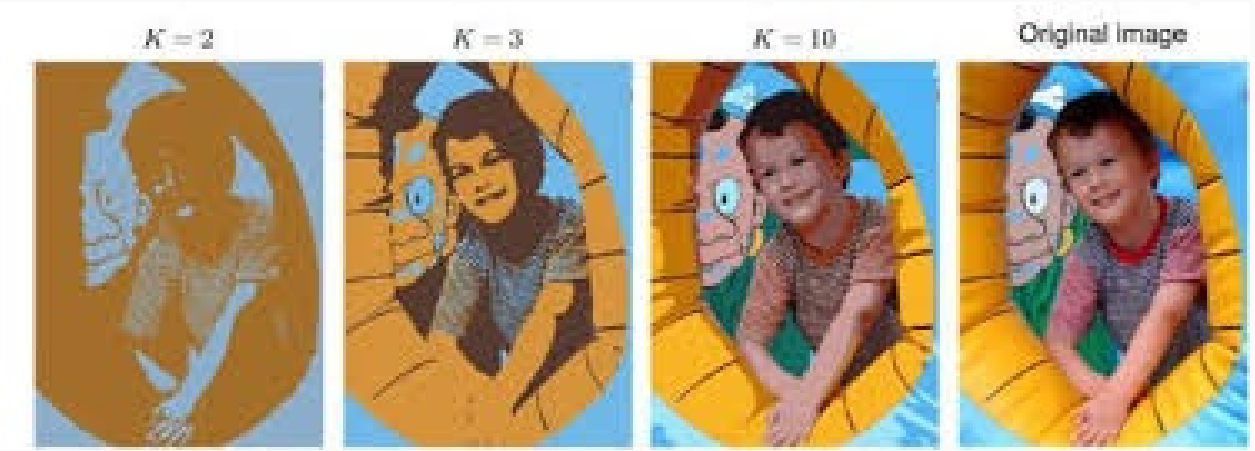


Image from C. Bishop's book, Pattern recognition and Machine Learning, Springer

Learning from labeled and unlabeled data

Semi-supervised learning ?

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

Applications

- image search, (Fergus et al., 2009)
- genomics (Shi and Zhang,2011)
- natural language parsing (Liang, 2005)
- speech analysis (Liu and Kirchhoff, 2013)

Definition of semi-supervised learning

Goal

- Labeled data : $\mathcal{S}_\ell = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$, it Assumption: i.i.d. data drawn from the joint probability distribution P(X,Y)
- Unlabeled data : $\mathcal{X}_u = \{x_{\ell+1}, \ldots, x_{\ell+u}\}$, $n = \ell + u$: available during training ! *Assumption:* i.i.d. data drawn from the marginal P(X)
- Usually $\ell << u$
- Test data : $\mathcal{X}_{test} = \{x_{n+1}, \ldots, x_{n+m}\}$: not available during training, again with *Assumption:* i.i.d. data drawn from the marginal P(X)
- **Learn a function f: $\mathcal{X} \to \mathcal{Y}$ (regression/classification) that generalizes well on test data**

Outline

Outline

Outline

Outline

Clustering algorithms

Let $\mathcal{X}$ the data space.

A data-analysis point of view

In exploratory data analysis, you are not interested on learning a function but only on labeling the available dataset $\mathcal{S}_n = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$.

$D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is supposed to be a dissimilarity (a distance without the triangle inequality).

For a given $K \in \mathbb{N}$, clustering aims at finding the best partition of $\mathcal{S}_n$ according to D. We can define this partition by a matrix C of size $n \times k$ with $c_{ij} = 1$ if $x_i \in$ cluster j, 0 otherwise.

Definitions

- Dissimilarity : distance without the triangle inequality

Between-class dispersion: for a given K-clustering $\mathcal{C}$:
$B(\mathcal{C}) = \frac{1}{2} \sum_k \sum_{i,j,C(i)=k,C(j)\neq k} d(x_i, x_j)$
Within-class dispersion:
$W(\mathcal{C}) = \frac{1}{2} \sum_k \sum_{i,j,C(i)=k,C(j)=k)} d(x_i, x_j)$
Nota bene: Total dispersion $T(x_1, \ldots, x_n) = \frac{1}{2} \sum_{i,j} d(x_i, x_j)$
$T = B(\mathcal{C}) + W(\mathcal{C})$, for all $\mathcal{C}$

Minimizing W or maximizing W is equivalent for a given dataset.

Clustering algorithms

Definition : a data-analysis point of view

Given a set of data $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, a chosen K and a dissimilarity d, you want to find a K-partition of $\mathcal{S}$, such that the between-class dispersion (inertia) is the largest or alternatively the within-class dispersion is the smallest.

Classic methods for clustering

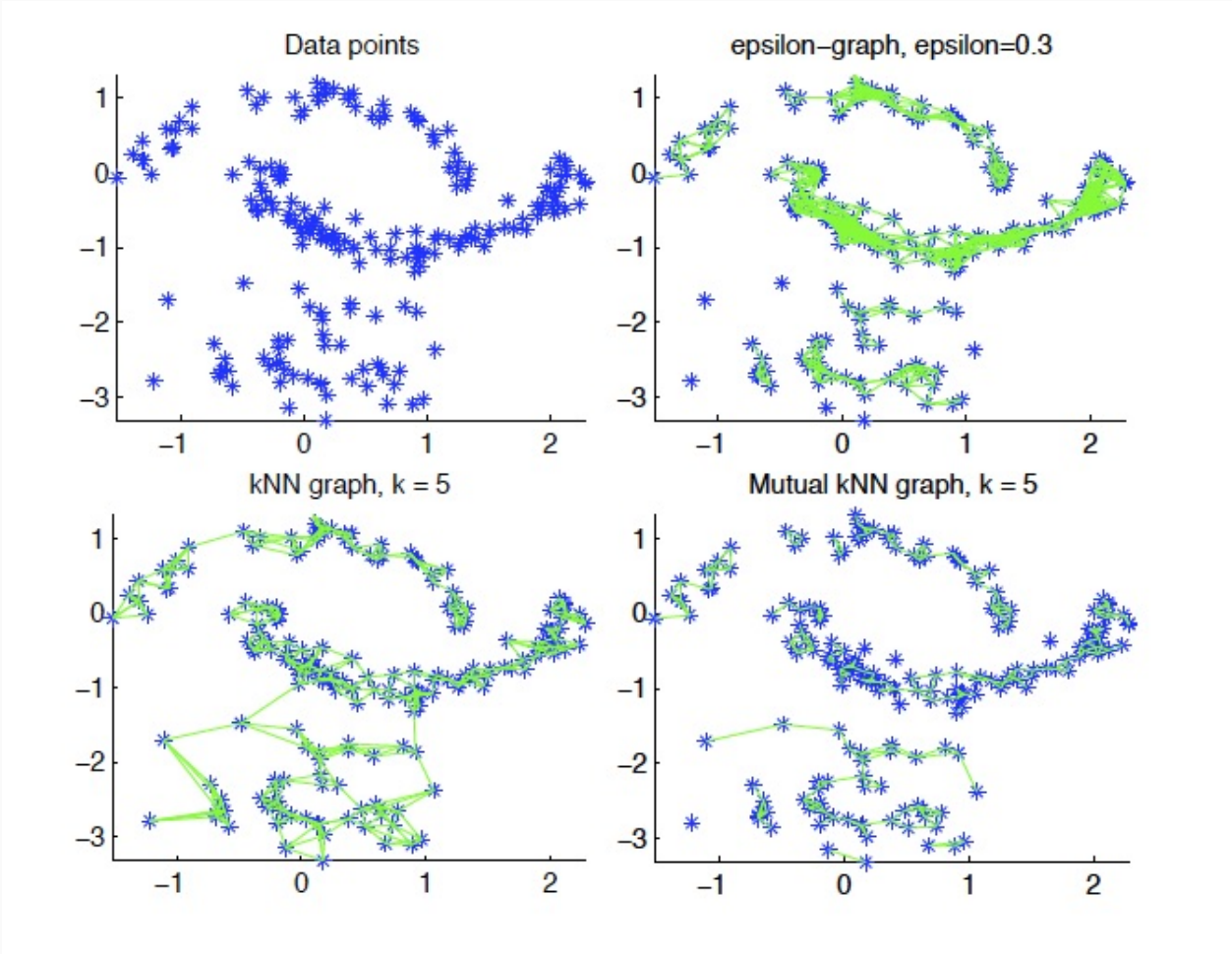
- k-means
- Hierarchical clustering
- Louvain method
- **Spectral Clustering**

Outline

Graphs and data

- Graph data: social networks, gene networks, internet, ...(see lecture of Aurelien Bellet)
- From data to graphs: a way to represent any-data

From data to graphs



Credits: Image : U. V .Luxburg.

From data to graphs

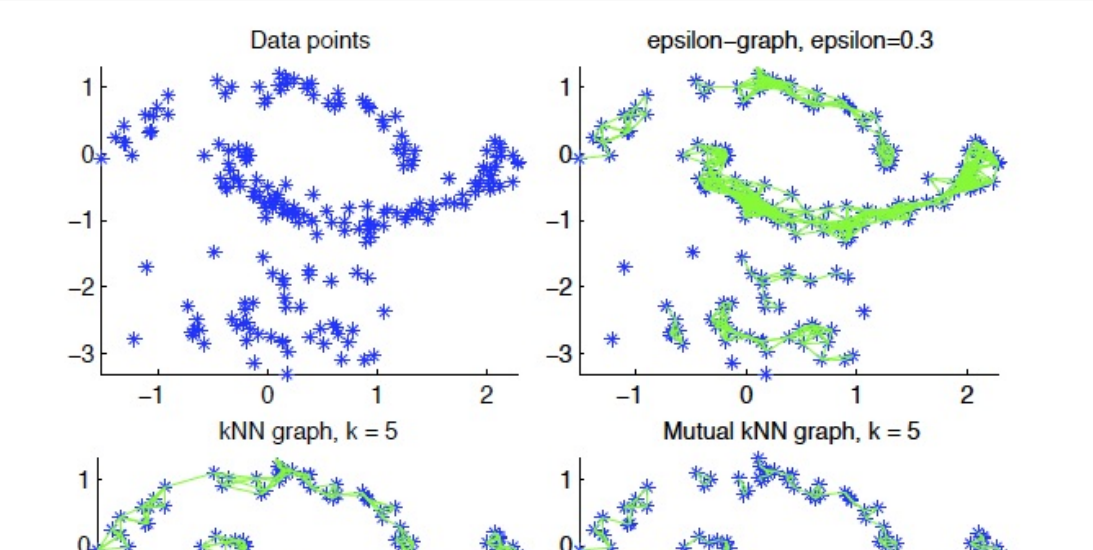
- Data $x_1, \ldots, x_n$ with their similarity values $s_{ij} \geq 0$ or with their distance d_{ij} values
- Build a graph $G = (V, E)$
- V: set of vertices. A vertex v_i corresponds to data x_i
- E : set of edges. An edge links two nodes if x_i and x_j are close according to the ϵ-graph method or the k-nn method
- W: adjacency matrix = binary symmetric matrix
- Definition: $w_{ij} = 1$ if there is an edge between node v_i and node v_j, 0 otherwise.

ϵ leverages the noise level in the graph representation: hyperparameter to keep in control.

Graph construction

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-neighbor-graph: connect v_i and v_j if x_i is among the k-nearest-neighbors of x_j OR x_i is among the k-nearest-neighbours of x_j
- k-nearest-neighbor-mutual-graph: connect v_i and v_j if x_i is among the k-nearest-neighbors of x_j AND x_i is among the k-nearest-neighbours of x_j



Graph notions

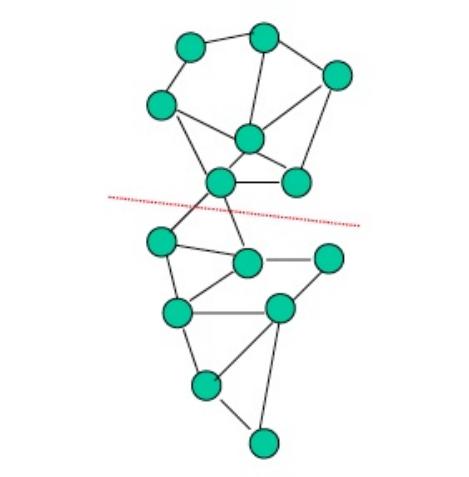
Notations : A and B are two disjoint subsets of the nodes set V that form a partition

- $cut(A, B) = \sum_{t \in A, u \in B} w_{t,u}$
- $vol(A) = \sum_{t \in A, u \in V} w_{t,u}$
- $|A| =$ nb of edges

Clustering as a min cut problem

Mincut problem

- $Cut(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} w_{ij}$
- Let $f_i \in \{-1, 1\}$ be the index class of x_i
- Clustering:=Find $(f_1, \ldots, f_n) \in \{-1, 1\}$ such that $Cut(A, \bar{A})$ is minimized.



Balanced cuts

For sake of simplicity: $B = \bar{A}$. Ratiocut:

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

Normalized cut

$$\text{Ncut}(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(B, A)}{vol(B)}$$

Elements of spectral graph theory

Some references:

- Courses/slides: Dan Spielman (Godel prize in 2015), Yale, ▸ Link
- Spectral Graph Theory, Fan R. K. Chung, Published by AMS , 1997, ▸ Link

Elements of spectral graph theory

Definitions

- W matrix : adjacency matrix
- Degree matrix D: $d_{ii} = \sum_j w_{ij}$, if $i \neq j$, $d_{ij} = 0$
- Unnormalized Graph Laplacian : $L = D - W$
- Normalized Graph Laplacians: $L_{sym} = D^{-1/2}(D - W)D^{-1/2}$, $L_{rw} = D^{-1}(D - W)$.

Graph Laplacian Properties 1/3

Eigenvalue/eigenvectors

1. L is a symmetric and positive semi-definite matrix
2. Vector 1_n is a eigenvector of L with eigenvalue 0.

Graph Laplacian Properties 2/3

Proof:

1.

$$
\begin{aligned}
f^T L f &= f^T (D - W) f \\
&= f^T D f - f^T W f \\
&= \sum_i d_i f_i^2 - \sum_{ij} w_{ij} f_i f_j \\
&= \frac{1}{2} \left(\sum_i d_i f_i^2 - 2 \sum_{ij} w_{ij} f_i f_j + \sum_j d_j f_j^2 \right) \\
&= \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2
\end{aligned}
$$

2. We notice that: $(D - W) 1_n = 0$.

Graph Laplacian Properties 3/3

Connected components
Proposition

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

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

Properties of L_{sym} and L_{rw}

The normalized Laplacians satisfy:

1. For every $f \in \mathbb{R}^n$, $f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}})^2$.
2. λ is an eigenvalue of L_{rw} with eigenvector u iff λ is an eigenvalue of L_{sym} with eigenvector: $v = D^{1/2} u$.
3. λ is an eigenvalue of L_{rw} with eigenvector u iff λ and u solve the generalized eigen problem: $Lu = \lambda Du$.
4. 0 is an eigenvalue of L_{rw} with the constant vector 1_n. 0 is an eigenvalue of L_{sym} with eigenvector $D^{1/2} 1$.

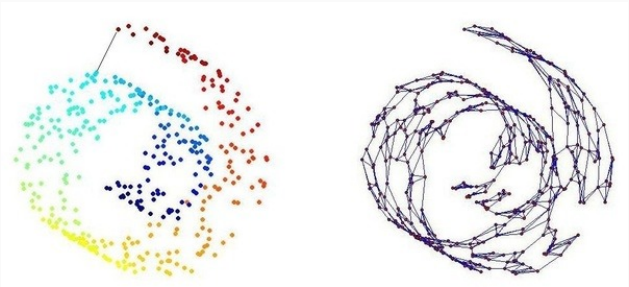
Graph function and smoothness

A function $f : V \to \mathbb{R}$.

Smoothness of the graph function:

$$\|f\|_L^2 = f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

Manifold regularization



Manifold $\mathcal{M}$: topological space that locally resembles Euclidean space near each point.

More generally, the measure of the smoothness of a function on a manifold is:

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

Outline

Come back to clustering: a balanced mincut problem

- $f_i, i = 1, \ldots, n$: membership of data i to clusters
- $f_i = 1$ if $x_i \in A$, otherwise -1 (in B)

Balanced Mincut problem
Find $f \in \{-1, 1\}^n$ that minimizes $J(f) = \sum_{i \in A, j \in B} w_{ij}$ such that $|A| = |B|$

Notice that $|A| = |B| \iff \sum_{i=1}^{n} f_i = 0$ (as many 1's than -1's).
$\sum_{i=1}^{n} f_i = 0 \iff f \perp 1_n$.

Two-ways spectral clustering: a relaxation of mincut problem

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

Constraints:

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

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

Two-ways spectral clustering

First Order Optimality Conditions to solve the optimization problem:

- Equality constraint
- Build the Lagrangian: $\mathcal{L}(f, \lambda) = f^T L f + \lambda(n - ||f||^2)$
- at the minimum, we have : $\frac{\partial L(f,\lambda)}{\partial f} = 2Lf - 2\lambda f = 0$

If we solve this eigenvector problem and take the second eigenvector, $\hat{f}$, we get $\hat{f} \perp 1$, 1 being the first eigenvector.
To get final integer values: threshold the values of f to get discrete values1 and -1 OR use 2-means (better).

k-ways spectral clustering

Algorithm

- Solve the previous relaxed problem $\rightarrow$ take the k first eigenvectors (note that you can omit $1_n = v_1$ in balanced min cut)
- 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

Variants of Spectral Clustering

- Relaxation of Ratiocut
- Relaxation of Mincut

Relaxation of Ratiocut

$$
\begin{aligned}
Ratiocut(A,B) &= \frac{cut(A,B)}{|A|} + \frac{cut(B,A)}{|B|} \\
&= cut(A,B)(\frac{1}{|A|} + \frac{1}{|B|})
\end{aligned}
$$

Define (1):

if $v_i \in A$, $f_i = \sqrt{\frac{|B|}{|A|}}$.

if $v_i \in B$, $f_i = -\frac{\sqrt{|A|}}{\sqrt{|B|}}$

Relaxation of Ratiocut

$$
\begin{aligned}
f^T L f &= \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{2}\sum_{i\in A, j\in B} w_{ij}\left(\sqrt{\frac{|B|}{|A|}} + \sqrt{\frac{|A|}{|B|}}\right)^2 + \frac{1}{2}\sum_{i\in B, j\in A}\left(-\sqrt{\frac{|A|}{|B|}} - \sqrt{\frac{|B|}{|A|}}\right)^2 \\
&= cut(A,B)\left(\frac{|B|}{|A|} + \frac{|A|}{|B|} + 2\right) \\
&= cut(A,B)\left(\frac{|A|+|B|}{|A|} + \frac{|A|+|B|}{|B|}\right) \\
&= |V| ratiocut(A,B)
\end{aligned}
$$

Relaxation of Ratiocut

We have also:

- f as defined for Ratiocut satisfies: $\sum_i f_i = 0$
- $\|f\|^2 = n$

Altogether:

Approximating Ratiocut
$\min_f f^T L f$, s.t. $f \perp 1$, $||f||^2 = n$

Normalized Spectral Clustering

- Normalized cut (avoid isolated subset) :
  $Ncut(A, B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(B,A)}{vol(B)}$
- $f_i = \sqrt{\frac{vol(B)}{vol(A)}}$ if $v_i \in A$, $\sqrt{\frac{vol(A)}{vol(B)}}$, if $v_i \in B$.
- Notice that:
  - $vol(V) = f^T Df$.
  - $(Df)^T 1 = 0$
  - $f^T Lf = vol(V)Ncut(A, B)$

Normalized Spectral Clustering

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

Normalized Spectral Clustering

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

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

The problem boils down to find second eigenvector of L_{sym}.

Normalized Spectral Clustering

Solve

$$\arg\min_{v \in \mathbb{R}^n} v^T L_{sym} v, s.t.\ v^T v = vol, v^T D^{1/2} 1_n.$$

Properties of spectral clustering

- Importance of the initial graph : several ways to construct it (k-neighbors)
- Able to extract clusters on a manifold
- Consistency (U. Von Luxburg)
- Stability
- Model selection : eigengap
- High complexity in time

How to define the cluster number ? Eigengap heuristic

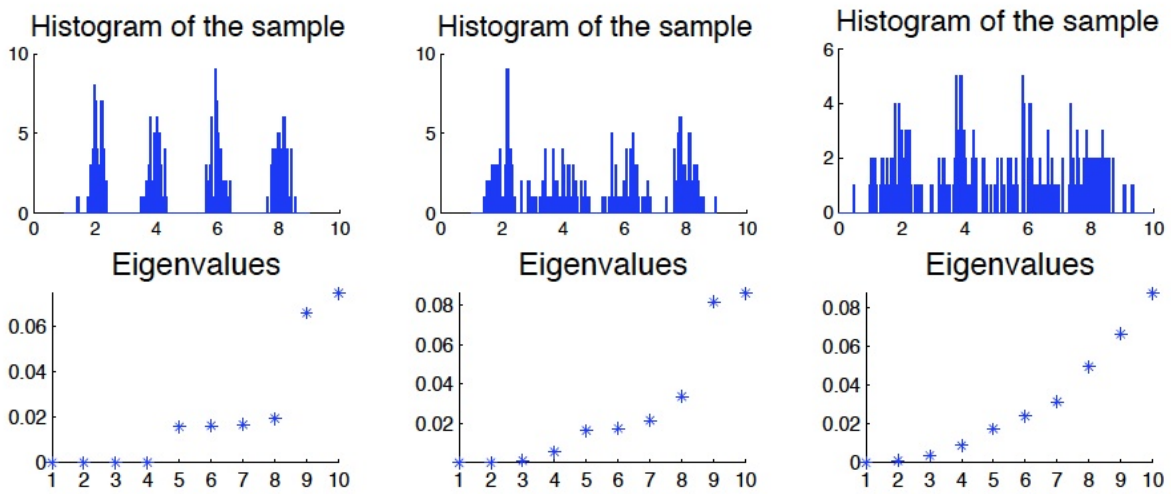
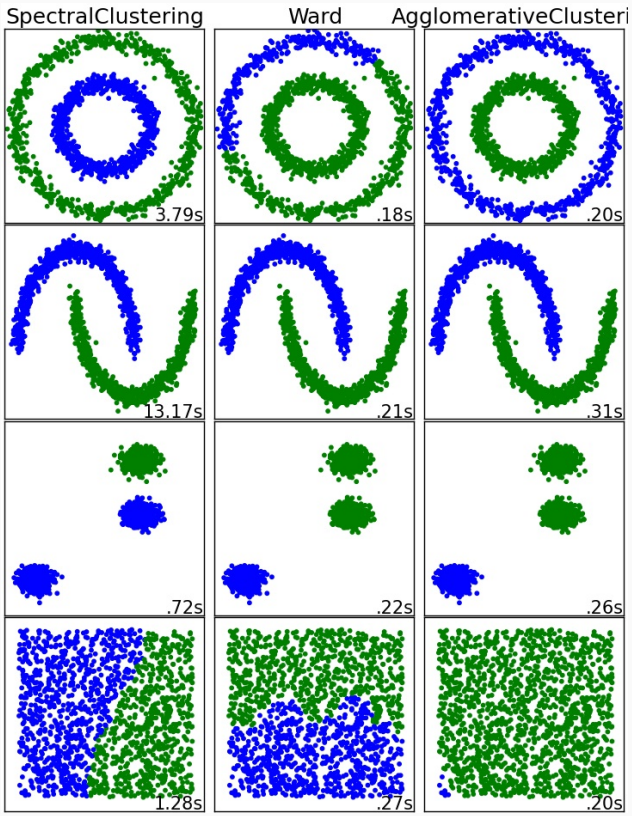


Figure 4: Three data sets, and the smallest 10 eigenvalues of L_{rw}.

- Source Tutorial U. Von Luxburg

Difficult clustering tasks



- Figure from scikitkearn:

Outline

CANNOT-LINK and MUST-LINK constraints

In real applications, it is not rare that some background knowledge might be used

MUST-LINK and CANNOT-LINK constraints

- MUST-LINK constraint: Let us define a relation ML such that $ML(i, j) = 1$ if datapoint i and datapoint indexed by j are required to be in the same cluster, 0 otherwise.
- CANNOT-LINK constraint: Let us define a relation CL such that $CL(i, j) = 0$ if datapoint i and datapoint indexed by j are required not to be in the same cluster, 0 otherwise.

Taking into account the constraints into a binary clustering task

- Now define Q a $n \times n$ matrix such that: $Q_{ij} = Q_{ji} = +1$ if $ML(i,j) = 1$, and -1 $CL(i,j) = 1$,0 if no side information is available.
- Let $f \in \{-1, +1\}^n$ be a cluster indicator vector.

$$f^T Q f = \sum_{i=1}^{n} \sum_{j=1}^{n} f_i f_j Q_{ij}$$

exactly measures how well the constraints are satisfied.
Like previously, we relax f in $\mathbb{R}^n$ and Q now in $\mathbb{R}^{n \times n}$. We want that $f^T Q f$ be important.

Constrained Clustering

Now fix α a constant and substitute f with $v = D^{-1/2}f$, the new problem is:

$$\arg\min_{v \in \mathbb{R}^n} v^T L_{sym} v, s.t.\ v^T Q_{sym} v \geq \alpha, v^T v = vol, v \neq D^{1/2} 1_n.$$

This quadratic constrained problem is solved using KKT constraints

Constrained Clustering: algorithm

Algorithm 1: Constrained Spectral Clustering

Input: Affinity matrix A, constraint matrix Q, β;
Output: The optimal (relaxed) cluster indicator $\mathbf{u}^*$;

$vol \leftarrow \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}$, $D \leftarrow \text{diag}(\sum_{j=1}^{N} A_{ij})$;
$\bar{L} \leftarrow I - D^{-1/2}AD^{-1/2}$, $\bar{Q} \leftarrow D^{-1/2}QD^{-1/2}$;
$\lambda_{max}(\bar{Q}) \leftarrow$ the largest eigenvalue of $\bar{Q}$;
if $\beta \geq \lambda_{max}(\bar{Q}) \cdot vol$ **then**
    return $\mathbf{u}^* = \emptyset$;
end
else
    Solve the generalized eigenvalue system in Eq.(13);
    Remove eigenvectors associated with non-positive eigenvalues and normalize the rest by $\mathbf{v} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|}\sqrt{vol}$;
    $\mathbf{v}^* \leftarrow \text{argmin}_{\mathbf{v}}\ \mathbf{v}^T\bar{L}\mathbf{v}$, where $\mathbf{v}$ is among the feasible eigenvectors generated in the previous step;
    return $\mathbf{u}^* \leftarrow D^{-1/2}\mathbf{v}^*$;
end

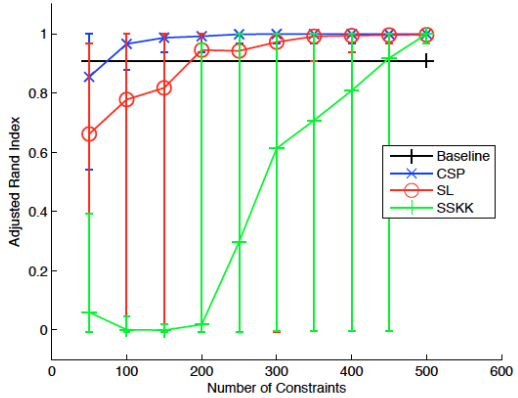
Evaluation

Constrained Clustering is typically evaluated against labeled data.

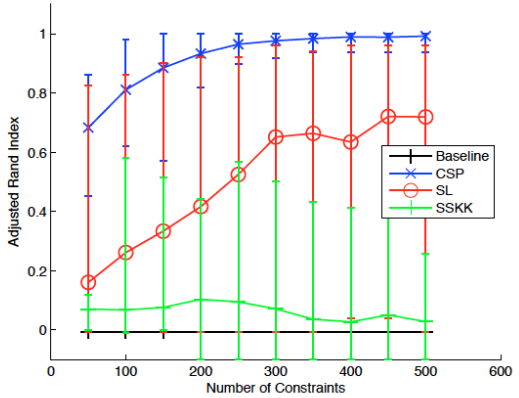
- Normalized mutual information (NMI)
- RandIndex: Accuracy for pairs of data

Measuring the interest of adding constraints is done by (artificially) varying the number of constraints

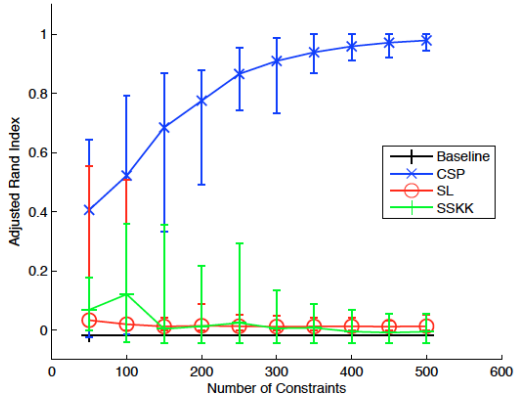
Evaluation



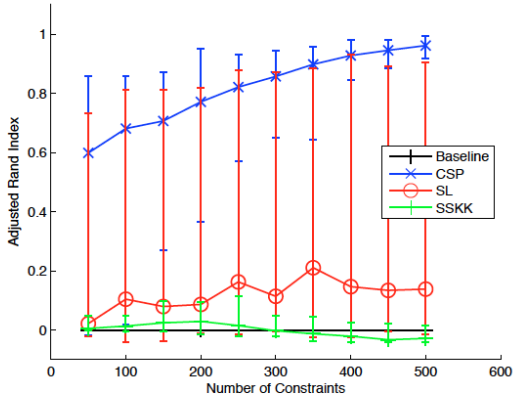
(c) Wine



(d) Glass



(e) Ionosphere



(f) Breast Cancer

Recap

- Spectral clustering : very stable method, large scale only if approximation of landmarks
- Link between Laplacian regularization in SSL and graph cuts use in spectral clustering
- Constrained Clustering: another way to take advantage of labeled or weaker information
- Constrained variants exist for near all clustering methods

Outline

Definition of semi-supervised learning

Goal

- Labeled data : $\mathcal{S}_\ell = \{(x_1, y_1), \ldots, (x_\ell, y\ell)\}$, it Assumption: i.i.d. data drawn from the joint probability distribution P(X,Y)
- Unlabeled data : $\mathcal{X}_u = \{x_{\ell+1}, \ldots, x_{\ell+u}\}$, $n = \ell + u$: available during training ! *Assumption:* i.i.d. data drawn from the marginal P(X)
- Usually $\ell << u$
- Test data : $\mathcal{X}_{test} = \{x_{n+1}, \ldots, x_{n+m}\}$: not available during training, again with *Assumption:* i.i.d. data drawn from the marginal P(X)
- **Learn a function f: $\mathcal{X} \to \mathcal{Y}$ (regression/classification) that it generalizes well on test data**

Transductive learning

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 $\ell << u$
- Goal: find $\hat{y}_{\ell+1}, \ldots, \hat{y}_{\ell+u}$
- No function f: $\mathcal{X} \to \mathcal{Y}$ to be learned !!

When does transductive learning is relevant ?

Example : information retrieval

- A user enters a query
- Search engine provides sample documents
- The user labels a subset of returned documents
- Now how to label all the document in the database ?

When is transductive learning relevant ?

Example : proteome

- A target organism:
- the set of its proteins (supposedly known)
- Some proteins have a known functional class
- Predict the functional classes for the remaining set of proteins

This course: semi-supervised learning

Assumptions for Semi-supervised learning

Learn f from $\mathcal{X}$ to $\mathcal{Y}$ using $\mathcal{S}_\ell = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ and $\mathcal{X}_u = \{x_{\ell+1}, \ldots, x_{\ell+u}\}$

Smoothness assumption of semi-supervised learning
SSL cannot work in all cases. Certain assumptions have been proposed.

- if two input points x_1 and x_2 in a high density region are close, then so should be the corresponding y_1 and y_2.

When Can Semi-Supervised Learning Work ? 1/2

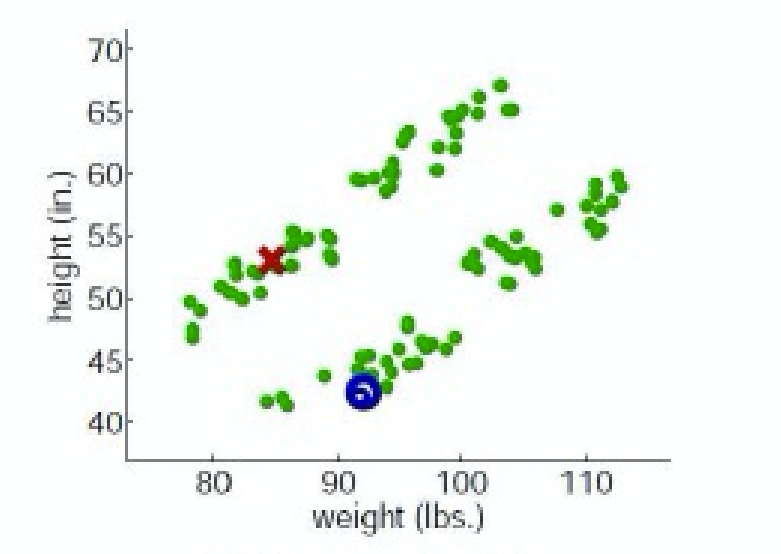
When does the knowledge brought by the unlabeled data on $p(x)$ carry information useful for $P(y|x)$?

Assumptions for both classification and regression

- (Supervised) Smoothness assumption:[1] If two points x_1, x_2 are close, then so should be the corresponding outputs y_1, y_2
- Semi-Supervised Smoothness assumption: If two points x_1, x_2 in high-density region are close, then so should be the corresponding outputs y_1, y_2
- Manifold assumption: the data lie on a low-dimensional manifold

[1] Be aware that the assumption refers to continuity rather than smoothness !

When Can Semi-Supervised Learning Work ? 2/2



Assumption for classification only

- Cluster assumption: if points are in the same cluster, they are likely to be of the same class; N.B.: a special case of the Semi-supervised Smoothness assumption.
- Low Density separation: the decision boundary should lie in a low density region

Methods

- 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 (k-NN, EM)
  - Margin for unlabeled data
  - Smoothness penalty (graph-based semi-supervised learning)

Outline

Self-training algorithm for supervised classification

- Any classifier: f

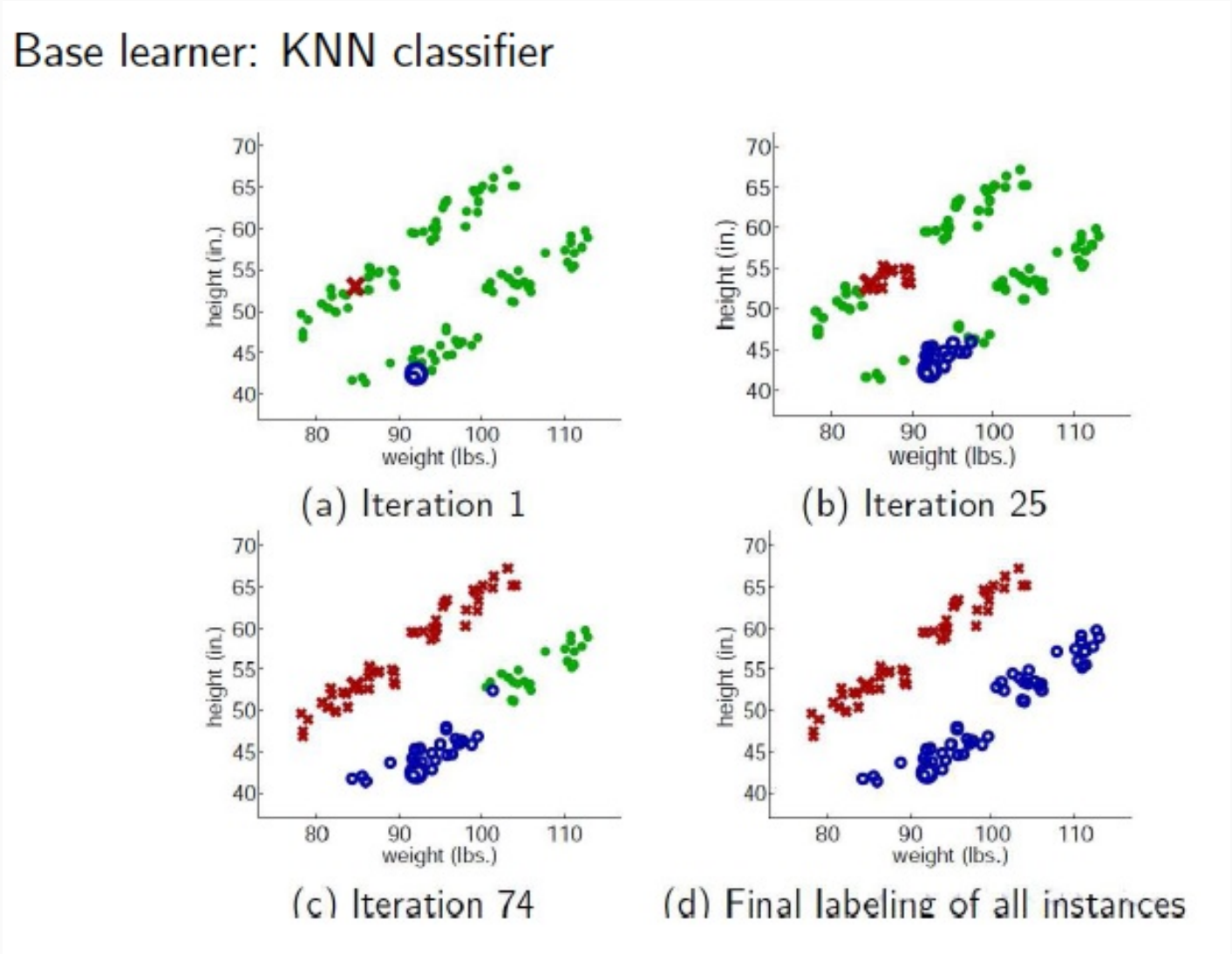
Principle

1. k=0; $\mathcal{S}_0 = \mathcal{S}$; $\mathcal{D}_0 = \emptyset$
2. Learn f_0 by training on $\mathcal{S}_0$
3. Delta $= 1000$
4. WHILE (Delta $\geq \epsilon$ and $k \leq$ Max) DO
   - Use f_k to label $\mathcal{X}_u - \mathcal{D}_k$ and get $\mathcal{D}_{k+1}$, subset of $\mathcal{X}_u - \mathcal{D}_k$ with the most confident labels predicted by f_k
   - build $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \mathcal{D}_{k+1}$
   - Learn f_{k+1} by training on $\mathcal{S}_{k+1}$
   - $Delta = Distance(f_{k+1}, f_k)$; k := k+1

How to define the most confident labels ? and how many ?

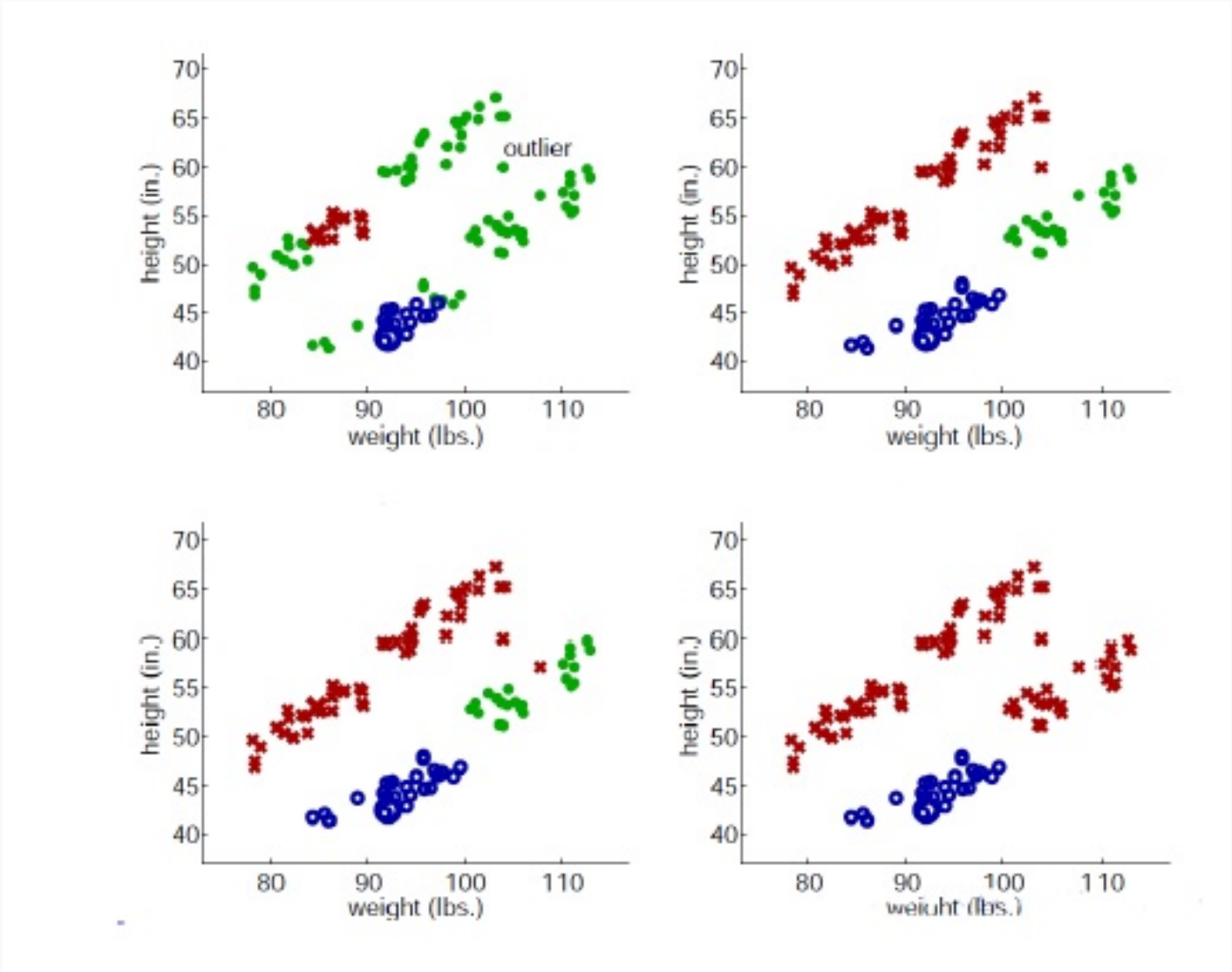
Self-training: example with k-NN (1)

- Two nice clusters without outliers [example Piyush Ray]



Self-training: example with k-NN (2)

- Two clusters with outliers



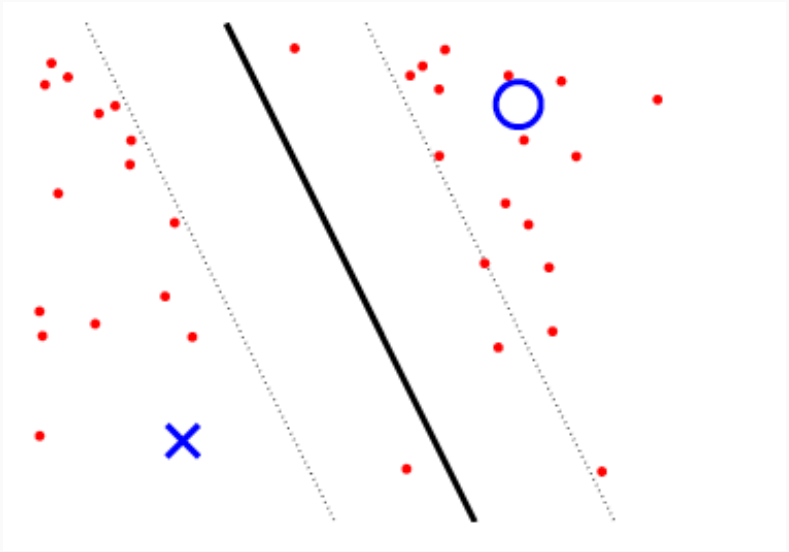
Self-training

- Pros
  - Can be used with any classifier
  - Simple
- Cons
  - Heuristic and *ad hoc* approach
  - Not well founded
  - Even in case of generative models: Avrum and Cohen (2006) showed a possible performance degradation.

Outline

Learn the unknown labels of the training set

Using Transductive SVM: here is the data



Idea: during the learning phase, learn both the parameters of the SVM and the unknown labels of $\mathcal{X}_u$

Transductive Support Vector Machine (Joachims, 1999)

Joachims proposed a Transductive SVM with a soft margin. Let us call $\mathbf{y}^* = [y_1^*, \ldots, y_u^*]$ the prediction vector for the unknown labels of the unlabeled part of the training set.

TSVM

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

under the constraints

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

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

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

$$\xi_i \geq 0$$

$$\xi_j^* \geq 0$$

Ref: Joachims, 1999.

Transductive Support Vector Machine (Joachims, 1999)

A few remarks about the nature of the method

- It is called transductive because the algorithm seems to focus on learning the unknown labels of the training set: therefore it solves a transductive problem
- However it is an inductive method in the sense that after learning you can use the resulting prediction function on new un labeled data : a single name for different variants of the method is now used: Semi-Supervised SVM or S^3VM

About the optimization problem:

- It is a combinatorial problem
- It is NP-hard to find the integer y_i^*'s !

Solution to the optimization problem in S^3VM

Keeping the exact/combinatorial problem

- Mixed integer programming method: S^3VM by Bennet and Demiriz 1999, 2001:
- Branch and bound algorithm, Chapelle, Sindwani and Keerthi, 2006

Solution to the optimization problem in S^3VM

Relaxing the exact/combinatorial problem

- Relaxation by Semi-definite programming: De Bie and Cristianini, 2004,2006
- Heuristic Joachims, 2003

Semi-definite programming for S^3VM (1)

solve the problem in the dual space

$$\min_{Y_u} \max_{\alpha} 2\alpha^T 1 - \alpha^T (K \odot YY^T)\alpha$$

under the constraints

$$Y = \begin{pmatrix} Y \\ Y_u \end{pmatrix}$$

$$Y_u = \{-1, 1\}^u$$

Semi-definite programming for S^3VM (2)

Reformulate with matrix $\Gamma = YY^T$

$$\min_{\Gamma} \max_{\alpha} 2\alpha^T 1 - \alpha^T (K \odot \Gamma)\alpha$$

under the constraints

$$\Gamma = \begin{pmatrix} Y_\ell Y_\ell^T & Y_\ell Y_u^T \\ Y_u Y_\ell^T & Y_u Y_u^T \end{pmatrix}$$

$$Y_U = \{-1, 1\}^u$$

Re-parametrize the problem in terms of Γ a sdp matrix, with rank 1 constraint. ref: De Bie, Cristianini, 2006, `http://www.tijldebie.net/publications/SSLusingSDP`

Semi-supervised learning with a new margin definition

Let us go further, we do not need to explicitly find Y_u. Let us define a margin for the unlabeled data as: $\rho(x, h) = |h(x)|$. Then,

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

under the constraints

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

$$|\mathbf{w}^T\mathbf{x}_i + \mathrm{b}| \geq 1 - \xi_i,\ i = \ell + 1, \ldots, \ell + u$$

$$\xi_i \geq 0, i = 1, \ldots, n = \ell + u$$

ref:Collobert et al., JMLR, 2006.

Use differentiable functions of the margin to solve the problem + concave - convex methods

Semi-supervised learning with a new margin definition

- 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, ...

Outline

Outline

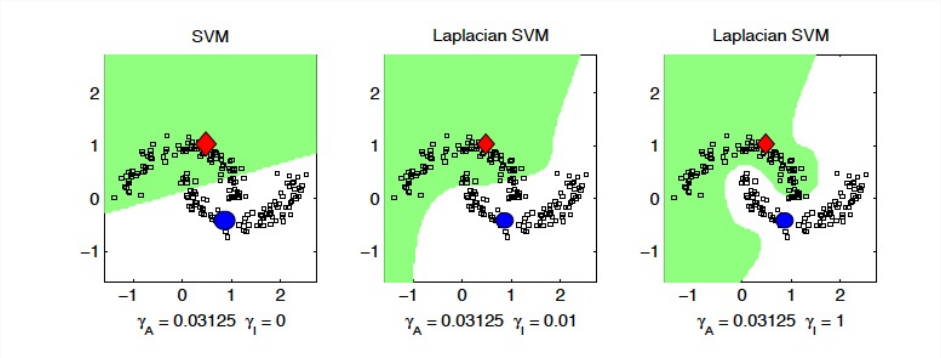
How to use the geometry of the marginal distribution P_x ?

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 similar.
- In other words: the conditional probability $P(y|x)$ varies smoothly along the geodesics in the intrinsic geometry of P_x
- so, $\int_{\mathcal{M}} ||\nabla P(Y|x)||^2 dP_x$ is small

Now for our predictive model f, we wish the same property.

Manifold regularization



- Assume that the marginal distribution P_x is known
- If $\mathcal{M}$, the support of P_x, is a submanifold $\subset \mathbb{R}^p$, we want to minimize the following penalty

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

that reflects the intrinsic structure of the distribution P_x.

The gradient of f is taken with respect to the Riemanian manifold $\mathcal{M}$, meaning that we take into account the geometry underlying the support of P_x.

Where the Laplace-Beltrami operator comes into scene

Expression of $||f||_I^2$

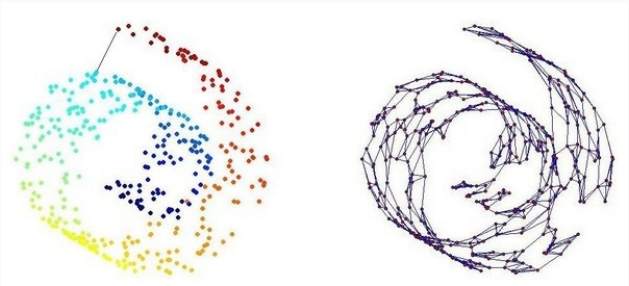
If the manifold is infinite or if the support of P_x vanishes at the boundary of $\mathcal{M}$, then the following holds:

$$||f||_I^2 = \int_{\mathcal{M}} ||\nabla_{\mathcal{M}} f||^2 dP_x = \int_{\mathcal{M}} f \mathcal{L} f dP_x = \langle f, \mathcal{L} f \rangle_{L^2(P_x)} \tag{1}$$

where $\mathcal{L}$ is the Laplace-Beltrami operator on functions:

$$\mathcal{L} f = \text{div } \nabla f$$

Approximation of Laplace-Beltrami operator



Let $\mathcal{G}$ be a graph with an adjacency matrix W. $L^2(\mathcal{G})$ is a space of functions $\mathcal{G} \to \mathbb{R}$. The graph Laplacian :

$$L = D - W$$

where D is the diagonal matrix of degrees (sum of weights of each node), is a positive definite operator on $L^2(\mathcal{G})$. Eigenfunctions of the Laplacian for an orthonormal basis for $L^2(\mathcal{G})$.

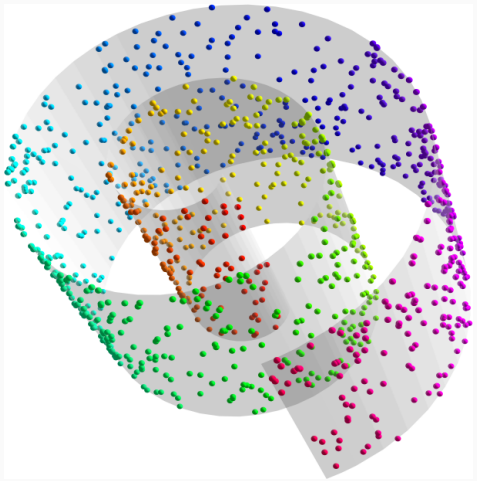
Graph function and smoothness

A function $f : V \to \mathbb{R}$.

Smoothness of the graph function:

$$\|f\|_L^2 = f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

Coming back to Manifold regularization



Manifold $\mathcal{M}$: topological space that locally resembles Euclidean space near each point.

More generally, measure of the smoothness of a function on a manifold:

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

Approximation of $\|f\|_I^2$

We have:

$$\sum_{ij} w_{ij}(f(x_i) - f(x_j))^2 = 2f^T Lf$$

Manifold regularization : smoothness on the data graph

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

Ref:Belkin, Nyogi and Sindwani (2006)

Outline

Semi-supervised learning with a smoothness constraint

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)

Semi-supervised learning with a smoothness constraint 2/2

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 $\mathcal{H}_k$:

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

Semi-supervised learning with a smoothness constraint 2/2

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 $\mathcal{H}_k$:

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

Representer theorem

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

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

Laplacian Regularized Least Square regression

- 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.
As expected, $Y^T = [y_1, \ldots, 0, \ldots 0]$, the vector that concatenates the $y_1, \ldots, y_\ell$ with the null vector of dimension u.

Laplacian SVM

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.

Laplacian SVM

In practise, we solve :

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

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

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

Laplacian SVM

In practise, we solve :

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

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

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

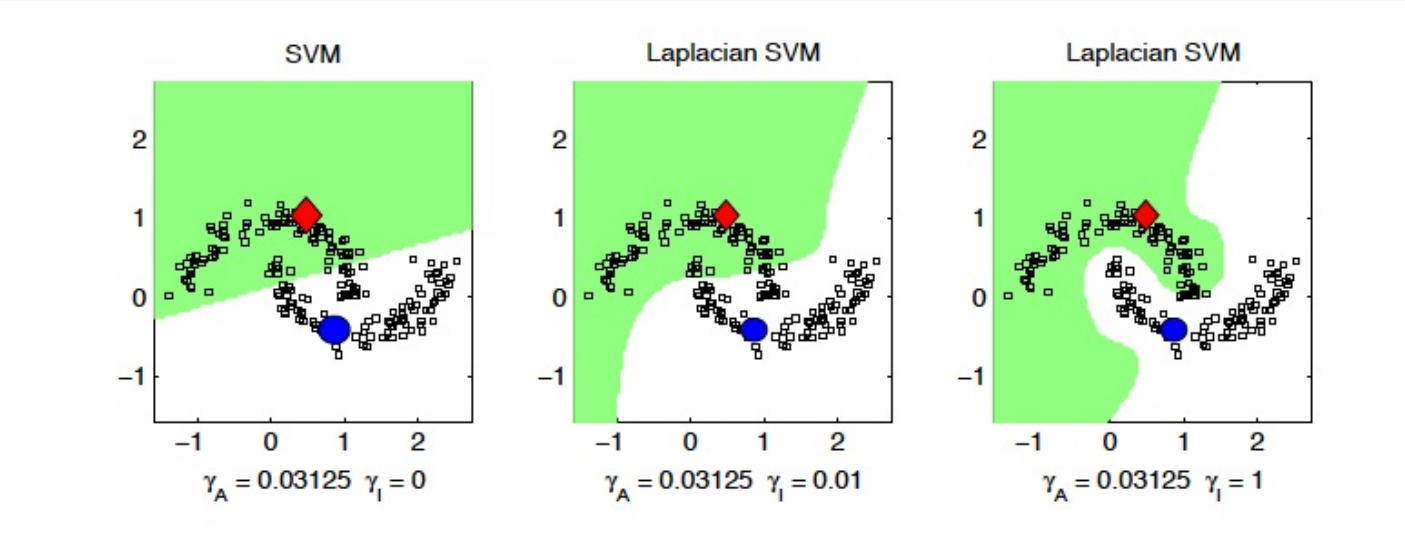
Laplacian SVM/RLS algorithm

Results: Belkin et al. 2006, in Book: Semi-supervised Learning .

	Laplacian SVM/RLS
Input:	l labeled examples $\{(x_i, y_i)\}_{i=1}^{l}$, u unlabeled examples $\{x_j\}_{j=l+1}^{l+u}$
Output:	Estimated function $f : \mathbb{R}^n \to \mathbb{R}$
Step 1	Construct data adjacency graph with $(l+u)$ nodes using, e.g, k nearest neighbors. Choose edge weights W_{ij}, e.g., binary weights or heat kernel weights $W_{ij} = e^{-\|x_i - x_j\|^2/4t}$.
Step 2	Choose a kernel function $K(x, y)$. Compute the Gram matrix $K_{ij} = K(x_i, x_j)$.
Step 3	Compute graph Laplacian matrix : $L = D - W$ where D is a diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$.
Step 4	Choose γ_A and γ_I.
Step 5	Compute α^* using Eqn (11.7) for squared loss (Laplacian RLS) or using Eqns (11.9,11.10) together with the SVM QP solver for soft margin loss (Laplacian SVM).
Step 6	Output function $f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* K(x_i, x)$.
	Equivalently, after step 4 construct the kernel function $\tilde{K}(x, y)$ given by Eqn 11.15, and use it in standard SVM/RLS (or with other suitable kernel methods).

Laplacian SVM:results

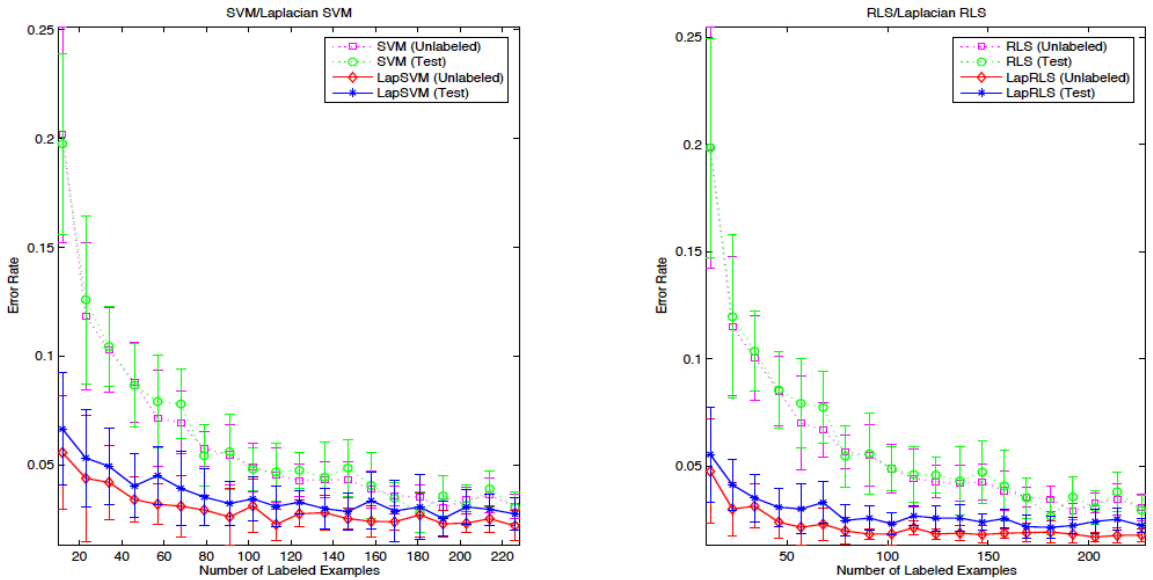
Results: Belkin et al. 2006.



Laplacian SVM:results on image classification (2 classes)

Results: Belkin et al. 2006.

Figure 11.5 Image Classification: Laplacian SVM/RLS performance with respect to number of labeled examples on unlabeled and test data.



Link with unsupervised learning

Again, working in RKHS $\mathcal{H}_k$, we would like to solve the following (relaxed) clustering problem:

$$\min_{f \in \mathcal{H}_k, \sum_i f(x_i)=0, \sum_i f(x_i)^2=1} \gamma \|f\|_k^2 + \sum_{ij} w_{ij}(f(x_i) - f(x_j))^2 \qquad (2)$$

This approach can be seen as a *regularized spectral clustering*. It also benefits from a representer theorem: f^* the minimizer of Eq. 2 satisfies:

$$f^*(\cdot) = \sum_{i=1}^{n} \alpha_i \; k(\cdot, x_i)$$

Therefore the problem boils down to solve:

$$\min_{\alpha \in n, 1^T K \alpha = 0, \alpha^T K^2 \alpha = 1} \gamma \alpha^T K \alpha + \alpha^T K L K \alpha$$

Regularized version of spectral clustering

$$\min_{\alpha \in n, 1^T K \alpha = 0, \alpha^T K^2 \alpha = 1} \gamma \alpha^T K \alpha + \alpha^T K L K \alpha$$

- show that $\alpha^* = Pv$, v being the eigenvector with the smallest eigenvalue of a generalized eigenvector problem.
- Remark: This clustering provides a natural out-of-sample extension (classification of new datapoints)

Outline

Multinomial Logistic Regression with Laplacian prior

Let us consider a C-class classification problem. We have $x_i \in \mathbb{R}^p$ and $y_i = [y_i^{(1)} \dots y_i^{(C)} \in \{0,1\}^C$ is a binary vector such that: if x_i belong to class c, then $y_i^{(c)} = 1$ and $y_i^{(j)} = 1$, for $j \neq c$. For $c = 1, \dots, C$, we define the model:

$$\log P(y^{(c)} = 1|x) = \frac{\exp((w^{(c)})^T x)}{\sum_{j=1}^{C} \exp((w^{(j)})^T x)}$$

Log-likelihood for labeled training data: $\mathcal{L}(W) = \log P(y_1, \dots, y_\ell)|x)$
Prior $p(w^{(j)})$

$$p(w^{(j)}) \propto \exp(-1/2(w^{(j)})^T(\gamma_l^{(j)} X^T L X + D^{(j)})(w^{(i)})^T),$$

where $D^{(i)}$ is a parameterized diagonal matrix which plays the role of the ℓ_2 regularization in previous approaches.
The model can be learned through Bayesian Inference and EM algorihm.

Outline

Semi-supervised embedding

3 ways proposed by Weston et al. 2008.

(a) Add a semi-supervised loss (regularizer) to the supervised loss on the entire network's output (6):

$$\sum_{i=1}^{M} \ell(f(x_i), y_i) + \lambda \sum_{i,j=1}^{M+U} L(f(x_i), f(x_j), W_{ij}) \tag{9}$$

This is most similar to the *shallow* techniques described before, e.g. equation (5).

(b) Regularize the k^{th} hidden layer (7) directly:

$$\sum_{i=1}^{M} \ell(f(x_i), y_i) + \lambda \sum_{i,j=1}^{M+U} L(f^k(x_i), f^k(x_j), W_{ij}) \tag{10}$$

where $f^k(x) = (h_1^k(x), \ldots, h_{HU_k}^k(x))$ is the output of the network up to the k^{th} hidden layer (HU_k is the number of hidden units on layer k).

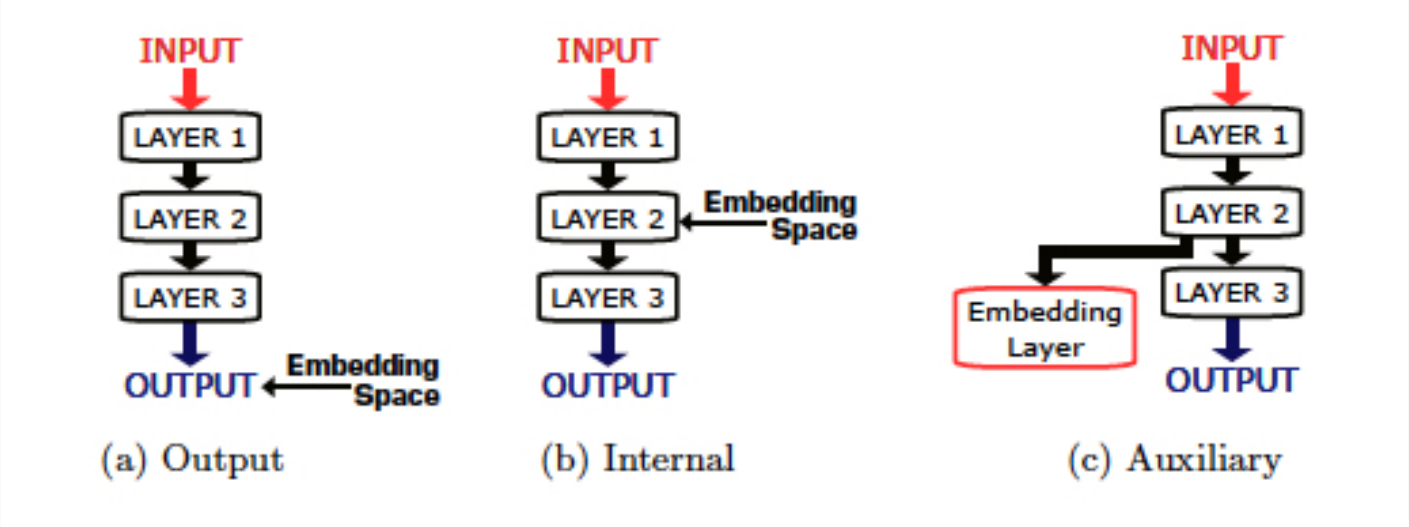
(c) Create an auxiliary network which shares the first k layers of the original network but has a new final set of weights:

$$g_i(x) = \sum_j w_j^{AUX,i} \; h_j^k(x) + b^{AUX,i} \tag{11}$$

We train this network to *embed* unlabeled data simultaneously as we train the original network on *labeled* data.

Semi-supervised embedding

Architectures



Semi-supervised embedding

Algorithm 1 *Embed*NN

Input: labeled data (x_i, y_i), $i = 1, \ldots, M$, unlabeled data x_i, $i = M+1, \ldots, U$, set of functions $f(\cdot)$, and embedding functions $g^k(\cdot)$, see Figure 1 and equations (9), (10) and (11).

repeat

Pick a random *labeled* example (x_i, y_i)

Make a gradient step to optimize $\ell(f(x_i), y_i)$

for each embedding function $g^k(\cdot)$ **do**

Pick a random pair of neighbors x_i, x_j.

Make a gradient step for $\lambda L(g^k(x_i), g^k(x_j), 1)$

Pick a random unlabeled example x_n.

Make a gradient step for $\lambda L(g^k(x_i), g^k(x_n), 0)$

end for

until stopping criteria is met.

Ref:Weston et al., ICML 2008.

Outline

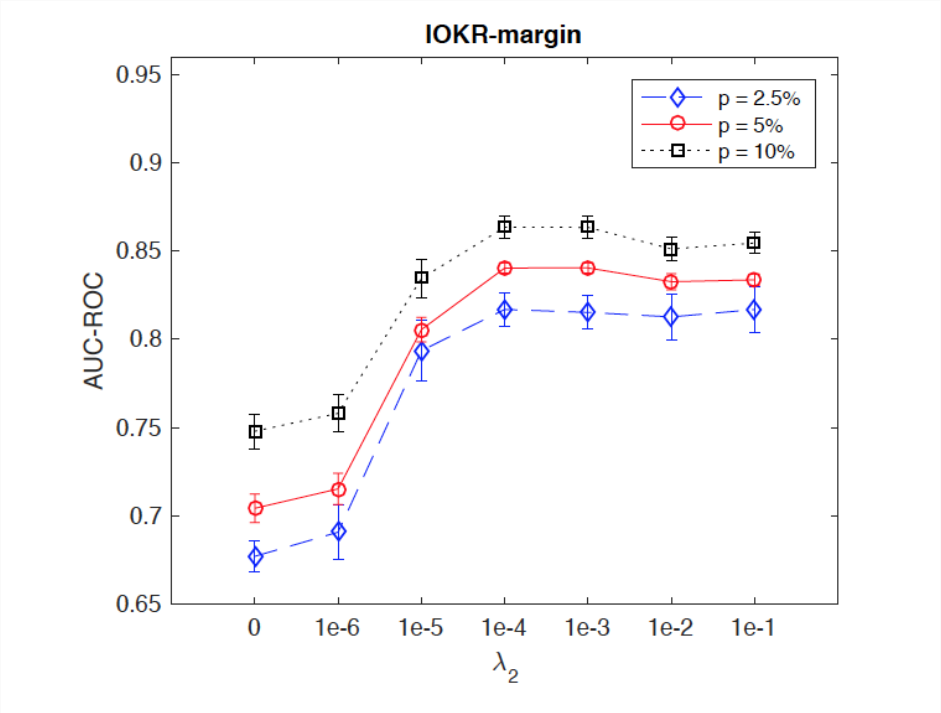
Evaluation

Realistic applications correspond to a given labeled training set of fixed size and a potentially huge unlabeled data

- In semi-supervised learning, there exist two main important "hyperparameters"
- the number of unlabeled data added to the given labeled dataset
- the hyperparameter that controls the weight of the SSL penalty

Evaluation

When assessing the performance on an existing labeled dataset, one plays with the percentage of labeled data in the training set



Outline

Outline

Semi-supervised methods

Addressing the task with

- Extension of the margin notion: $m(x, h) = |h(x)|$
  - Application: any supervised classification method with margin loss (SVM, Boosting)
  - Control: hyperparameter C leveraging the importance of unsupervised margin errors
- Graph-based regularization
  - Application: any supervised method based on regression (at least continuous, in general differentiable: kernel methods, neural networks)
  - Control: hyperparameter governing the importance of the smoothness regularization

In practise, semi-supervised learning can still bring improvement if the hyperparameter governing its weights is well chosen

Semi-supervised methods

- Self-training method no more used
- Generative methods within deep learning are still used however (group of Welling)
- Transductive SVM and S3VM variants: lack of scalability
- Manifold regularization: quite general and efficient, can be applied on any continuous machine
- In practise, semi-supervised learning can still bring improvement if the hyperparameter governing its weight is well chosen
- Assumptions and theory: a few attempts in PAC-learning theory
- Results of Maximov et al. 2016 (further reading)

References

- M.-F. Balcan, A. Blum, A discriminative model for semi-supervised learning. J. ACM 57(3): 19:1-19:46 (2010)
- **Mikhail Belkin, Partha Niyogi, Vikas Sindhwani: Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. J. Mach. Learn. Res. 7: 2399-2434**
- Thorsten Joachims, Transductive Inference for Text Classification using Support Vector Machines. ICML 1999: 200-209
- Y. Mao, M. Xi, H. Yu and X. Wang, Semi-supervised logistic regression via manifold regularization, IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, 2011.
- **J. Weston, F. Rattle, H. Mobahi, R. Collobert, Deep Learning via Semi-Supervised Embedding, Neural Networks: Tricks of the Trade (2nd ed.) 2012: 639-6552012.**