Advanced method for decision

Semi-supervised learning

PB - EG - PH - RL

UTT - OS14

May 30, 2013

Outline

Outline

Unsupervised versus supervised learning

Unsupervised Learning (clustering)

- Data : observations $\in \mathcal{X}$ without class information
- Result : groups of data (clusters)
- Use : Induced for new observations the group they belong to using the previous result.

Supervised learning (Classification)

- Data : des points étiquetés de n classes différentes (ensemble d'apprentissage)
- Result : General rule that can be apply to any sample (ie. border in the representation space $\mathcal{X}$)
- Use : deduce a label for any sample of the representation space

Semi-supervised learning

- Data: Data with label and data without label
- Semi-supervised classification: learn using the data with label and also use the data without label
- Semi-supervised clustering: clustering the data without label using the data with label

We will focus on semi-supervised classification

Benefit

Analyzing requires experts, equipment. Takes long time and cost a lot. Generally, it is difficult to obtain data with label and much less difficult to obtain data without label.

Examples :
Parsing natural language
Reaction to medication

Aim:
obtain a more efficient classifier using the data set, rather than using only the labelled data.

Profit ?

Conditions to satisfy in order that semi-supervised learning improve result: distributions of the training set should be representative of the underlying true distribution - not enough.

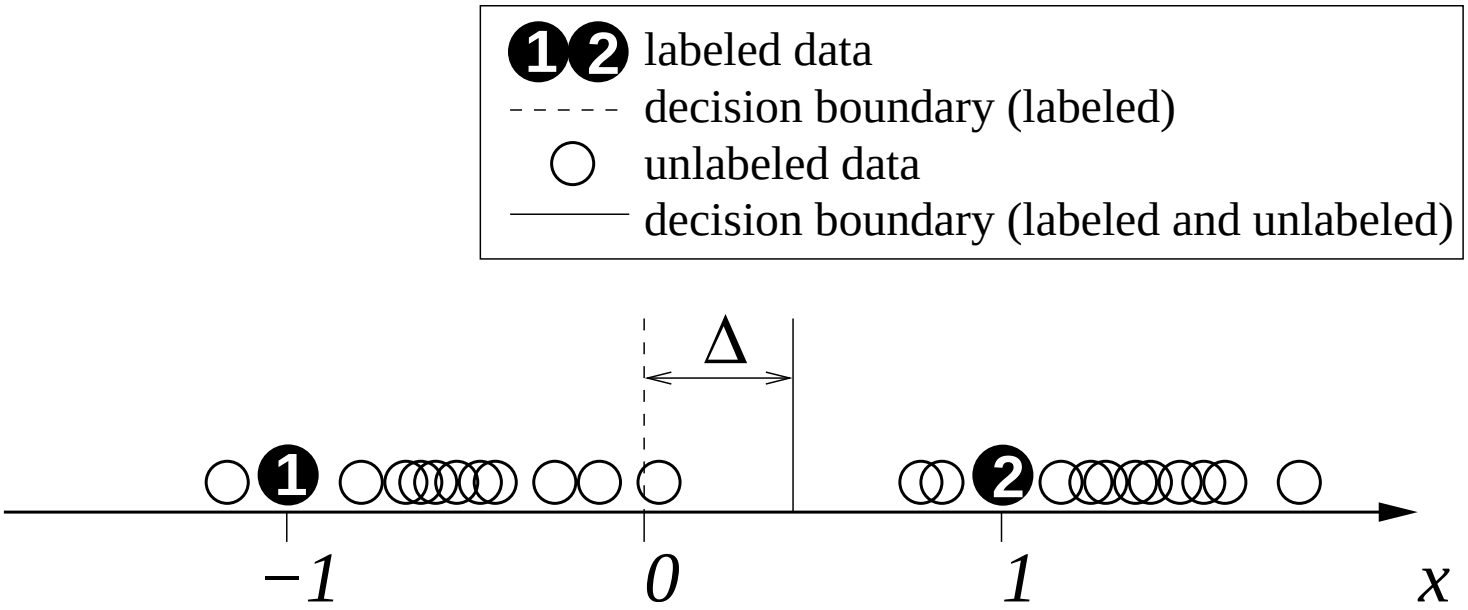
Ability to learn from unlabeled data: assumptions are made.

Assumptions on which most methods are based:

- "Smoothness" : if two samples are close in a high density area, they belong to the same class.
- group : if two samples are in the same group, they belong to the same class.
- "manifold" : data (in large dimension) are generally distributed in a small size variety

How can unlabeled data help ?

Example : assuming two unimodal classes



Limites : A bad assumption between the structure of the chosen model and the problem can degrade performance rather than improve.

Transductive learning / inductive learning

- transductive : method only works on the database (labeled and unlabeled data) $\Rightarrow$ f is a good predictor on $\mathcal{D}_u$.
- inductive : the goal is to find a general rule $\Rightarrow$ f defines a partition of the representation space $\mathcal{X}$.

Notations

- sample $x \in \mathcal{X}$,
- label $y \in \{1, \dots M\}$
- trained function $f : \mathcal{X} \mapsto \mathcal{Y}$
- labeled data
  $\mathcal{D}_l = \{(x_i, y_i) | i = 1..l\}$, $X_l = \{x_1 \dots x_l\}$, $Y_l = \{y_1 \dots y_l\}$
- unlabeled data
  $\mathcal{D}_u = \{x_{l+j}, | j = 1..m\}$, $X_u = \{x_{l+1} \dots x_{l+m}\}$,

Semi-supervised learning Methods

They are many of them:

- generative models,
- self training,
- transductive SVM,
- graph based methods,
- co-training,
- ...

Outline

Self-training principle

Principle

Use predictions to determine iteratively labels

- Build a classifier f using $\mathcal{D}_l = \{(X_l, Y_l)\}$
- Determine the labels of samples in $\mathcal{D}_u$ using $f(x)$
- Add a subset of set $\mathcal{D}_u$ to $\mathcal{D}_l$ based on the induced labels
- Repeat until all are labeled

variations

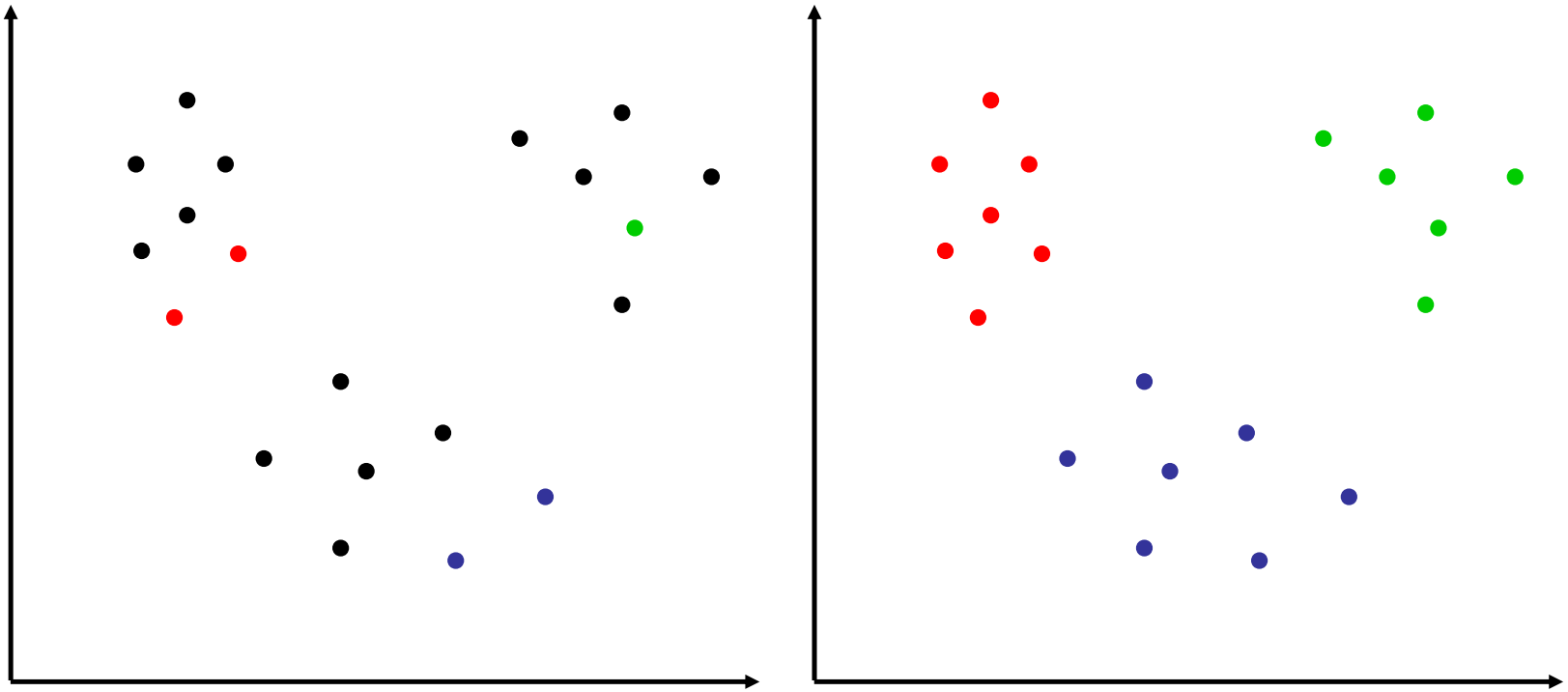
1. At each iteration, add only labeled samples $(x, f(x))$ when $\|f(x)\|$ is large
2. At each iteration, add labeled samples $(x, f(x))$ with weights that dependent on $f(x)$

Outline

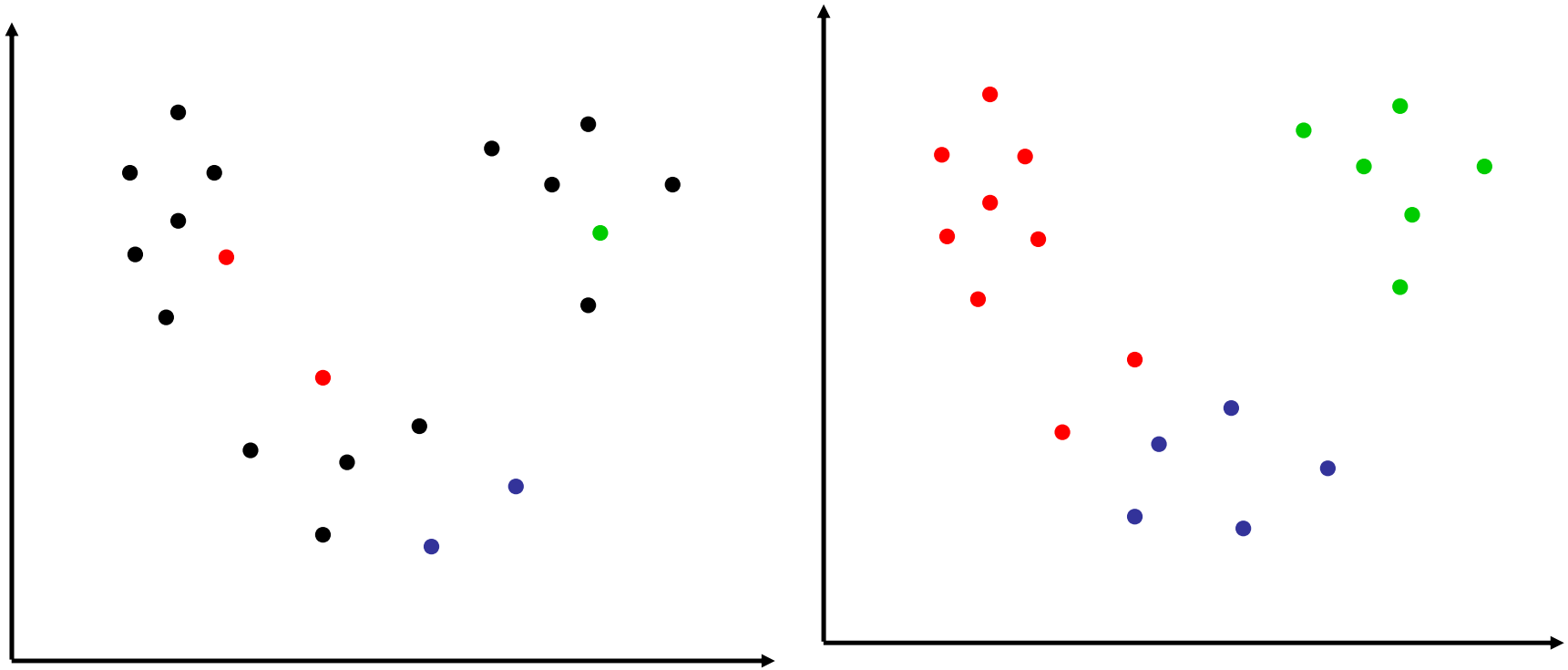
Principle

1. Determining groups using a clustering algorithm using $\{D_l, D_u\}$
2. Assign a label to each group using the labeled samples in each group (the chosen class is given by the label most represented).

Example 1



Example 2



Outline

Principle of generative models for semi-supervised learning

Principle

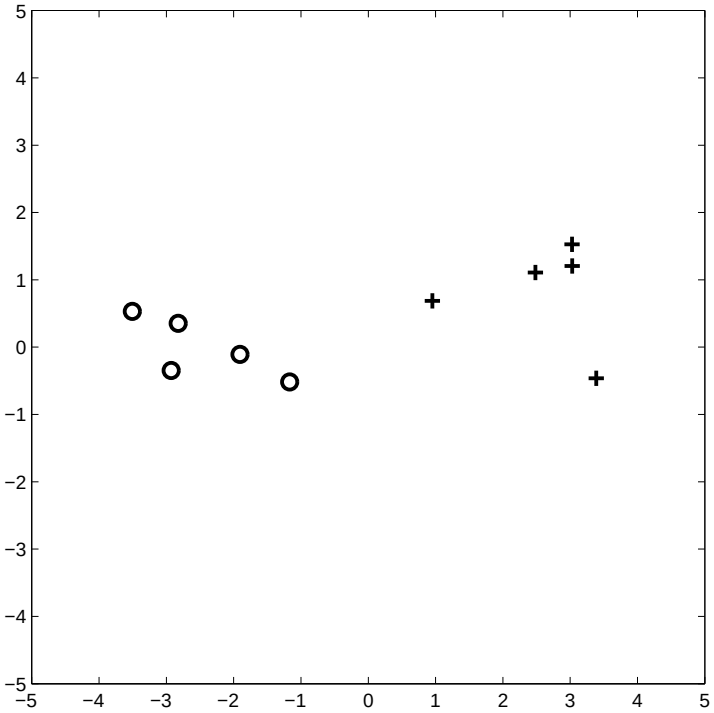
Estimate the distribution $P(X/Y)$ using a mixture model based on parameterized distributions (θ is the parameter).

Criteria based on joint and marginal likelihood :
$P(X_l, Y_l, X_u|\theta) = \sum_{Y_u} P(X_l, Y_l, X_u, Y_u|\theta)$

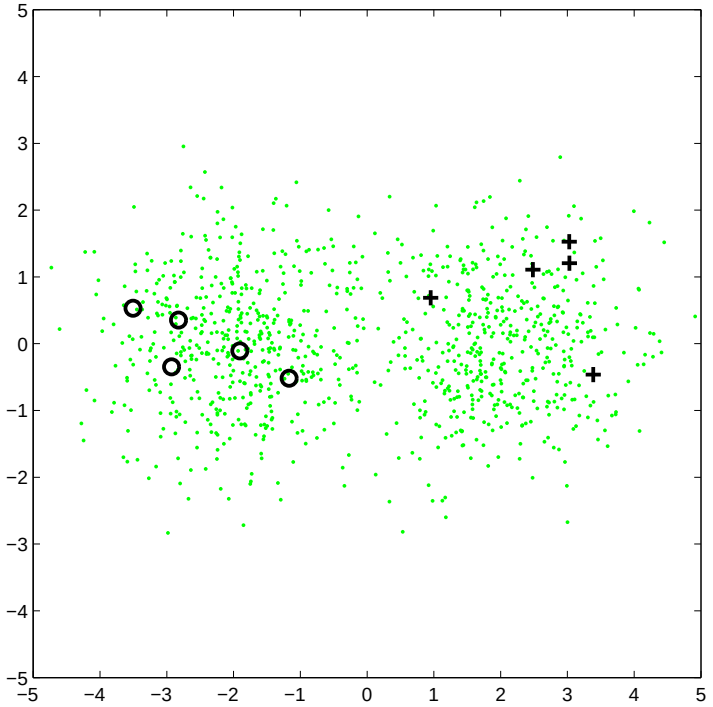
Goal: find the maximum likelihood estimate (MLE) of θ or the maximum a posterior (MAP)

Illustration

Labeled data

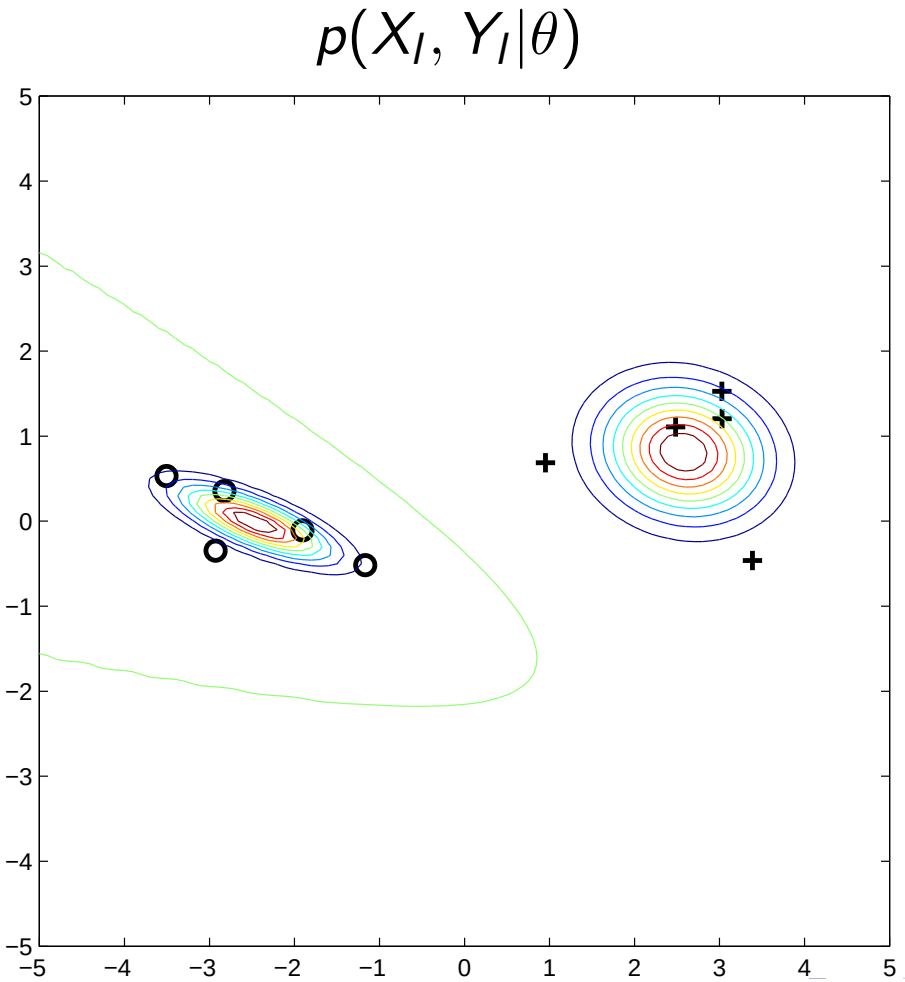


Labeled data
and unlabeled data

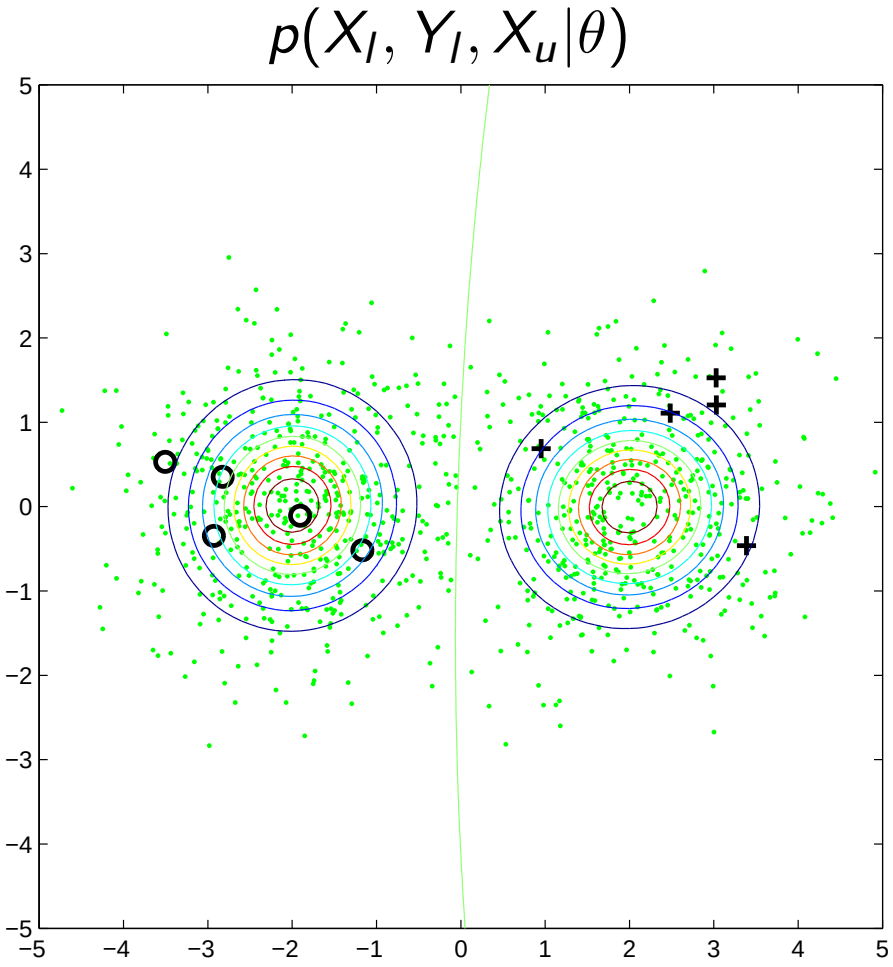


Illustration

Assuming 2 gaussian class



Illustration



MLE criteria to estimate θ

Let $\mathcal{D}$ be the data set

$$p(\mathcal{D}|\theta) = p(X_l, Y_l, X_u|\theta) = \prod_{i=1}^{l+m} p(x_i|\theta) = \prod_{i=1}^{l+m} \sum_{y=1}^{M} p(x_i|y,\theta)p(y|\theta)$$

$$= \prod_{i=1}^{l} p(x_i|y_i,\theta)p(y_i|\theta) \prod_{i=l+1}^{l+m} \sum_{y=1}^{M} p(x_i|y,\theta)p(y|\theta)$$

$$\log p(\mathcal{D}|\theta) = \log p(X_l, Y_l, X_u|\theta)$$

$$= \sum_{i=1}^{l} \log p(x_i|y_i,\theta)p(y_i|\theta) + \sum_{i=l+1}^{l+m} \log \sum_{y=1}^{M} p(x_i|y,\theta)p(y|\theta)$$

MLE criteria to estimate θ

Determination of θ :

Learning via the Expectation-Maximization (EM) algorithm (Baum-Welch) Expectation-Maximization (EM) algorithm is an iterative method to find a local optimum.

Given θ, a label is assigned to each sample $x_i \in \mathcal{D}_u$ using conditional posterior probability.
Let z_{iy} a random variable which takes value 1 if the class corresponding to the individual x_i is y and 0 otherwise.

MLE criteria to estimate θ

$$\log p(\mathcal{D}|\theta) = \sum_{i=1}^{l} \log p(x_i|y_i,\theta)p(y_i|\theta)$$

$$+ \sum_{i=l+1}^{l+m} \sum_{y=1}^{M} z_{iy} \log p(x_i|y,\theta)p(y|\theta)$$

Then

$$E(\log p(\mathcal{D}|\theta)) = \sum_{i=1}^{l} \log p(x_i|y_i,\theta)p(y_i|\theta)$$

$$+ \sum_{i=l+1}^{l+m} \sum_{y=1}^{M} E(z_{iy}) \log p(x_i|y,\theta)p(y|\theta)$$

And $E(z_{iy}) = p(y/x_i,\theta)$

EM Algorithm

The step E consist in estimating the probability $p(y/x_i, \theta)$ for each class, for all the elements of all unlabeled data

$$p(y|x_i, \theta) = \frac{p(x_i|y, \theta)p(y|\theta)}{p(x_i|\theta)}$$

$$p(y|x_i, \theta) = \frac{p(x_i|y, \theta)p(y|\theta)}{\sum_{y=1}^{M} p(y|\theta)p(x_i|y, \theta)}$$

EM Algorithm

The M step is to determine $\hat{\theta}$ that maximizes:

$$\sum_{i=1}^{l} \log p(x_i|y_i,\theta)p(y_i|\theta)$$

$$+ \sum_{i=l+1}^{l+m} \sum_{y=1}^{M} p(y|x_i,\theta) \log p(x_i|y,\theta)p(y|\theta)$$

EM Algorithm

1. Start with $\theta^{(0)}$ estimated from labeled data.

EM Algorithm

1. Start with $\theta^{(0)}$ estimated from labeled data.
2. Step E: determine the probability for each class, for all the samples of the unlabeled data set:

$$p(y/x_i, \theta^{(C)})$$

EM Algorithm

1. Start with $\theta^{(0)}$ estimated from labeled data.
2. Step E: determine the probability for each class, for all the samples of the unlabeled data set:
$$p(y/x_i, \theta^{(C)})$$
3. Step M : update the MLE:
determine $\theta^{(C+1)}$ that maximizes the MLE using $p(y/x_i, \theta^{(C)})$.

EM algorithm for Gaussian mixture model (GMM)

- Parameters of the model $\theta = \{\omega_y, \mu_y, \Sigma_y, \quad (y = 1..M)\}$
- $$p(y|\theta) = \omega_y$$
- $$p(x_i|y, \theta) = \mathcal{N}(x_i; \mu_y, \Sigma_y)$$
- Classification

$$p(y|x_i, \theta) = \frac{p(x_i, y|\theta)}{\sum_{y=1}^{M} p(x_i, y|\theta)} = \frac{\omega_y \mathcal{N}(x_i; \mu_y, \Sigma_y)}{\sum_{y=1}^{M} \omega_y \mathcal{N}(x_i; \mu_y, \Sigma_y)}$$

EM algorithm for Gaussian mixture model (GMM)

1. Start with $\theta^{(0)} = \{\omega_y^{(0)}, \mu_y^{(0)}, \Sigma_y^{(0)}, \quad (y = 1..M)\}$ estimated using labeled data.

$$\omega_y^{(0)} = \frac{1}{l}\sum_{i=1}^{l} \delta_{y_i,y} \quad \mu_y^{(0)} = \frac{\sum_{i=1}^{l} \delta_{y_i,y}\, x_i}{\sum_{i=1}^{l} \delta_{y_i,y}}$$

$$\Sigma_y^{(0)} = \frac{\sum_{i=1}^{l} \delta_{y_i,y}\, (x_i - \mu_y)^T (x_i - \mu_y)}{\sum_{i=1}^{l} \delta_{y_i,y}}$$

avec $\delta_{y_i,y} = 1$ si $y_i = y$ et 0 sinon

EM algorithm for Gaussian mixture model (GMM)

1. Start with $\theta^{(0)} = \{\omega_y^{(0)}, \mu_y^{(0)}, \Sigma_y^{(0)}, \quad (y = 1..M)\}$ estimated using labeled data.

$$\omega_y^{(0)} = \frac{1}{l}\sum_{i=1}^{l} \delta_{y_i,y} \quad \mu_y^{(0)} = \frac{\sum_{i=1}^{l} \delta_{y_i,y}\, x_i}{\sum_{i=1}^{l} \delta_{y_i,y}}$$

$$\Sigma_y^{(0)} = \frac{\sum_{i=1}^{l} \delta_{y_i,y}\, (x_i - \mu_y)^T (x_i - \mu_y)}{\sum_{i=1}^{l} \delta_{y_i,y}}$$

avec $\delta_{y_i,y} = 1$ si $y_i = y$ et 0 sinon

2. Step E : determine the probability for each class, for all the samples of the unlabeled data set:

$$p(y|x_i, \theta^{(C)}) = \frac{\omega_y^{(C)} \mathcal{N}(x_i; \mu_y^{(C)}, \Sigma_y^{(C)})}{\sum_{y=1}^{M} \omega_y^{(C)} \mathcal{N}(x_i; \mu_y^{(C)}, \Sigma_y^{(C)})}$$

EM algorithm for Gaussian mixture model (GMM)

1. Step M : update MLE
   $\theta^{(C+1)} = \{\omega_y^{(C+1)}, \mu_y^{(C+1)}, \Sigma_y^{(C+1)}, \quad (y = 1..M)\}$ estimated using data set $\mathcal{D}_l$ and the data from $\mathcal{D}_u$ labeled at previous step (E).

$$\omega_y^{(C+1)} = \frac{1}{l+m}\left(\sum_{i=1}^{l} \delta_{y_i,y} + \sum_{i=l+1}^{l+m} p(y|x_i, \theta^{(C)})\right)$$

$$\mu_y^{(C+1)} = \frac{1}{(l+m)\,\omega_y^{(C+1)}}\left(\sum_{i=1}^{l} \delta_{y_i,y}\, x_i + \sum_{i=l+1}^{l+m} p(y|x_i, \theta^{(C)}) x_i\right)$$

$$\Sigma_y^{(C+1)} = \frac{1}{(l+m)\,\omega_y^{(C+1)}}\left(\sum_{i=1}^{l} \delta_{y_i,y}\, (x_i - \mu_y)^T (x_i - \mu_y) + \sum_{i=l+1}^{l+m} p(y|x_i, \theta^{(C)})(x_i - \mu_y)^T (x_i - \mu_y)\right)$$

Generative model Examples

- Gaussian Mixture Models
  - application : image classification
  - optimisation : EM algorithm
- Mixture of multinomial distributions (Naive Bayes)
  - application : text classification
  - optimisation : EM algorithm
- Hidem Markov Model (HMM)
  - application : speech recognition
  - optimisation : Baum-Welch algorithm

Advantages of generative models

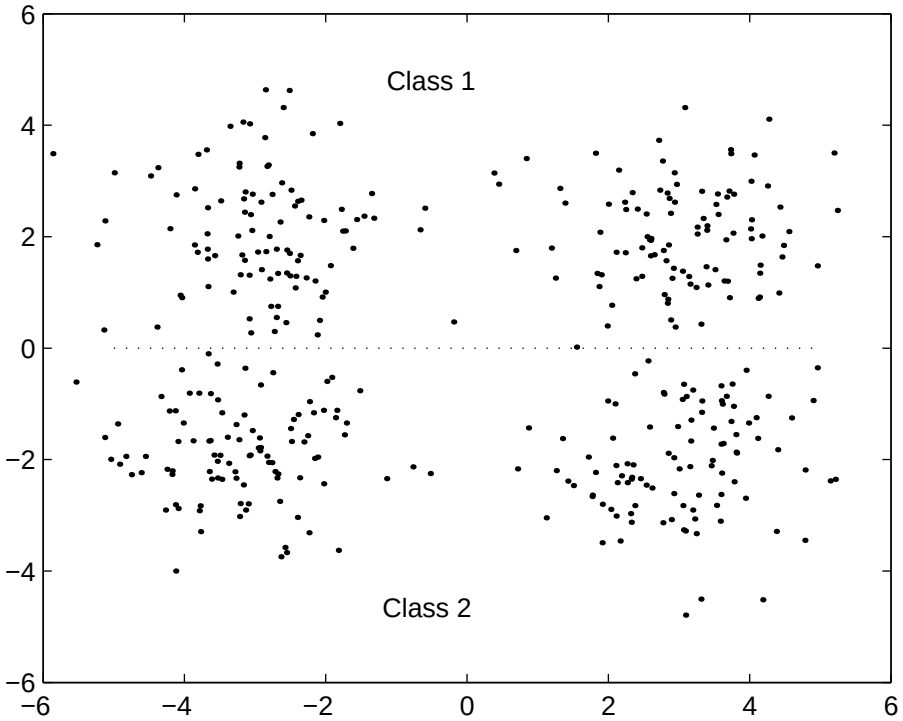
- Probabiliste approach
- Is efficient if the model fits the data well.

Limits of generative models

- Difficult to assess the accuracy of the model
- The model must be identifiable
- The EM algorithm provides a local optimum
- Unlabeled data can degrade performance if the model is wrong.

Case where unlabeled data degrade result

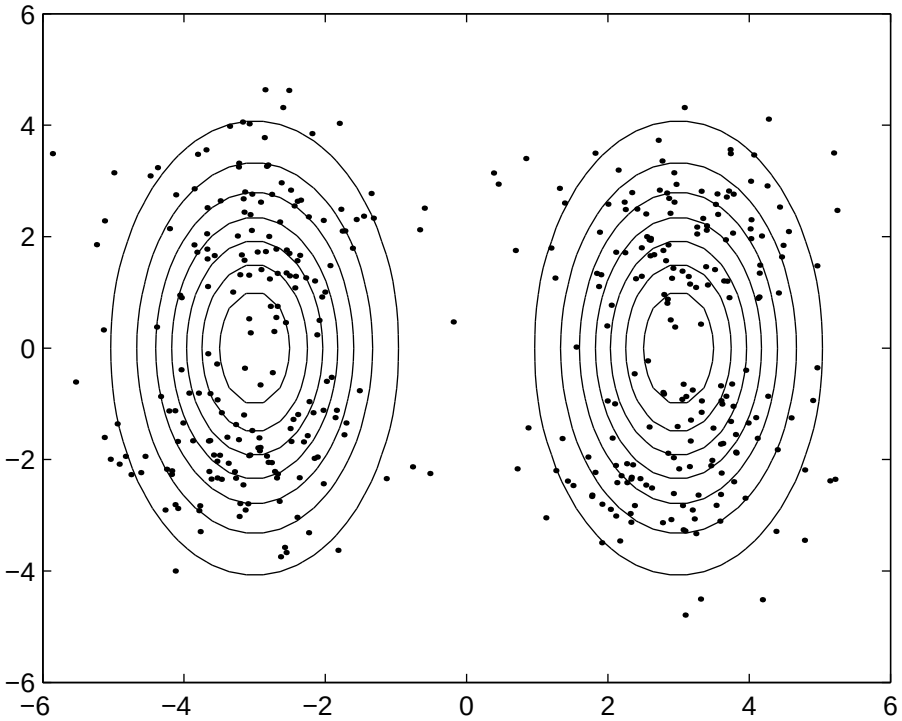
Given the following problem:



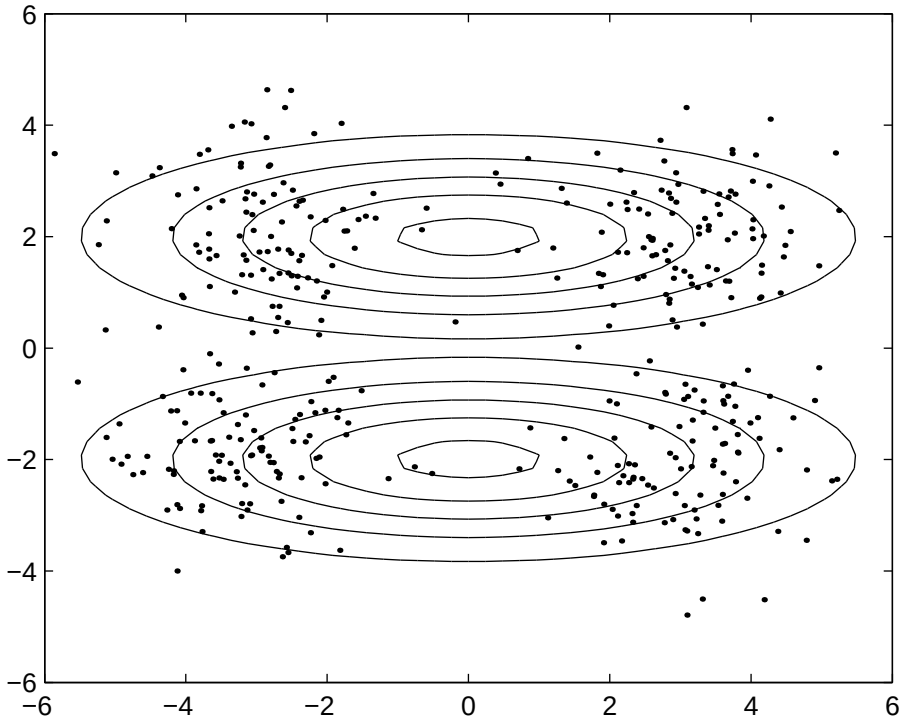
Case where unlabeled data degrade result

Likelihood results:

Hight likelihood



Low likelihood



Variation

To limit the influence of unlabeled data

The number of labeled examples is usually much lower than the number of unlabeled examples.
It is possible to balance the labeled examples of those not labeled differently:

$$\frac{1-\lambda}{l}\sum_{i=1}^{l}\log p(x_i|y_i,\theta)p(y_i|\theta)+\frac{\lambda}{m}\sum_{i=l+1}^{l+m}\log\sum_{y=1..M}p(x_i|y,\theta)p(y|\theta)$$

if $\lambda = m/(m+l)$ the initial model is found

Variation

To take into account the existence of some modes within a class:

have a model with several Gaussian distributions for each class

Outline

Graph based methods principle

Principle

Build a graph
The nodes represent the samples, labeled or not,
The edges represent the similarities between samples
Known labels are used to disseminate information in order to label all nodes.

These methods are based on the property of "smoothness" of the graph.
These are transductive methods.

Building a graph

The edge weights w_{ij} are based on a distance d_{ij} between the linked nodes.
The weight w_{ij} should be a decreasing function of the distance d_{ij}.

Building a graph

There are different approaches to build a graph:

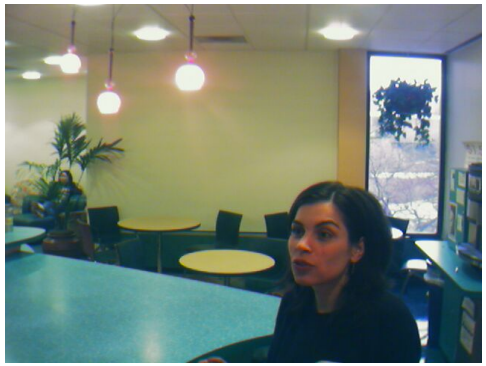
- Using a priori knowledge: special cases for this type of construction of graphs requires an understanding of the structure of the problem.

Illustration

Example: Video monitoring: the graph of the camera images is build using time, color and face edges.


image 4005


neighbor 1: time edge


neighbor 2: color edge


neighbor 3: color edge


neighbor 4: color edge


neighbor 5: face edge

Building a graph

Different approaches to build a graph (continued):

- Fixed grid: the nodes in a predefined grid are connected with a weight equal to 1.
  Example: image segmentation
- Graph *kNN* Unweighted: nodes j part of k nearest neighbors of a node i are connected by an edge which weight equals to 1.
- Graph ε-ball Unweighted: nodes i and j such that $d_{ij} \leq \varepsilon$ are connected by an edge which weight equals to 1.

Building a graph

Different approaches to build a graph (continued):

- weighted graphs: enables continuous graph with respect to the distance w_{ij} and smooth cut
  - using a tangent function

$$w_{ij} = \frac{2\arctan(\alpha_1(d_{ij} - \alpha_2) - 1)}{\pi} + 1$$

    the slope is controlled by α_1 and the position of the cut by α_2,
  - by a Gaussian function

$$w_{ij} = exp\frac{-d_{ij}^2}{2\sigma^2}$$

    the band width σ controls the decay rate.

Determination of labels

We consider real and not discrete labels.

2 classes case
Labels are given by a vector Y of dimension n. The labels are -1 or 1.
The initial vector is $\widehat{Y}^{(0)}$ with $(y_1, ...y_l, 0,\ ldots0)$ and y_i which is $+1$ or -1.
A vector $\widehat{Y}$ of dimension n is determined.
The classification is obtained by assigning to x_i class given by the sign of $\widehat{y}_i$.

N classes case Labels are given by a matrix Y of dimension $n \times c$. Labels are in $1 \dots M$.
The initial matrix is $\widehat{Y}^{(0)}$ where $Y_{ij} = 1$ if the label of x_i is j and $Y_{ij} = 0$ otherwise.
A matrix $\widehat{Y}$ of dimension $n \times c$ is determined.
The classification is obtained by assigning the label x_i given by $\arg\max_{j \leq M} \widehat{Y}_{ij}$.

Notations

- Matrix W with $w_{ij} = 0$ if the nodes i and j are not connected.
- Diagonal matrix D defined by $D_{ii} = \sum_j w_{ij}$ and $D_{ij,i \neq j} = 0$
- Non-normalised Laplacien Graph $L = D - W$
- Normalised Laplacien Graph
$\mathcal{L} = D^{-1/2} L D^{-1/2} = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2} W D^{-1/2}$

Modified version

- W' tel que $w'_{ij} = w_{ij}$ pour $i \neq j$ et $w'_{ii} = 0$
- $D'_{ii} = \sum_j w'_{ij} = D_{ii} - w_{ij}$
- $\mathcal{L}' = D'^{-1/2}(D' - W')D'^{-1/2} = I - D'^{-1/2} W' D'^{-1/2}$

- Matrix S : $S_{ii} = 1$ for $1 \leq i \leq l$ and 0 otherwise

Propagation of labels using iterative algorithm

Algorithm 1 (Zhu et Ghahramani (2002))

Initialize $\widehat{Y}^{(0)}$ with $(y_1, ...y_l, 0, \ldots 0)^T$

Repeat

1. Determine

$$\widehat{Y}^{(t+1)} = D^{-1} W \widehat{Y}^{(t)}$$

namely:

$$\widehat{y_i}^{(t+1)} = \frac{\sum_j w_{ij} \widehat{y_j}^{(t)}}{\sum_j w_{ij}}$$

2. Constraints associated to data with label: $\widehat{Y}_l^{(t+1)} = Y_l$

Until convergence.

Give x_i label corresponding to the sign of $\widehat{y}_i^{(\infty)}$.

Propagation of labels using iterative algorithm

Algorithm 2 (inspired by Jacobi)

Algorithm that enable to

- force $w_{ii} = 0$
- allow $\widehat{y}_l \neq y_l$
- add a regularization term to improve the numerical stability.

Propagation of labels using iterative algorithm

Algorithm 2 (inspired by Jacobi)

Algorithm :
Initialization:

- Choose a parameter $\alpha \in [0\ , 1[$ and a parameter $\epsilon \simeq 0$.

Let $\mu = \frac{\alpha}{1-\alpha} \in [0\ , \infty[$

- Compute the diagonal matrix A such that $A_{ii} = S_{ii} + \mu D'_{ii} + \mu\epsilon$
- Initialize $\widehat{Y}^{(0)}$ with $(y_1, ...y_l, 0, \ldots 0)^T$
- Repeat until convergence

$$\widehat{Y}^{(t+1)} = A^{-1}(\mu W' \widehat{Y}^{(t)} + \widehat{Y}^{(0)})$$

Give to x_i the label which corresponds to the sign of $\widehat{y}^{(\infty)}$.
The parameter ϵ avoids numerical problem when de denominator is small.

Propagation of labels using iterative algorithm

Algorithm 2 (inspired by Jacobi)

The solution can be written as follow:

$$\widehat{y_i}^{(t+1)} = \frac{1}{S_{ii} + \mu D'_{ii} + \mu\epsilon} \left(S_{ii} y_i + \mu \sum_j w'_{ij} \widehat{y_j}^{(t)} \right)$$

namely *for the data*:

with the labels $(1 \leq i \leq l)$: *without labels* $(l+1 \leq i \leq n)$:

$$\widehat{y_i}^{(t+1)} = \frac{\sum_j w'_{ij} \widehat{y_j}^{(t)} + \frac{1}{\mu} y_i}{\sum_j w'_{ij} + \frac{1}{\mu} + \epsilon} \qquad \widehat{y_i}^{(t+1)} = \frac{\sum_j w'_{ij} \widehat{y_j}^{(t)}}{\sum_j w'_{ij} + \epsilon}$$

Propagation of labels using iterative algorithm

Algorithm 3 (Zhou et al. (2004))

Algorithm :

Initialization:

- Choose a parameter $\alpha \in [0\ 1[$
- Initialize $\widehat{Y}^{(0)}$ with $(y_1, ...y_l, 0, \ldots 0)^T$

Reapeat until convergence:

$$\widehat{Y}^{(t+1)} = \alpha(I - \mathcal{L}')\widehat{Y}^{(t)} + (1-\alpha)\widehat{Y}^{(0)}$$

We can show that:

$$\widehat{Y}^{(t)} \rightarrow \widehat{Y}^{(\infty)} = \left(I + \frac{\alpha}{1-\alpha}\mathcal{L}'\right)^{-1} \widehat{Y}^{(0)}$$

Determination of the labels using a quadratic cost

Consistency with the labeled data can be measured by:

$$\sum_{i=1}^{l}(\widehat{y}_i - y_i)^2 = \| \widehat{Y}_l - Y_l \|^2$$

Consistency with data geometry can be measured by:

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}(\widehat{y}_i - \widehat{y}_j)^2$$

$$= \frac{1}{2}\left(2\sum_{i=1}^{n}\widehat{y}_i^2\sum_{j=1}^{n} w_{i,j} - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}\widehat{y}_i\widehat{y}_j\right)$$

$$= \widehat{Y}^T(D - W)\widehat{Y} = \widehat{Y}^T L\widehat{Y}$$

Different variants have been proposed.

Determination of the labels using a quadratic cost

Criteria 1

Set $\widehat{Y}_l = Y_l$ and determine Y_u to minimize

$$\widehat{Y}^T L \widehat{Y}$$

Solution
The minimum function is harmonic. It satisfies: For labeled samples

$$\widehat{Y}_l = Y_l$$

For unlabeled samples

$$\widehat{Y}_u = -L_{uu}^{-1} L_{ul} \widehat{Y}_l$$

where L_{uu} and L_{ul} result of the division of L in 4 parts

$$L = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix}$$

Determination of the labels using a quadratic cost

Criteria 2 : Cost function proposed by Belkin and al (2004)

Criterion which minimizes errors while penalizing the non-homogeneity:

$$C(\widehat{Y}) = \| \widehat{Y}_l - Y_l \|^2 + \mu \widehat{Y}^T L \widehat{Y} + \mu\epsilon \| \widehat{Y} \|^2$$

$$= \| S(\widehat{Y} - Y) \|^2 + \mu \widehat{Y}^T L \widehat{Y} + \mu\epsilon \| \widehat{Y} \|^2$$

$\mu \in [0\ \infty[$ is the weighting parameter for inhomogeneity.
The last term is a regularization term (parameter $\epsilon \simeq 0$).

Solution

$$\widehat{Y} = (S + \mu L + \mu\epsilon I)^{-1} SY$$

Determination of the labels using a quadratic cost

Criteria 3 : Cost function proposed by Zhou and al (2004)

$$C(\widehat{Y}) = \sum_{i=1}^{l}(\widehat{y}_i - y_i)^2 + \sum_{i=l+1}^{n} \widehat{y}_i^2 + \frac{\mu}{2}\sum_{i,j} W'_{ij}\left(\frac{\widehat{y}_i}{\sqrt{D'_{ii}}} - \frac{\widehat{y}_j}{\sqrt{D'_{jj}}}\right)^2$$

$$C(\widehat{Y}) = \| \widehat{Y}_l - Y_l \|^2 + \| \widehat{Y}_u \|^2 + \mu \widehat{Y}^T \mathcal{L}' \widehat{Y}$$

Solution

$$\widehat{Y} = (I + \mu\mathcal{L}')^{-1} SY$$

Equivalence between iterative methods and cost approaches

It is possible to show that some solutions obtained using the iterative methods of propagation of the label are equivalent to solutions obtained by minimizing a quadratic cost criterion.

To show the equivalence, use the result of Saad (1996): An approximate solution of the system

$$Mx = b$$

can be obtained iteratively. The approximate solution at step $t+1$ is given by:

$$x_i^{(t+1)} = \frac{1}{M_{ii}} \left(b_i - \sum_{i \neq j} M_{ij} x_j^{(t)} \right)$$

Equivalence between iterative methods and cost approaches

We can show the equivalence between:

- algorithm 2 et criteria 2
- algorithm 3 et criteria 3

Manifold regularization

Estimate a function f which minimizes

$$\frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2$$

- V is a loss function
- $\|f\|_K$ is a norm corresponding to the kernel K (RKHS)
- $\|f\|_I$ is a penalty term which must reflect the intrinsic structure of the marginal probability density of data P_X
- γ_A control the complexity of the function
- γ_I control the complexity of the intrinsic structure of the marginal probability density of data P_X

Manifold regularization

- Since P_X is unknown, we can only have an estimate of $\|f\|_I$.
  For example
  $$\|f\|_I^2 = \frac{1}{(l+m)^2}\widehat{f}^T \Delta \widehat{f}$$
  where $\widehat{f}$ is estimated from the set of labeled data and all unlabeled data.

- For $V(x_i, y_i, f)$
  - LapRLS (Laplacian Regularized Least Squares)
    $$V(x_i, y_i, f) = (y_i - f(x_i))^2$$
  - LapSVM (Laplacian SVM)
    $$V(x_i, y_i, f) = max(0, 1 - y_i f(x_i))$$

Algorithm for Manifold regularization

1. Build the graph
2. Choose a kernel and compute the Gram matrix $K_{ij} = K(x_i, x_j)$
3. Compute Δ
4. Choose γ_I and γ_A
5. Compute α^* which depends on V
6. Determine the function f: $f = \sum_{i=1}^{l+m} \alpha^* K(x_i, x_j)$

Advantages / Disadvantages of methods based on graphs

Avantages

- Clear mathematical framework
- Good performance if the graph is correct

Inconvénients

- Sensitive to the structure of the graph and edge weights
- Poor performance if the graph is bad

Outline

Hypothesis

Hypothesis

- x data can be separated into two sets $[x^{(1)}, x^{(2)}]$
- $x^{(1)}$ and $x^{(2)}$ can be use to learn good classifiers
- $x^{(1)}$ and $x^{(2)}$ are conditionally independent

Principle

Principle

- Learn two classifiers $f^{(1)}$ and $f^{(2)}$
- classified unlabeled samples with two classifiers separately
- Add the most likely samples $(x, f^{(1)}(x))$ to classifier's data $x^{(2)}$
- Add the most likely samples $(x, f^{(2)}(x))$ to classifier's data $x^{(1)}$
- repeat

Outline

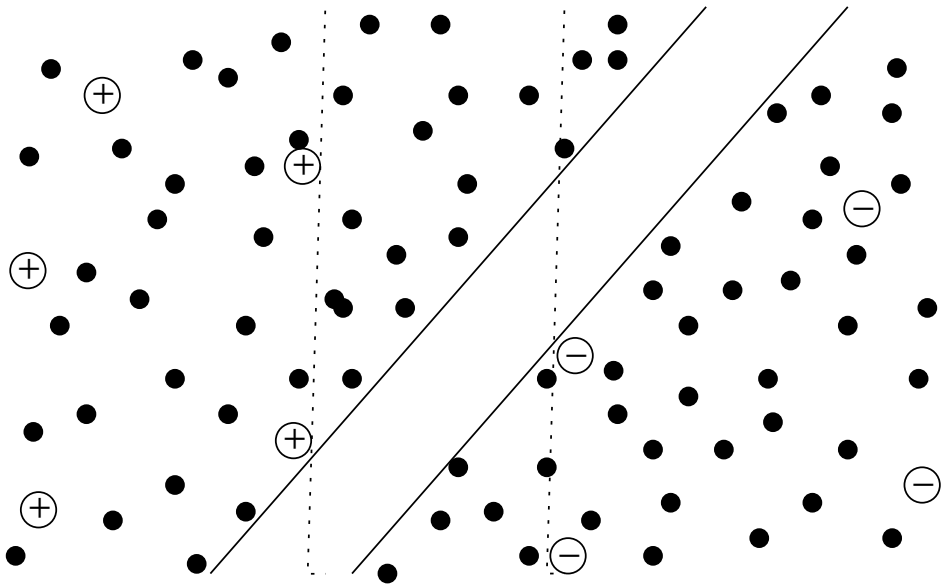
Principle of semi-supervised SVM

Principle

Extension of Support Vector Machines when unlabeled data are available

Short Name: Originally called Transductive Support Vector Machines (TVM)
and then Semi-Supervised Support Vector Machines (S3VM)
since it is a inductive method and not a transductive one.

Illustration



Supervised soft margin SVM

Goal : find the largest possible margin with as much as possible samples outside the margin and well classified.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i$$

Subject to contraints $y_i f(x_i) \geq 1 - \xi_i \quad$ with $\xi_i \geq 0 \quad \forall i = 1 \ldots l$

où

$$f(x_i) =< w, x_i > +b$$

Supervised soft margin SVM

Equivalent formulation with a loss function

To minimize ξ_i subject to the constraint $y_i f(x_i) \geq 1 - \xi_i$ with $\xi_i \geq 0$, the constraint can be restated as $\xi_i \geq 1 - y_i f(x_i)$ with $\xi_i \geq 0$, thus

- if $1 - y_i f(x_i) \leq 0$ then min $\xi_i = 0$
- if $1 - y_i f(x_i) > 0$ then min $\xi_i = 1 - y_i f(x_i)$

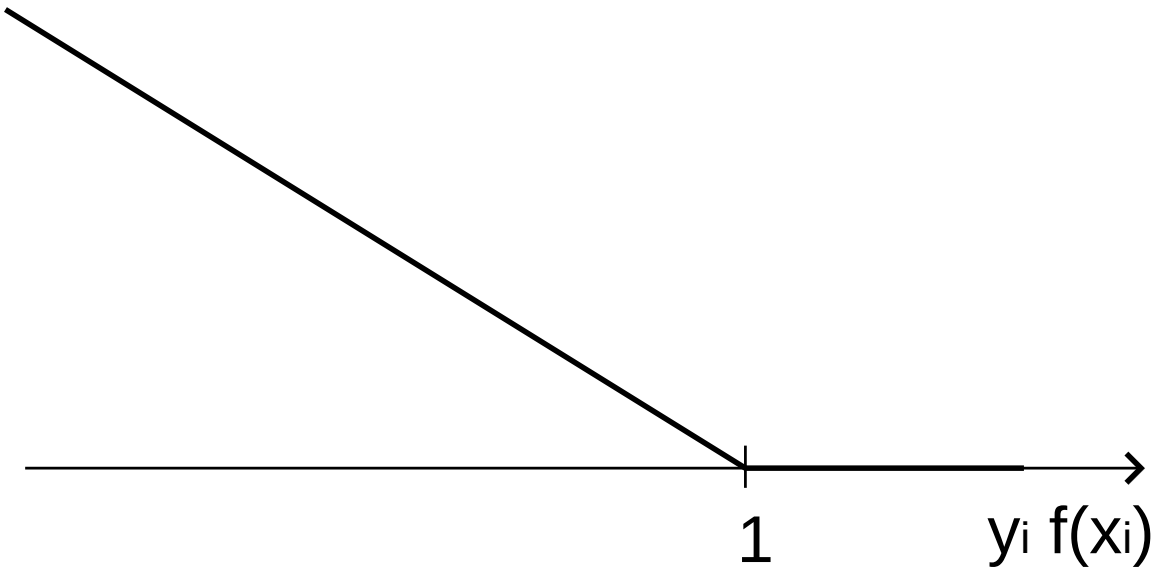
This means that the function to be minimized is $max(0\ ,\ 1 - y_i f(x_i))$ without any contraint.

So, the SVM problem can be written:

$$\min_{w,b} \frac{1}{2}||w||^2 + C \sum_{i=1}^{l} max(0\ ,\ 1 - y_i f(x_i))$$

Geometric interpretation of the loss function

Hinge loss: $\max(0\ ,\ 1 - y_i f(x_i))$



A linearly increasing cost has to be paid for misclassified samples.

Soft margin SVM - semi-supervised case

The label given to $x_i \in \mathcal{D}_u$ is equal to $sign(f(x_i))$. Thus

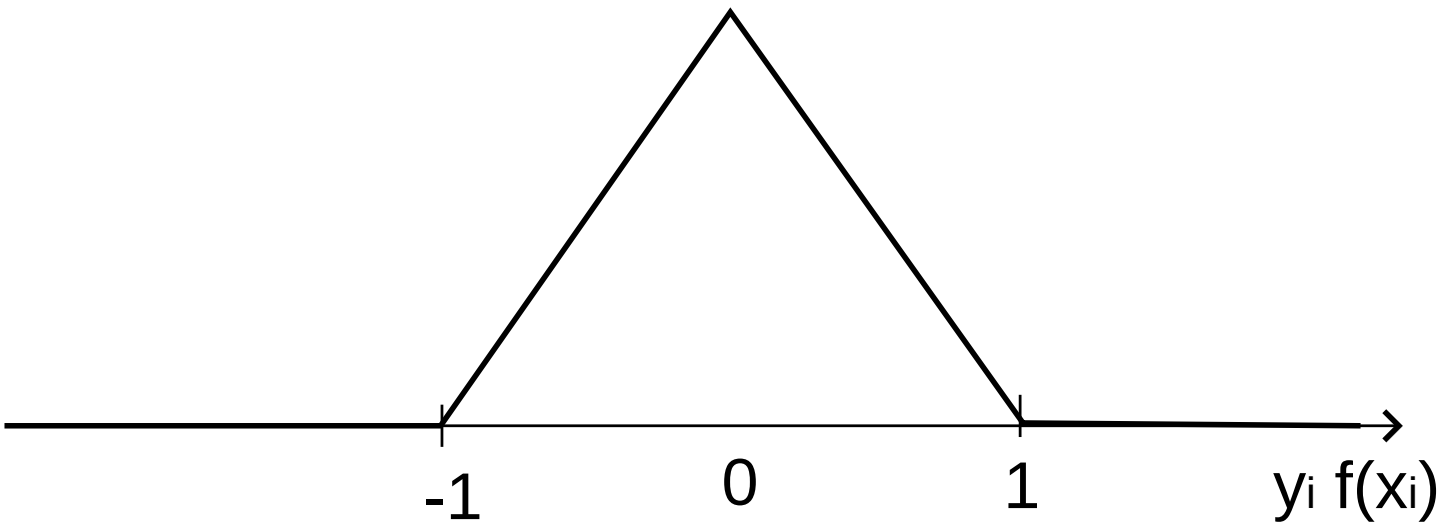
$$y_i f(x_i) = |f(x_i)|$$

The minimization problem involves the hyperplane parameters (w, b) and the label vector Y_u:

$$\min_{w,b,Y_u} \frac{1}{2}||w||^2 + C \sum_{i=1}^{l} max(0\ ,\ 1 - y_i f(x_i)) + C^* \sum_{i=l+1}^{l+m} max(0\ ,\ 1 - |f(x_i)|)$$

Geometric interpretation of the loss function

Function $\max(0\ ,\ 1 - |f(x_i)|)$



A non-zero cost is assigned to the classified samples within the margin.
The cost increases linearly approaching the border.

Soft margin SVM - semi-supervised case

Equivalent formulation with contraints

The minimization problem can be written:

$$\min_{w,b,\xi_i,y_i} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=l+1}^{l+m}\xi_i$$

Subject to contraints

$$y_i\left(< w, x_i > +b\right) \geq 1 - \xi_i \quad \forall i = 1 \ldots l + m$$

$$\xi_i \geq 0 \quad \forall i = 1 \ldots l + m$$

$$y_i \in \{-1, +1\} \quad \forall i = l + 1 \ldots l + m$$

Constraint on the probability of each class

To avoid obtaining solutions with classes whose probabilities are inconsistent with the problem, it is best to set these probabilities. We then set the additional constraint:

$$\frac{1}{m}\sum_{i=l+1}^{l+m} max(0, y_i) = r \quad \text{soit } \frac{1}{m}\sum_{i=l+1}^{l+m} y_i = 2r - 1$$

Indeed

$$\sum_{i=l+1}^{l+m} y_i = \sum_{i=l+1}^{l+m} max(0, y_i) - \sum_{i=l+1}^{l+m} max(0, -y_i) = mr - m(1-r)$$

where r is a priori fixed, usually given by

$$r = \frac{1}{l}\sum_{i=1}^{l} max(0, y_i)$$

Resolution of the optimization problem

The objective function is non-convex, in contrast to supervised. Solving the problem is difficult.
There are two approaches:

1. Combinatorial optimization
2. Continuous optimization

Combinatorial optimization

General Idea For a given solution Y_U solve the problem classically as a standard SVM. Thus

$$\mathcal{J}(Y_u) = \min_{w,b,\xi} I(w, b, Y_u)$$

where

$$I(w, b, Y_u) =$$

$$\min_{w,b,Y_u} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} max(0\ ,\ 1 - y_i f(x_i)) + C^* \sum_{i=l+1}^{l+m} max(0\ ,\ 1 - y_i f(x_i))$$

The number of functions $\mathcal{J}(Y_u)$ is equal to 2^m.
Taking into account the constraint on class probabilities, it is equal to C_{mr}^m.

Combinatorial optimization - Branch and Bound

Branch and Bound principle
Consider a function f to minimize in a space $\mathcal{X}$.

Branching: the region is recursively divided into smaller sub-regions. This leads to a tree. Each node of the tree corresponds to a subregion.

Bounding: Consider two sub-regions A and $B \subseteq \mathcal{X}$. If an upper bound of f on the set A is known and is less than a lower bound of f on the set B, then there is no need to explore the corresponding subtree.

Combinatorial optimization - Branch and Bound

Each node corresponds to a partial labeling unlabeled set:
◦ with all initial labels given by the set of original labels $\mathcal{D}_l$ and a subset S of the initial set without labels $\mathcal{D}_u$
◦ the set of unlabeled data is given by the initial set of unlabeled data $\mathcal{D}_u$ where S has been removed.

Different algorithms have been proposed depending on the method used for:

- Branching: how to choose the two sub-regions of a region?
- Bounding: what upper and lower bounds can we use?
- Exploration: in which the order should we examine the tree?

In general, the algorithms allow to obtain the optimal solution on small data sets, but on large sets these algorithms cost too much in computing time.

Combinatorial optimization - S3VM light

Method proposed by Joachims (1999)

Algorithm

- Learn SVM with labeled samples
- Assign class label 1 to unlabeled samples with the largest $f(x_i)$ and respect the constraint on class probability. Assign -1 to all the other samples.
- calculate $\widetilde{C} = 10^{-5} C^*$
- Repeat while $\widetilde{C} < C^*$
  Repeat
  - Minimize $I(w, v, Y_U)$ with the y_i set and $\widetilde{C}$ instead of C^*
  - Exchange interchangeable pairs (i, j), that is to say, the pairs that satisfy $y_i = 1, y_j = -1$ and

$$max(0\ ,\ 1 - f(x_i)) + max(0\ ,\ 1 + f(x_j)) > max(0\ ,\ 1 + f(x_i)) + max(0\ ,\ 1 - f(x_j))$$

  until no pair is interchangeable.

  $\widetilde{C} \leftarrow \min(1, 5\widetilde{C}, C^*)$

Combinatorial optimization - S3VM light

The algorithm then alternates between re-labeling steps and learning. The loop starts with a small value for C^* that gradually increases to reach the value C^*. As C^* controls the non-convex part of the function, this loop can be interpreted as a way to protect the search to choose a sub-optimal local minimum.

Combinatorial optimization - Deterministic annealing (DA)

Context of Deterministic Annealing

This is an optimization method used to solve combinatorial problems or non-convex problems.

General principle of Deterministic Annealing

It consist in transforming the discrete variables y_i in real variables p_i which are interpreted as probabilities.

Combinatorial optimization - Deterministic annealing (DA)

The function to be minimized is:

$$I''(w, b, p_u; T) = I'(w, b, p_u) - TH(p_u)$$

where $I'(w, b, p_u) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} max(0\ ,\ 1 - y_i f(x_i))$

$$+C^* \sum_{i=l+1}^{l+m} p_i max(0\ ,\ 1 - f(x_i)) + (1 - p_i)max(0\ ,\ 1 + f(x_i))$$

and $H(p_u)$ is an entropy term:

$$H(p_i) = -\sum_{i=l+1}^{l+m} p_i \log p_i + (1 - p_i)\log(1 - p_i)$$

The constraint for the probabilities of classes is given by:

$$\frac{1}{m}\sum_{i=l+1}^{l+m} p_i = r$$

Combinatorial optimization - Deterministic annealing (DA)

When $T = 0$ the optimal vector p_u give the optimal vector Y_u with $y_i = sign(< w, x_i > +b)$.
when $T \to \infty$, I'' is dominated by the entropy term, giving the solution with the maximum entropy, that is to say $p_i = r \; \forall i = l + 1 \ldots l + m$.

T parameterize a class of functions to minimize with an increasing degree of non-convexity.

For a given value of T, the optimal w_T, b_T, p_{uT} that minimizes $I''(w, b, p_u; T)$ can be obtained by using an alternating minimization method, or a gradient method.

Combinatorial optimization - Deterministic annealing (DA)

Algorithm

1. Initialize $p_i = r \forall i = l+1 \ldots l+m$
2. Compute $T = 10C^*$, $R = 1.5$, $\varepsilon = 10^{-6}$
3. While $H(p_{uT}) > \varepsilon$ repeat
   1. Solve, starting from the previous solution
      $w_T, b_T, p_{uT} = argmin_{w,b,p_u} I''(w, b, p_u; T)$
      with the constraint $\frac{1}{m} \sum_{i=l+1}^{l+m} p_i = r$
   2. $T \leftarrow T/R$
4. The solution is given by w_T, b_T

Continuous optimization

Principle :
The labels of unlabeled examples are not considered as variables to optimize.

The methods involve solving a modified version of:

$$\min_{w,b,Y_u} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} max(0\ ,\ 1 - y_i f(x_i)) + C^* \sum_{i=l+1}^{l+m} max(0\ ,\ 1 - |f(x_i)|)$$

Continuous optimization

For these methods, you should overcome the constraint on the probability of classes.
The constraint is approximated by:

$$\frac{1}{m}\sum_{i=l+1}^{l+m} < w, x_i > +b = 2r - 1$$

with $r = \frac{1}{l}\sum_{i=1}^{l} y_i$.
So that the constraint is satisfied simply center the unlabeled data to obtain $\sum_{i=l+1}^{l+m} x_i = 0$ and to set $b = 2r - 1$.
This leads back to a problem without constraint.

Continuous optimization - ∇ S3VM

Method proposed by Chapelle and all (2005)
The function $max(0\ ,\ 1-|t|)$ not differentiable is replaced by the differentiable function $\exp(-st^2)$ with $s=5$.
The problem then is to determine w and b which minimizes the function:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{l} max(0\ ,\ 1-y_i\left(< w, x_i > +b\right))$$

$$+C^*\sum_{i=l+1}^{l+m} \exp(-s\left(< w, x_i > +b\right)^2)$$

Continuous optimization - ∇ S3VM

Algorithm

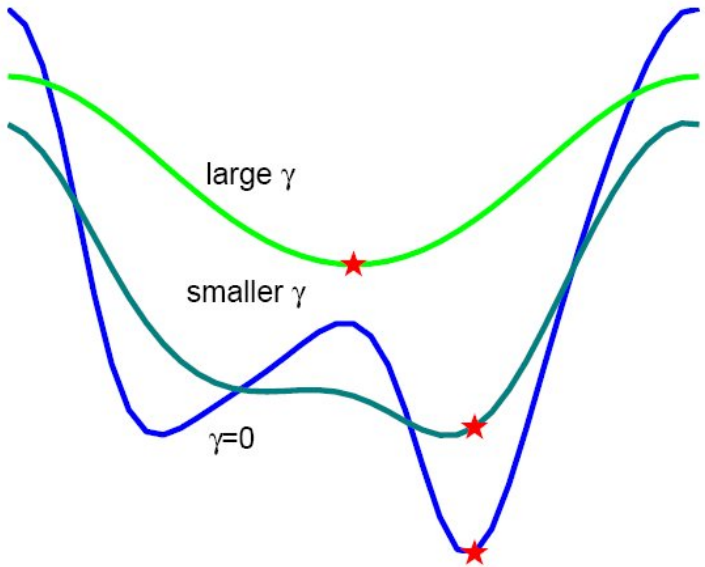
Initialize C^* with a small positive value.
Determine w et b.
Repeat while increasing the value of C^*.

Continuous optimization - Continuation S3VM

General Idea

Global optimization of non-convex objective function

- Convolve the function with a smoothing function, for example a Gaussian function
- Determine the global optimum of the smoothed function with a fairly large smoothing
- Reduce progressively the smoothing
- Determine the optimum from the point obtained at the previous step
- Repeat and stop when there is more smoothing



Continuous optimization - Continuation S3VM

General principle of the method "continuation" for a function f

Choose x_0 and $\gamma_0 > \gamma_1 > \ldots > \gamma_{p-1} > \gamma_p = 0$

At each step i, for $i = 0 \ldots p$,
beginning at point x_i, determine x_{i+1} the minimum of f_{γ_i} :

$$f_\gamma(x) = (\pi\gamma)^{-d/2} \int f(x-t) \exp(- \parallel t \parallel^2 / \gamma) dt$$