Semi-Supervised Learning on Graphs

Presented by Chunping Wang

April 19th, 2007

Outlines

- Introduction
- Gaussian Random Fields (GRF)
- Harmonic Energy Minimization (Zhu et. al)
- Logistic GRF (Krishnapuram et. al)
- Conclusions

Introduction (1)

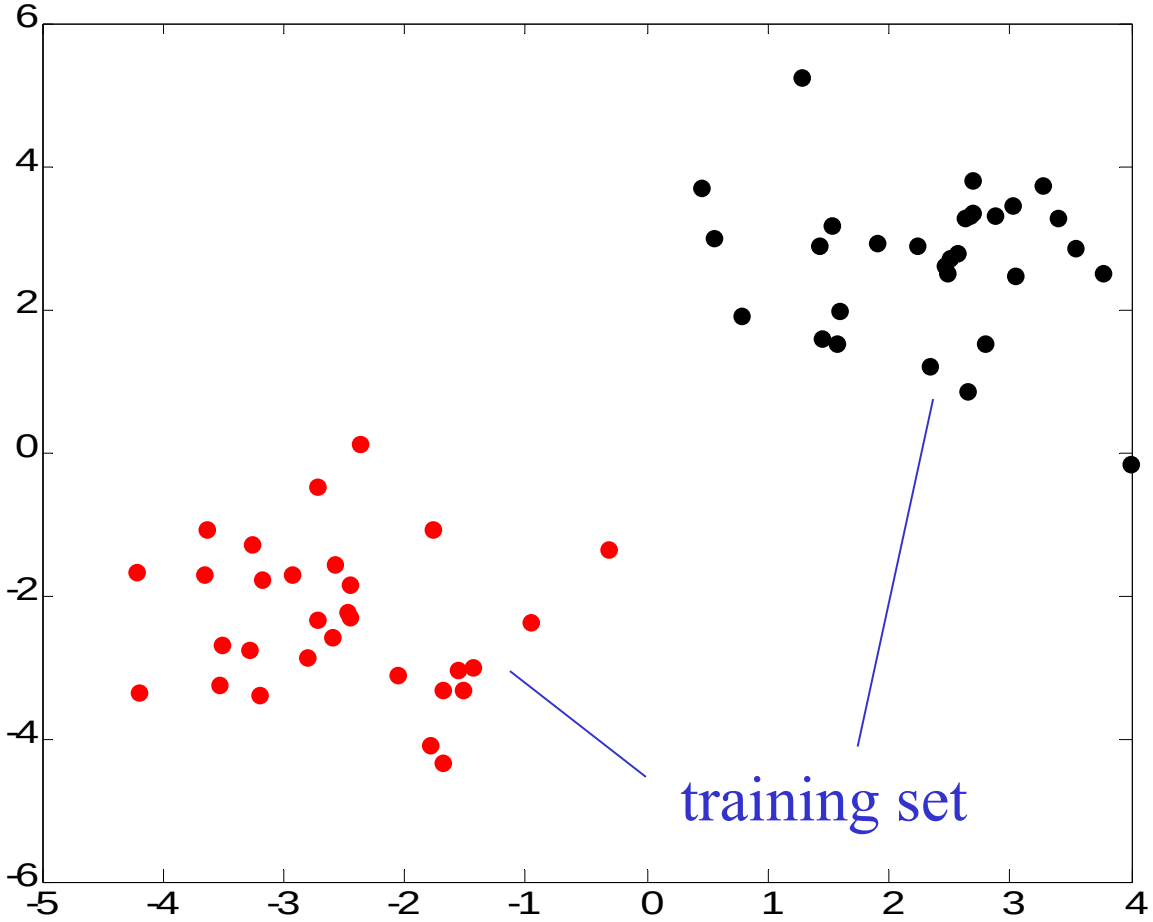
Classification problems $\{x_i, y_i\}_{i=1}^{L}$

2-D predictors

$\{x_i\}_{i=1}^{L}$

binary labels

$\{y_i\}_{i=1}^{L}$



Introduction (1)

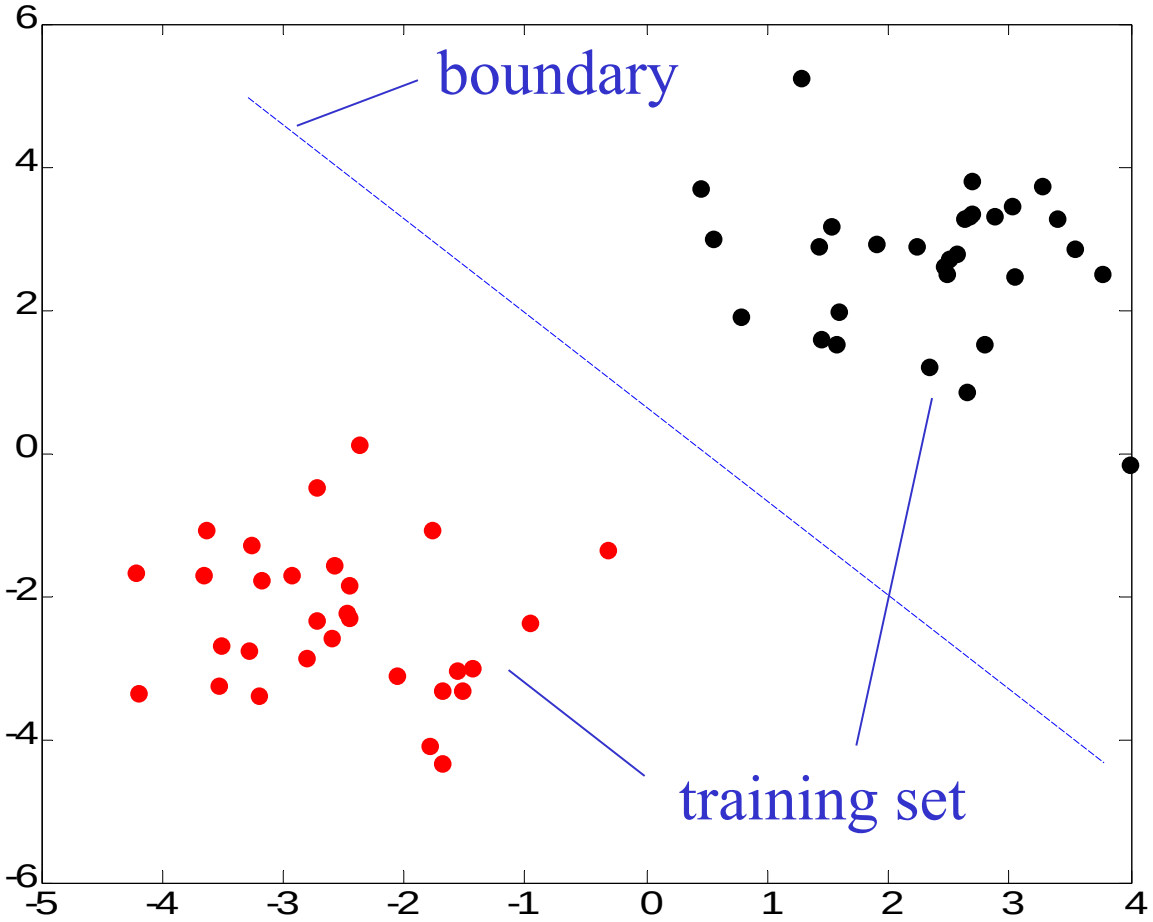
Classification problems $\{x_i, y_i\}_{i=1}^{L} \Rightarrow mapping\ rule$

2-D predictors

$\{x_i\}_{i=1}^{L}$

binary labels

$\{y_i\}_{i=1}^{L}$



Introduction (1)

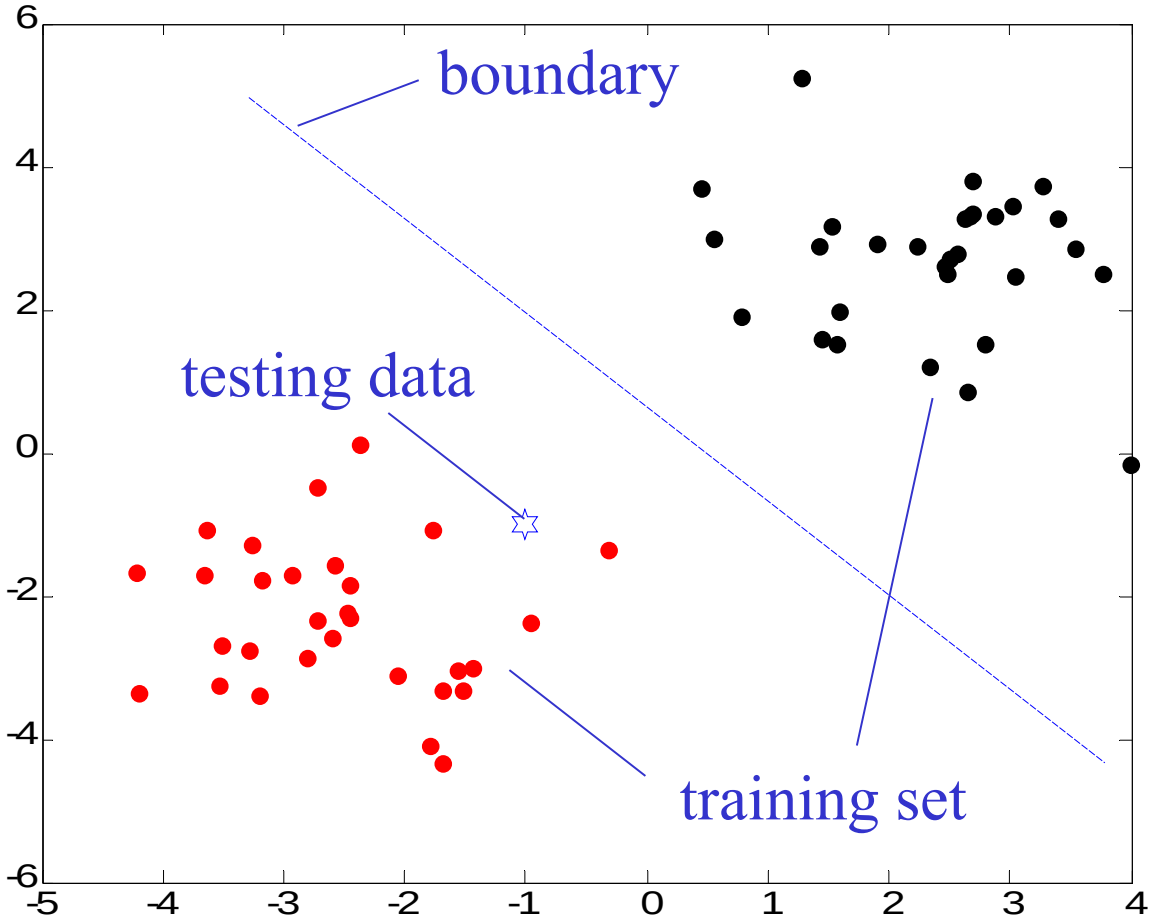
Classification problems $\{x_i, y_i\}_{i=1}^{L} \Rightarrow mapping\ rule \overset{x_i}{\Rightarrow} y_i$

2-D predictors

$\{x_i\}_{i=1}^{L}$

binary labels

$\{y_i\}_{i=1}^{L}$

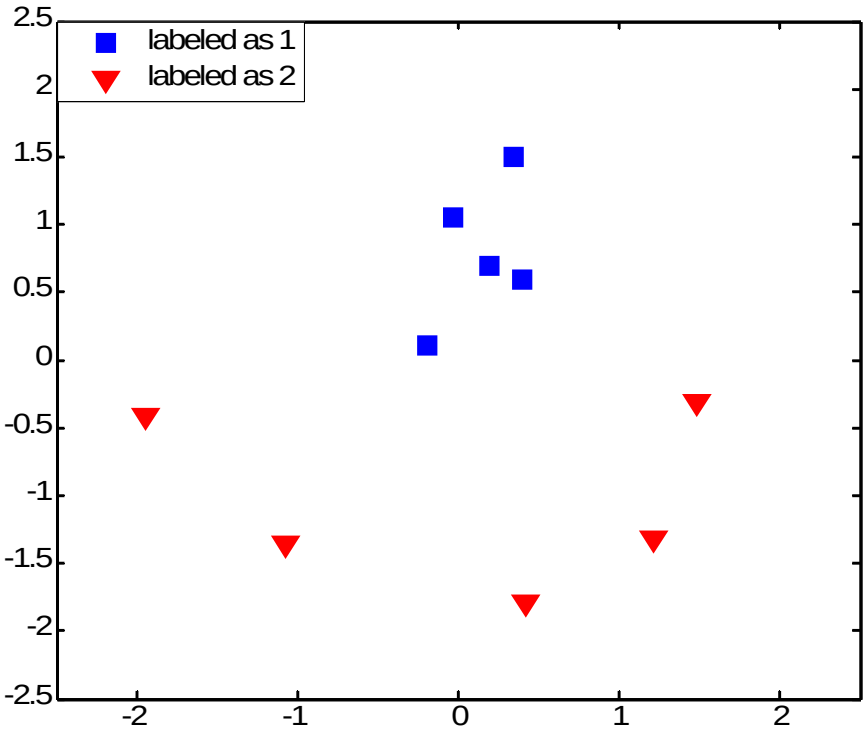


Introduction (2)

Only labeled are used data to learn the decision boundary --- supervised learning.

It is expensive to label data points in many applications, for example:

- protein shape classification
- medical diagnoses
- landmine detection



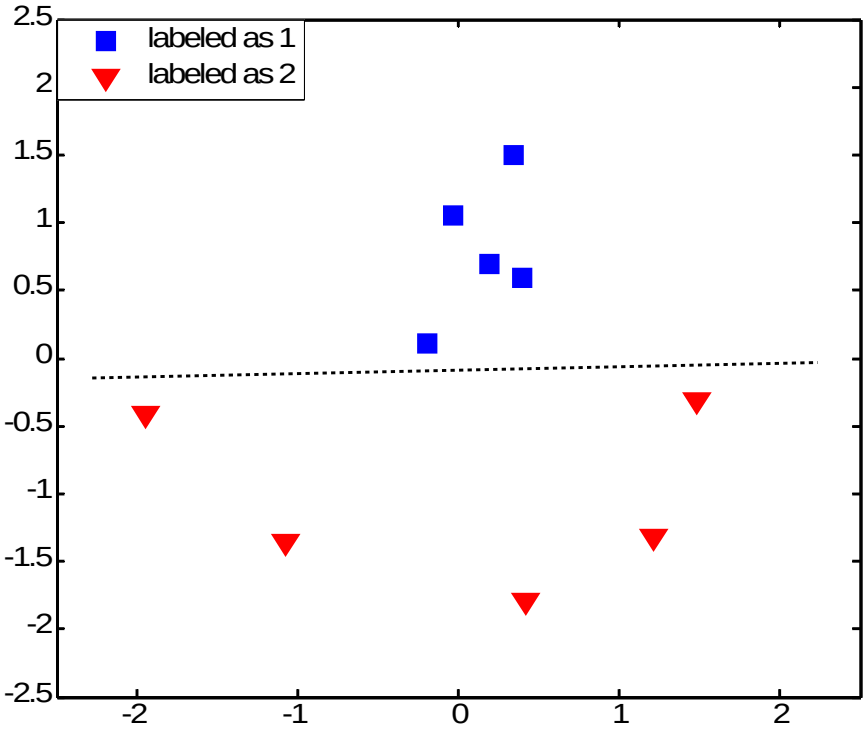
For these cases, labeled data are usually scarce although plenty of unlabeled data may be available.

Introduction (2)

Only labeled are used data to learn the decision boundary --- supervised learning.

It is expensive to label data points in many applications, for example:

- protein shape classification
- medical diagnoses
- landmine detection



For these cases, labeled data are usually scarce although plenty of unlabeled data may be available.

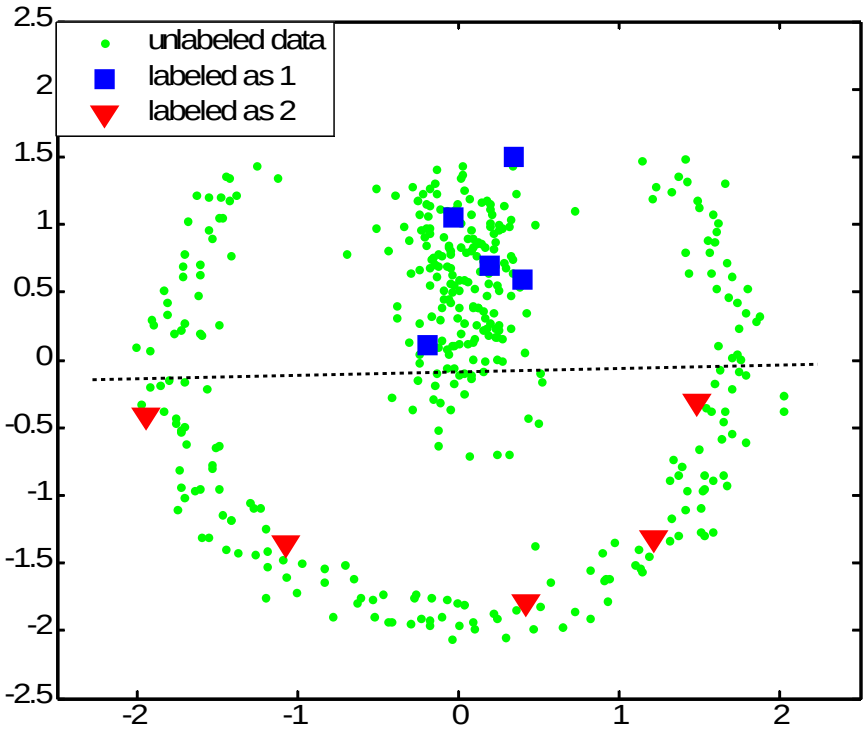
Introduction (2)

Only labeled are used data to learn the decision boundary --- supervised learning.

It is expensive to label data points in many applications, for example:

- protein shape classification
- medical diagnoses
- landmine detection

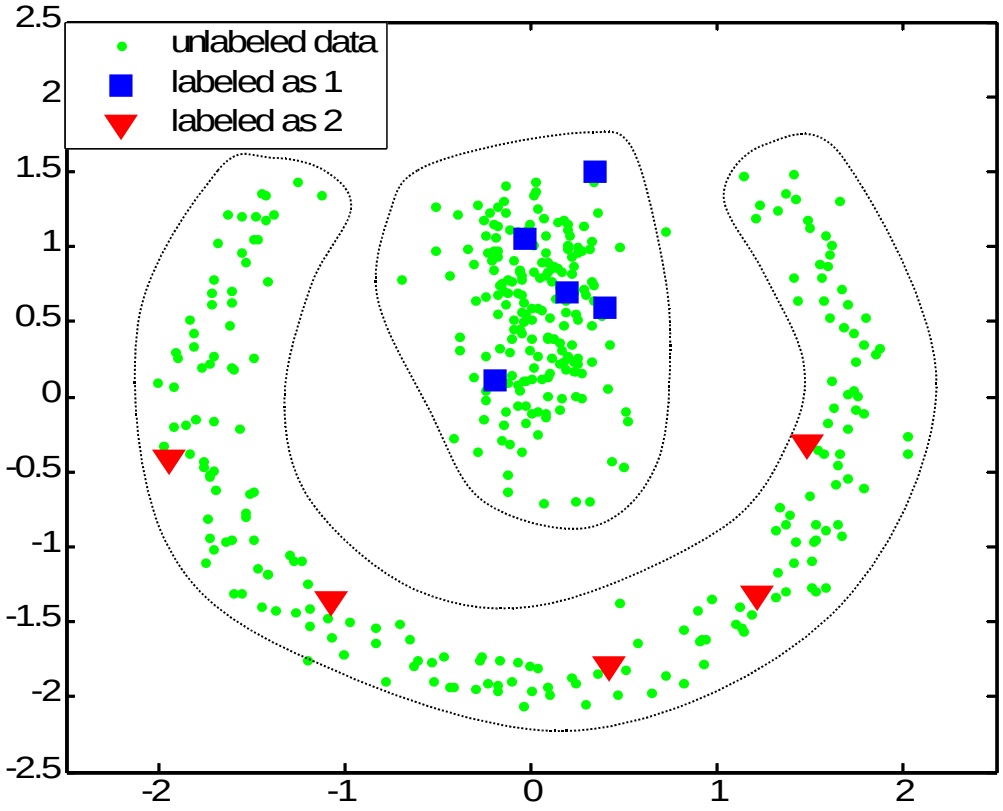
For these cases, labeled data are usually scarce although plenty of unlabeled data may be available.



Over-fitting

Introduction (3)

Semi-supervised learning: use the manifold constructed by both labeled and unlabeled data to learn.



Gaussian Random Fields (1)

Suppose feature vectors (predictors) $x_1, \cdots, x_N$. Consider a connected graph $G = (V, E, W)$ with nodes V corresponding to the N data points and an $N \times N$ symmetric weight matrix W on the edges E, where nearby points are assigned large weights, e.g.,

$$w_{ij} = \exp(-\| x_i - x_j \|^2 / \sigma^2)$$

Define a real-valued function f: $V \to \Re$

To measure how much f varies across the graph, a quadratic energy function is defined by

$$E(f) = f^T \Delta f = \tfrac{1}{2} \sum_{i,j} w_{ij} [f(i) - f(j)]^2 \qquad \Delta = D - W$$

Gaussian random field $\quad p(f) \propto \exp\{-\lambda E(f)\}$

Gaussian Random Fields (2)

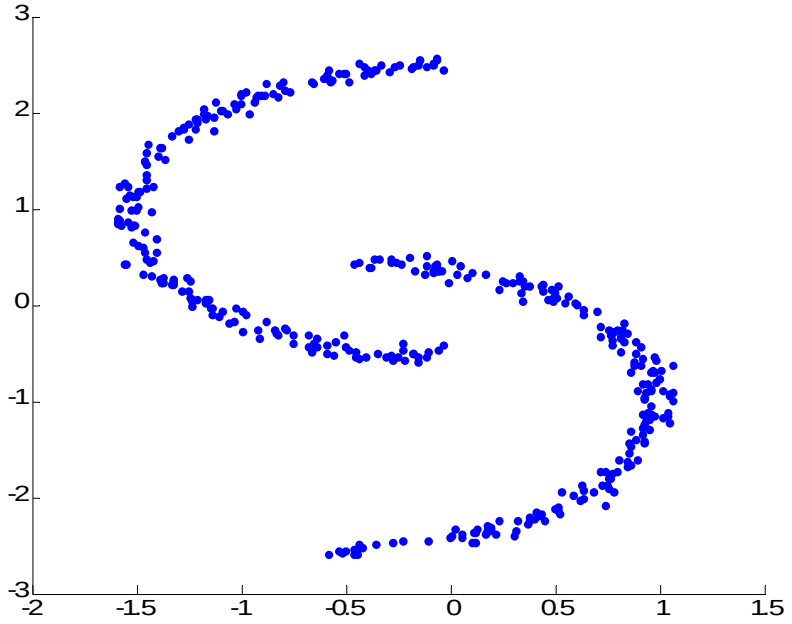
Gaussian random field $p(f) \propto \exp\{-\lambda E(f)\}$

Those functions f with lower energy E(f) have higher probability.

Consider separable cases

We hope the points in the same cluster have similar f(i). For such f,

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}[f(i) - f(j)]^2 = 0$$

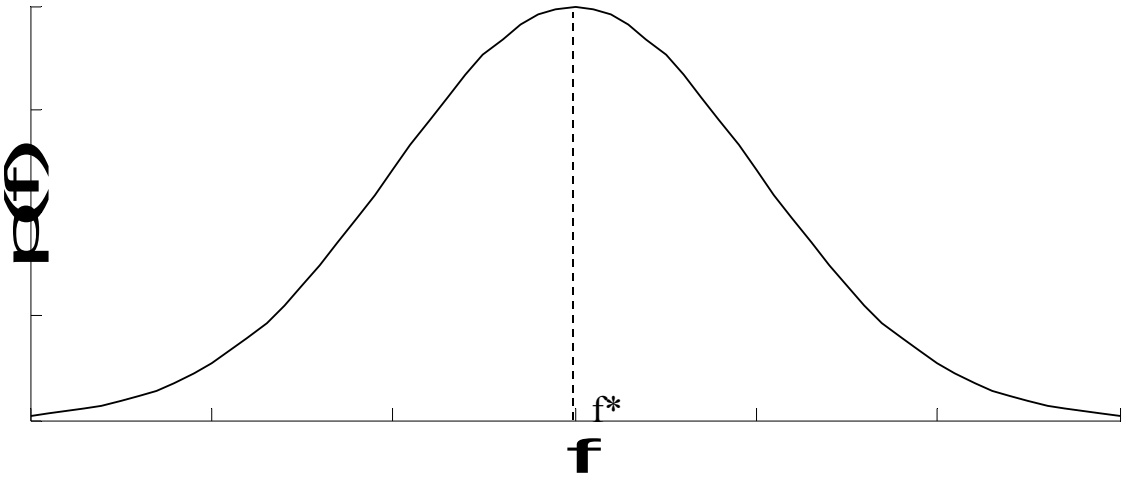


Harmonic Energy Minimization (1)

Assume binary labels $y_i \in \{0,1\}$, $i = 1, \cdots N$, but only first L of them are available, the last N-L are to be learned.

Gaussian random field $p(f) \propto \exp\{-\lambda E(f)\}$ $E(f) = f^T \Delta f$

Consider the most probable f, which is the mean of the GRF with constraint of $f(i) = f_l(i) = y_i$, $i = 1, \cdots L$.



$f^* = \arg\min_{f|f_l} E(f) \Leftrightarrow \Delta f^* = 0$ i.e., f* is harmonic

Harmonic Energy Minimization (2)

Harmonic property results in $(D-W)f^* = 0 \quad (1)$

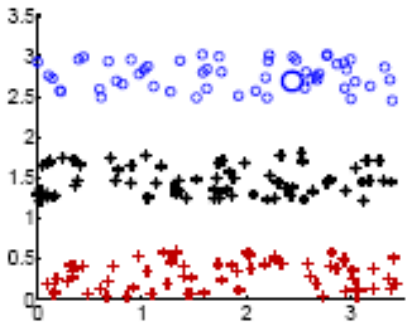
$$f^*(j) = \frac{1}{d_j}\sum_{i\sim j} w_{ij} f^*(i), \; for \; j = L+1, \cdots, N$$

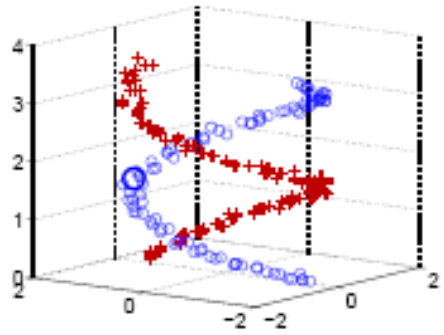
f* is smoothed over local neighborhood on the graph.

The uniqueness principle of harmonic functions guarantees a unique solution of f* for (1).

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}, \quad D = \begin{bmatrix} D_{ll} & 0 \\ 0 & D_{uu} \end{bmatrix}, \quad f^* = \begin{bmatrix} f_l \\ f_u^* \end{bmatrix}$$

$$f_u^* = (D_{uu} - W_{uu})^{-1} W_{ul} f_l$$



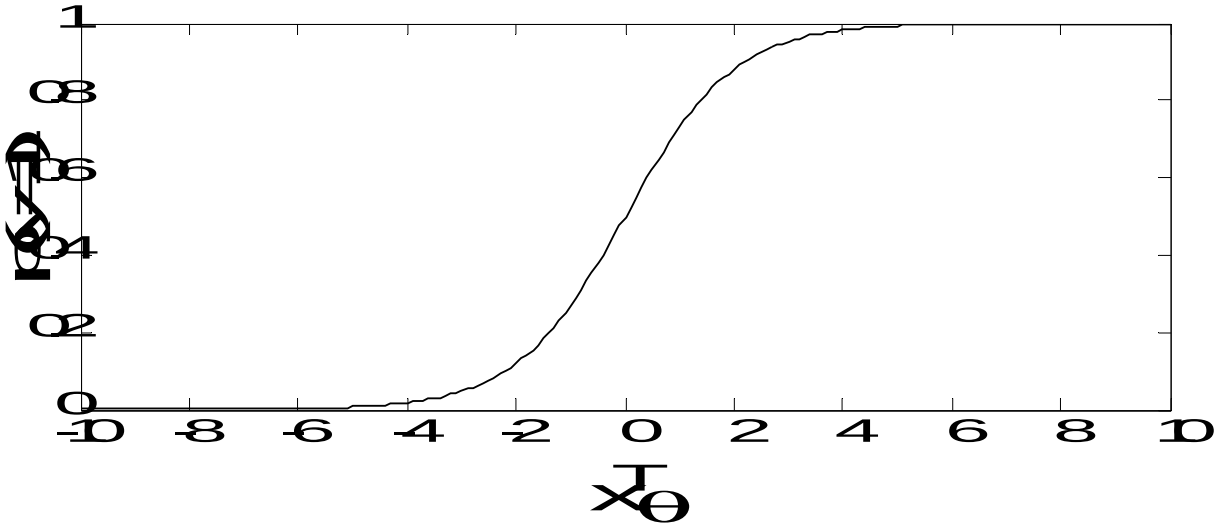


Logistic GRF – parametric (1)

$p(y_i)$ is explicitly related to x_i via a set of parameters θ

Logistic regression (binary case)

$$p(y_i = 1 \mid x_i, \theta) = \frac{1}{1 + \exp(-x_i^T \theta)} \quad (2)$$



$\Re \rightarrow [0, 1]$

Goal: to learn the posterior or point estimate of θ in light of labeled data, so that (2) could be used for prediction.

Logistic GRF – parametric (2)

Bayes theorem

$$\text{posterior } p(\theta \mid \{y_i\}_{i=1}^{L}) = \frac{\overbrace{p(\theta)}^{\text{prior}} \overbrace{\prod_{i=1}^{L} p(y_i \mid x_i, \theta)}^{\text{likelihood}}}{\underbrace{p(\{y_i\}_{i=1}^{L})}_{\text{evidence a constant independent of } \theta}} \propto p(\theta)\prod_{i=1}^{L} p(y_i \mid x_i, \theta)$$

Define f as a linear function of X

$$f(i) = f(x_i) = \theta^T x_i, \quad f = X\theta$$

The Gaussian random field induces a Gaussian prior on θ

$$p(f) \propto \exp\{-\lambda f^T \Delta f\} \Leftrightarrow p(\theta) \propto \exp\{-\lambda \theta^T X^T \Delta X \theta\}$$

Point estimate – maximize a posterior

Logistic GRF – parametric (3)

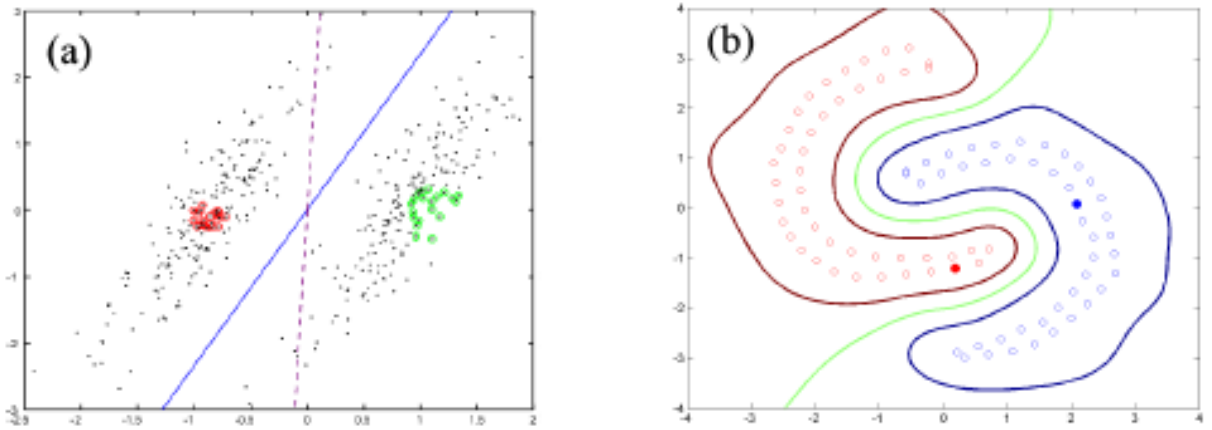


Figure 1: Synthetic two-dimensional examples. (a) Comparison of the supervised logistic linear classifier (boundary shown as dashed line) learned only from the labelled data (shown in color) with the proposed semi-supervised classifier (boundary shown as solid line) which also uses the unlabelled samples (shown as dots). (b) A RBF kernel classifier obtained by our algorithm, using two labelled samples (shaded circles) and many unlabelled samples.

Conclusions

✓ Resent semi-supervised work is usually based on graphs constructed by both labeled and unlabeled data;

✓ These two approaches we discussed are both based on a Gaussian random field. The first method concentrates on the use of only the mean of the field; however, in the second algorithm, the whole field is taken as the prior knowledge.

✓ In the aspect of the usage of the known labels, the first approach takes them as constraints while the second method regards them as the outputs of a probabilistic model.

✓ The first method is flexible for any decision boundary, whereas a kernel has to be introduced into the parametric way for a nonlinear boundary.

✓ After training phase, testing on new data points is straightforward for the parametric way.