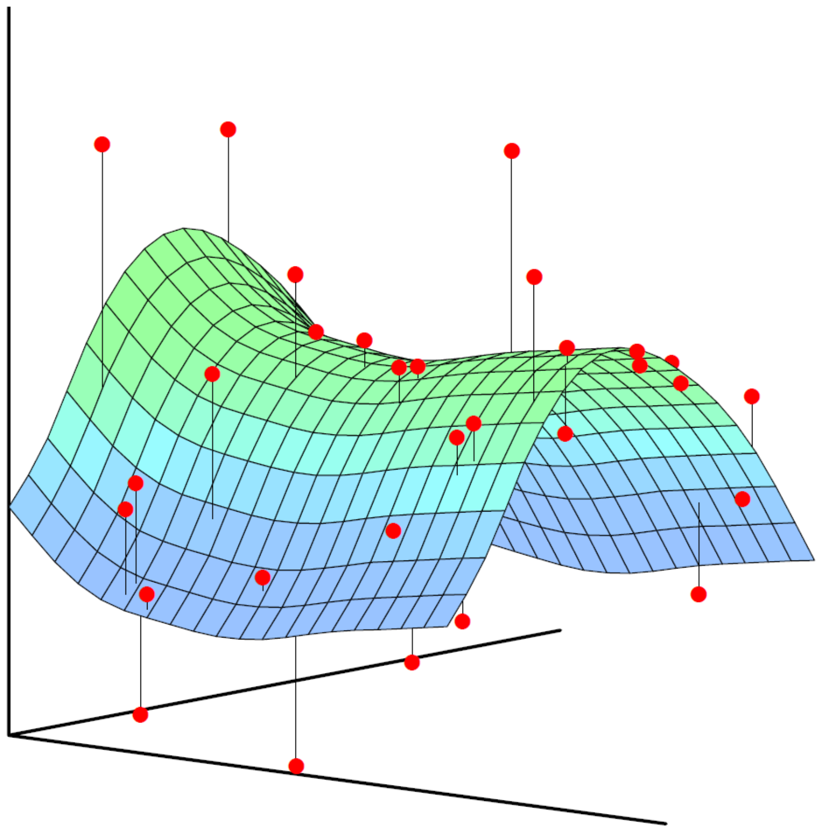
Machine Learning



第10讲 半监督学习

Semi-supervised learning

刘 峤

电子科技大学计算机科学与工程学院

10.1 Introduction to SSL

(Semi-supervised learning)

Semi-supervised learning (SSL)

- Traditional supervised learning is limited to using labeled data.
- SSL also uses unlabeled data to learn.
- Let (x,y) be a labeled instance and (x,ø) be an unlabeled instance.
  - ※ L: a set of ***n*** labaled instances.
  - ※ U: a set of ***m*** unlabeled instances.
  - ※ $n << m$
- SSL tries to use $L \cup U$ to learn a predictive model.

Semi-supervised learning (SSL)

- Suitable when just a small proportion of the training data is labeled.
- These algorithms try to learn also from the unlabeled data.

Labeled data


cat


dog

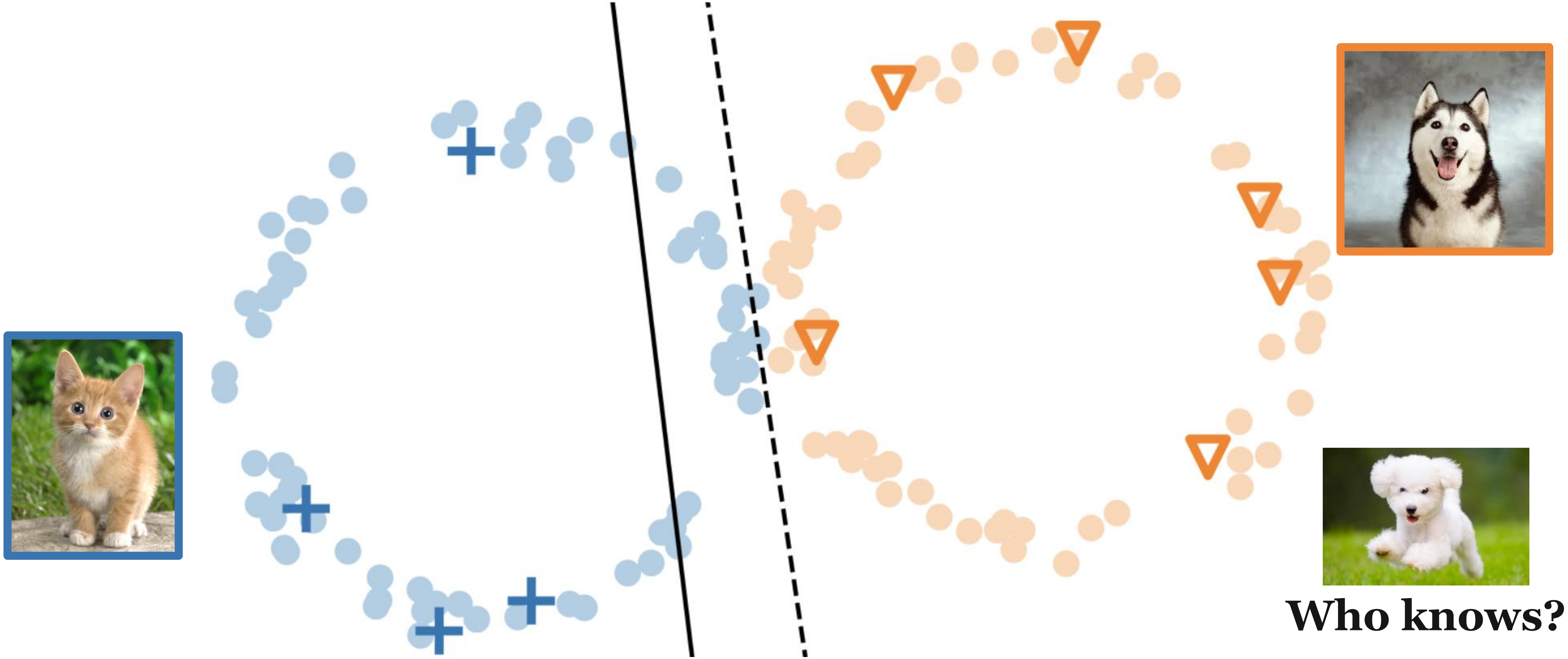
Unlabeled data

(Image of cats and dogs without labeling)

labeled

unlabaled

Why semi-supervised learning helps?



The distribution of the unlabeled data tell us something.

Usually with some assumptions

Semi-supervised learning (SSL)

- Semi-supervised learning:

  ※ A set of unlabeled data, usually U >> L(labeled)

  ※ Transductive learning: unlabeled data is the testing data

  ※ Inductive learning: unlabeled data is not the testing data

- Why semi-supervised learning?

  ※ Collecting data is easy, but collecting “labelled” data is expensive

  ※ We do semi-supervised learning in our lives

10.2 Semi-supervised Learning for Generative Model

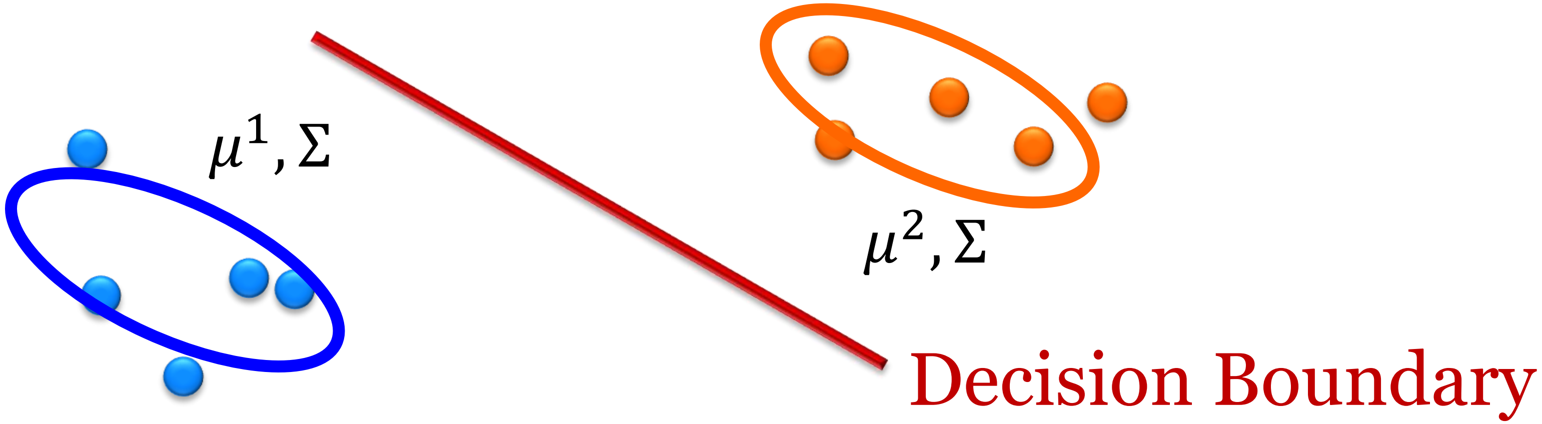
Supervised Generative Model

- Given labelled training examples $x^r \in \{C_1, C_2\}$

※ looking for most likely prior probability $P(C_i)$ and

※ class-dependent probability $P(X|C_i)$

※ $P(X|C_i)$ is a Gaussian parameterized by μ^i and Σ

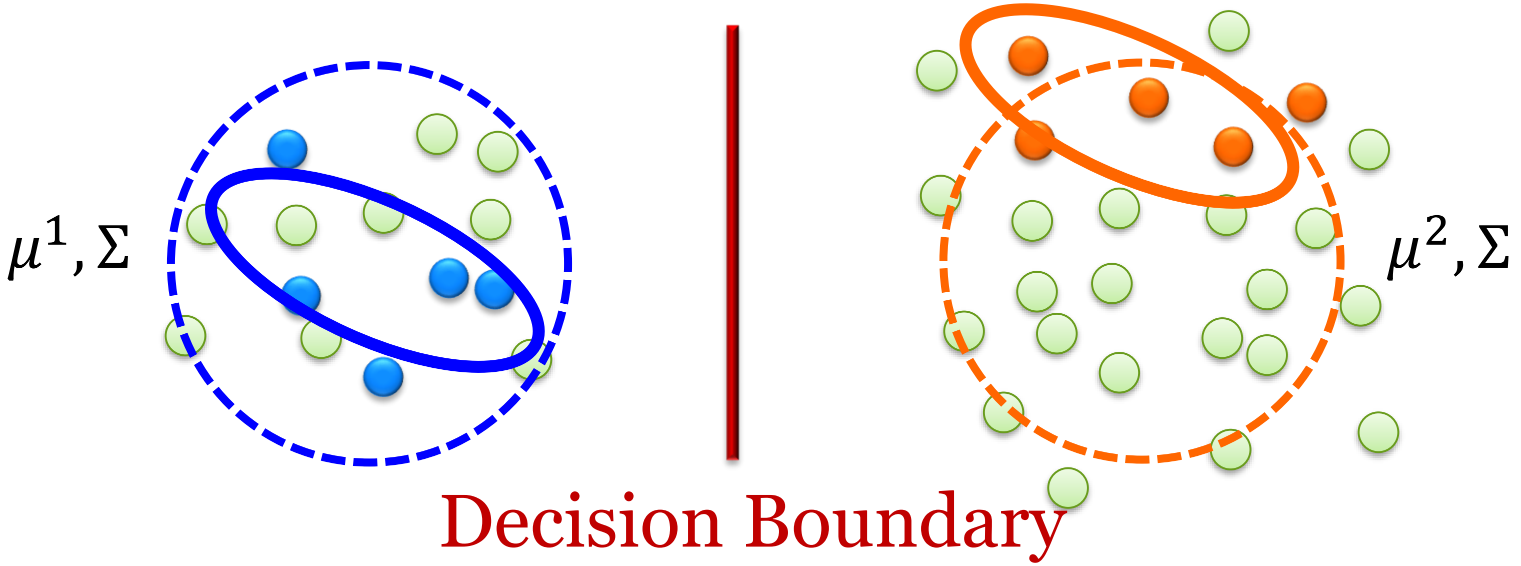


※ With $P(C_1)$, $P(C_2)$, μ^1, μ^2, Σ

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1)+P(x|C_2)P(C_2)}$$

Supervised Generative Model

- Given labelled training examples $x^r \in \{C_1, C_2\}$
  - ※ looking for most likely prior probability $P(C_i)$ and
  - ※ class-dependent probability $P(X|C_i)$
  - ※ $P(X|C_i)$ is a Gaussian parameterized by μ^i and Σ



※ The unlabeled data x^u help re-estimate $P(C_1)$, $P(C_2)$, μ^1, μ^2, Σ

Semi-supervised Generative Model

- Initialization: $\theta = \{P(C_1),\ P(C_2),\ \mu^1,\ \mu^2, \Sigma\}$
- Step 1: compute the posterior probability of unlabeled data $P_\theta(C_1|x^u)$

  ※ Depending on model θ

- Step 2: update model

  - N: total number of examples
  - N_1: number of examples belonging to C_1

$$P(C_1) = \frac{N_1 + \sum_{x^u} P(C_1|x^u)}{N}$$

$$\mu^1 = \frac{1}{N_1} \sum_{x^r \in C_1} x^r + \frac{1}{\sum_{x^u} P(C_1|x^u)} \sum_{x^u} P(C_1|x^u) x^u$$

- Back to step 1, until the algorithm converges

The algorithm converges eventually, but the initialization influences the results.

Why?

$$\theta = \{P(C_1),\ P(C_2),\ \mu^1,\ \mu^2, \Sigma\}$$

- Maximum likelihood with labelled data closed-form solution

$$\log L(\theta) = \sum_{x^r} \log P_\theta(x^r, \hat{y}^r) \qquad P_\theta(x^r, \hat{y}^r) = P_\theta(x^r|\hat{y}^r)P(\hat{y}^r)$$

- Maximum likelihood with labelled + unlabeled data Solved iteratively

$$\log L(\theta) = \sum_{x^r} \log P_\theta(x^r, \hat{y}^r) + \sum_{x^u} \log P_\theta(x^u)$$

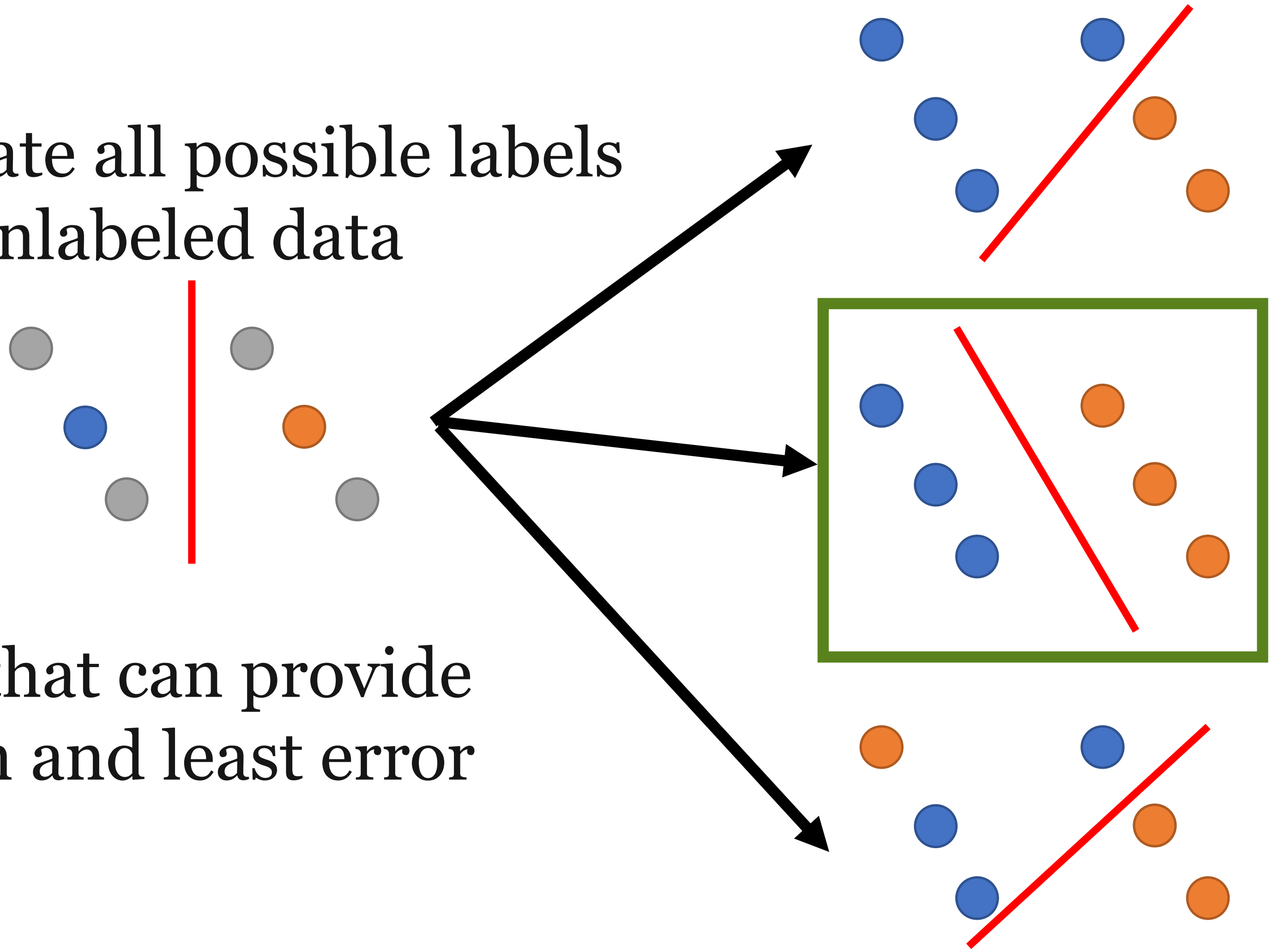
$$P_\theta(x^u) = P_\theta(x^u|C_1)P(C_1) + P_\theta(x^u|C_2)P(C_2)$$

(x^u can come from either C_1 and C_2)

10.3 Low-density Separation Assumption

Outlook: Semi-supervised SVM

Enumerate all possible labels for the unlabeled data



Find a boundary that can provide the largest margin and least error

Thorsten Joachims, ”Transductive Inference for Text Classification using Support Vector Machines”, ICML, 1999

Self-training

Self-training

- Given:
  - ※ labelled data set = $\{(x^r, \hat{y}^r)\}_{r=1}^{R}$
  - ※ unlabeled data set = $\{x^u\}_{u=1}^{U}$
- Repeat:
  - ※ Train model f^* from labelled data set

    You can use any model here.

    Regression?
  - ※ Apply f^* to the unlabeled data set
    - ➢ Obtain $\{(x^u, y^u)\}_{u=1}^{U}$

      Pseudo-label

      You can also provide a weight to each data.
  - ※ Remove a set of data from unlabeled data set
    - ➢ and add them into the labeled data set

      How to choose the data set remains open

Self-training

- Similar to semi-supervised learning for generative model

  ※ **Hard** label v.s. **Soft** label

- Considering using neural network

  ※ θ^* (network parameter) from labelled data

$x^u \rightarrow \boxed{\theta^*} \rightarrow \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$

Hard

Class 1 $\Rightarrow$ New target for x^u is $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

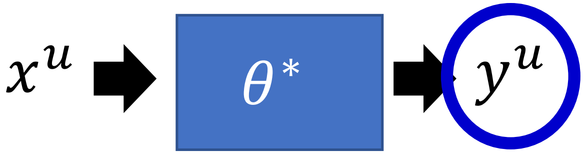
It looks like class 1, then it is class 1.

Soft

70% Class 1
30% Class 2 $\Rightarrow$ New target for x^u is $\begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$

Doesn't work …

Entropy-based Regularization

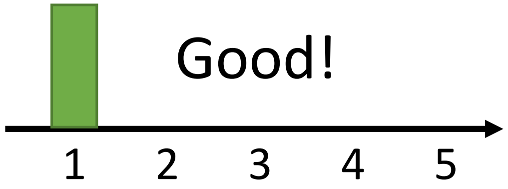


Distribution

Entropy of y^u

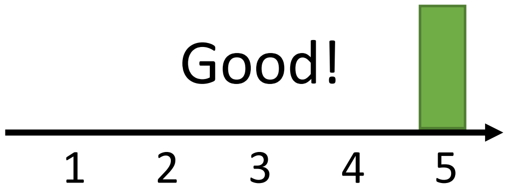
Evaluate how concentrate the distribution y^u is.

y^u



$$E(y^u) = 0$$

y^u



$$E(y^u) = 0$$

y^u

$$E(y^u) = -ln\left(\frac{1}{5}\right) = ln5$$

$$E(y^u) = -\sum_{m=1}^{5} y_m^u ln(y_m^u)$$

Want: as small as possible

$$L = \sum_{x^r} C(y^r, \hat{y}^r) + \lambda \sum_{x^u} E(y^u)$$

labelled data **unlabeled data**

Self-learning ‘improvements’

- Just add instances with the most confident predictions.
- Perform the procedure with batches of instances

  ※ instead of one instance at a time.
- Re-assess previous predictions.

- SSL is not always guaranteed to work!

  ※ Performance may also degrade due to noisy instances.
- Remember: garbage in, garbage out!

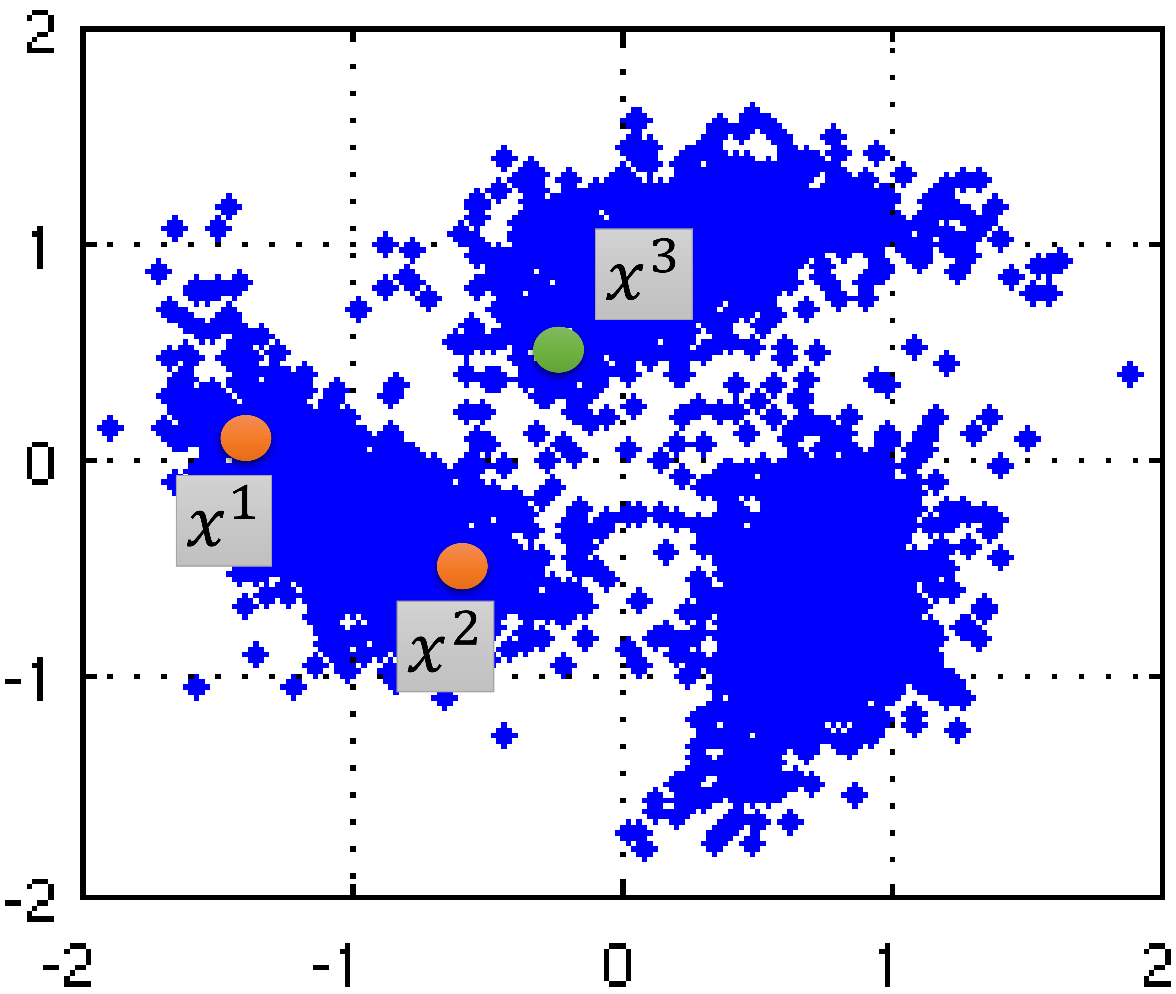
10.4 Smoothness Assumption

Smoothness Assumption

- Assumption: "similar" x has the same $\hat{y}$
- More precisely:
  - ※ x is not uniform.
  - ※ If x^1 and x^2 are close in
  - ※ a high density region,
  - ※ $\hat{y}^1$ and $\hat{y}^2$ are the same.

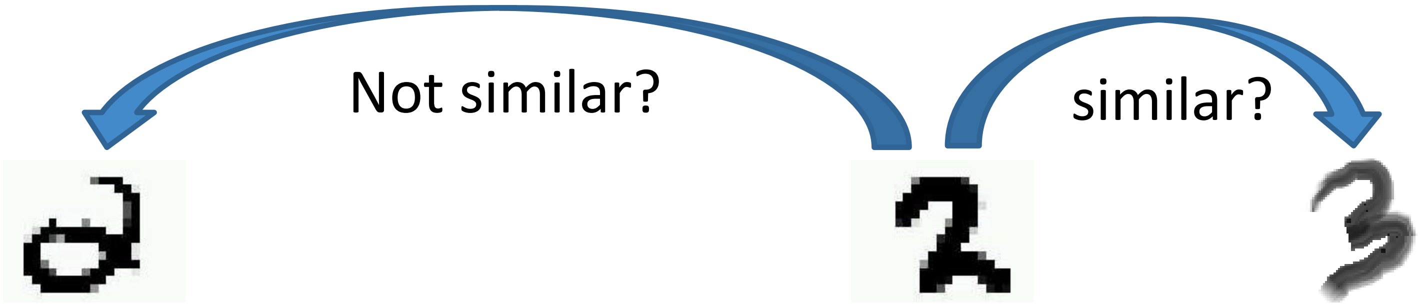
connected by a high density path

- x^1 and x^2 have the same label
- x^2 and x^3 have different labels



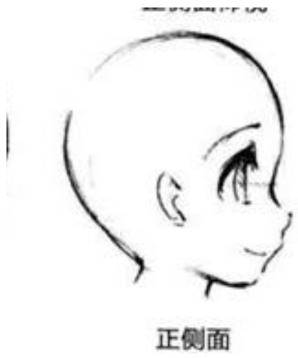
Source of image: http://hips.seas.harvard.edu/files/pinwheel.png

Smoothness Assumption



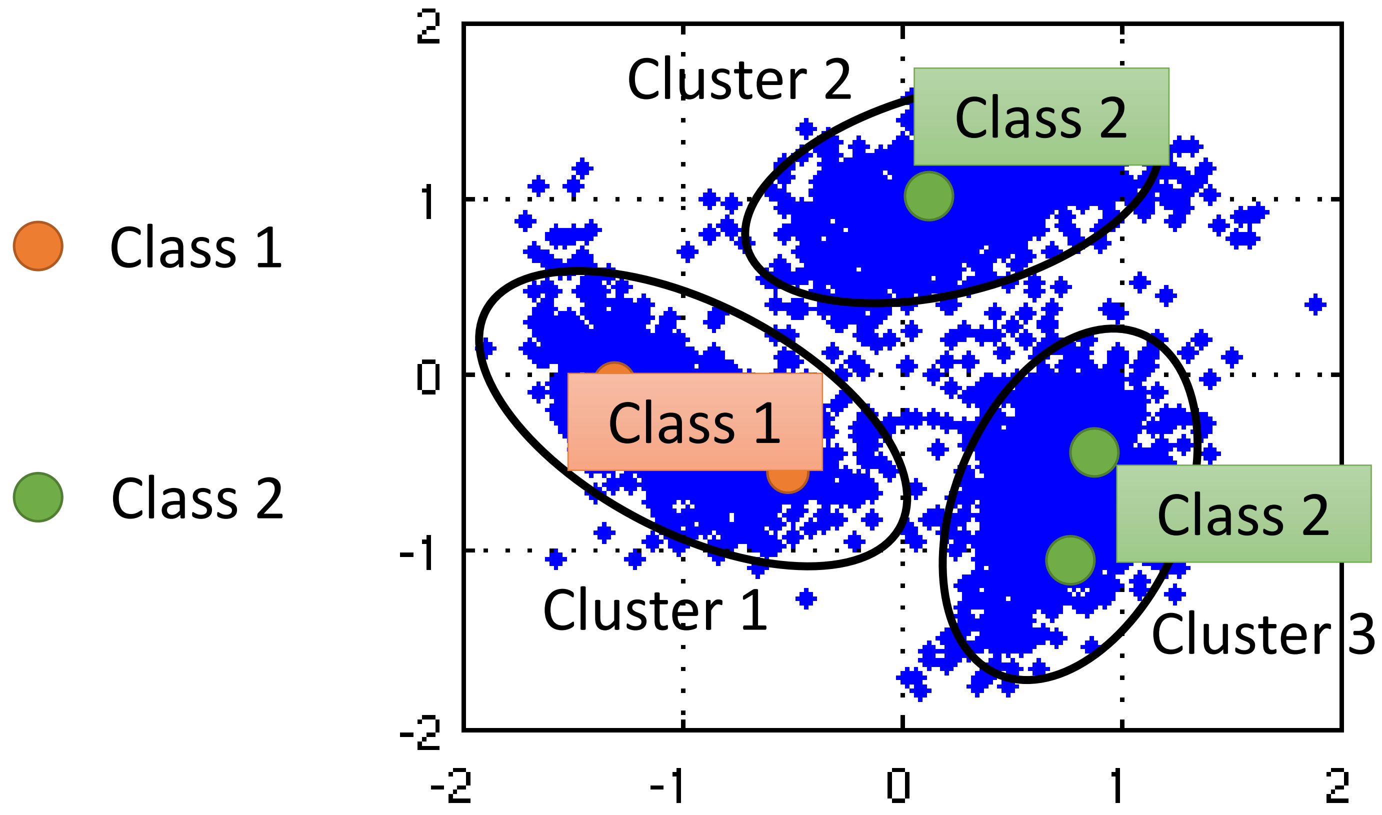
"indirectly" similar with stepping stones

(The example is from the tutorial slides of Xiaojin Zhu.)



Source of image: http://www.moehui.com/5833.html/5/

Cluster and then Label

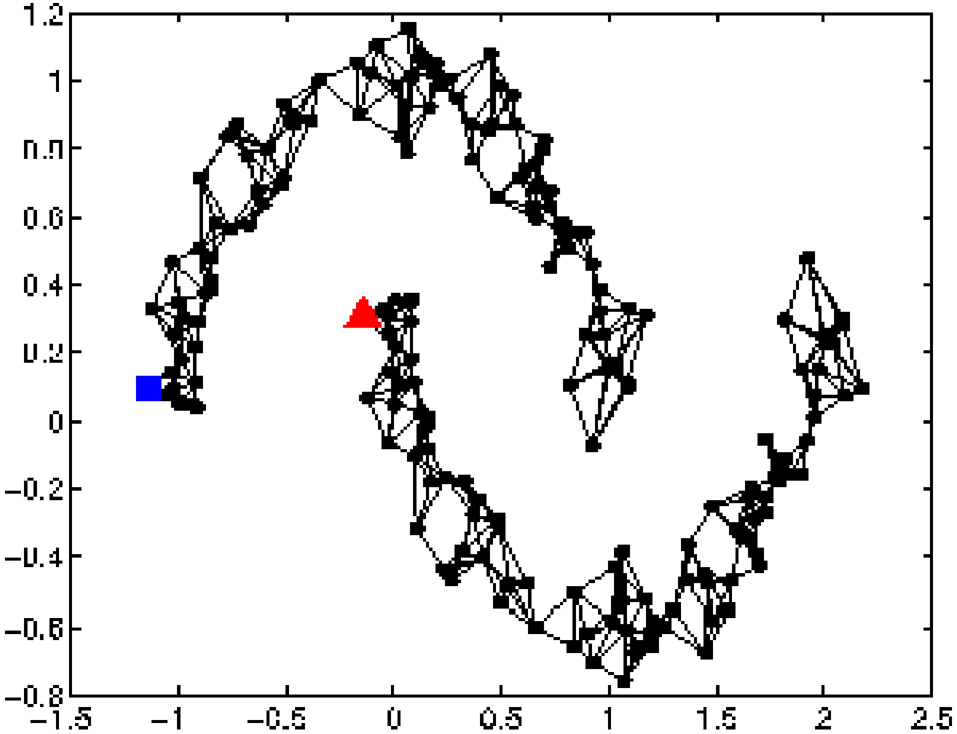


Using all the data to learn a classifier as usual

Graph-based Approach

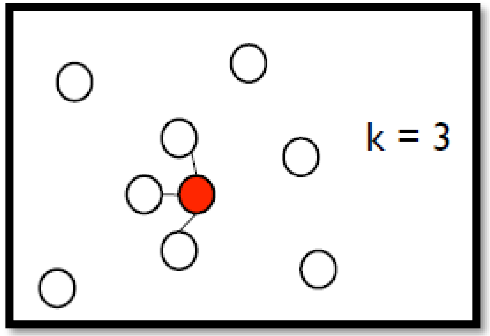
- How to know x^1 and x^2 are close in a high density region?
  ※ connected by a high density path
- Represented the data points as a graph
  ※ Graph representation is nature sometimes.
  - ➢ E.g. Hyperlink of webpages,
  - ➢ E.g. citation of papers

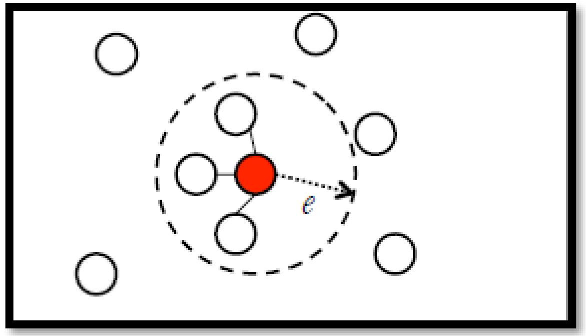
  ※ Sometimes you have to construct the graph yourself.



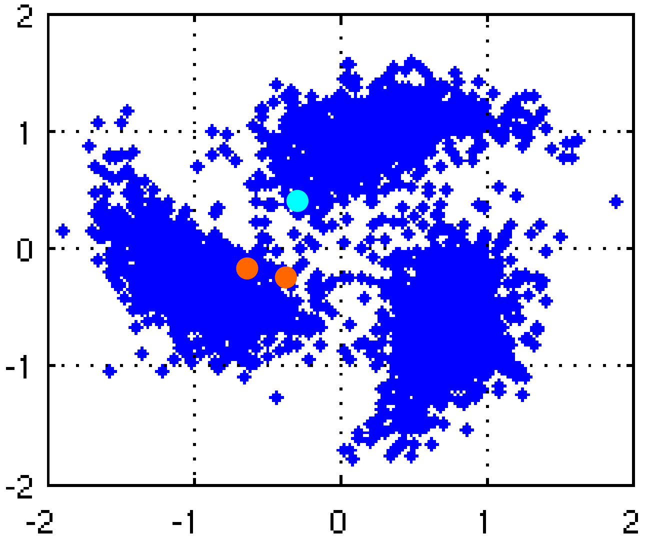
Graph-based Approach: Graph Construction

- Define the similarity $s(x^i, x^j)$ between x^i and x^j
- Add edge:
  - ※ K-Nearest Neighbor
  - ※ e-Neighborhood



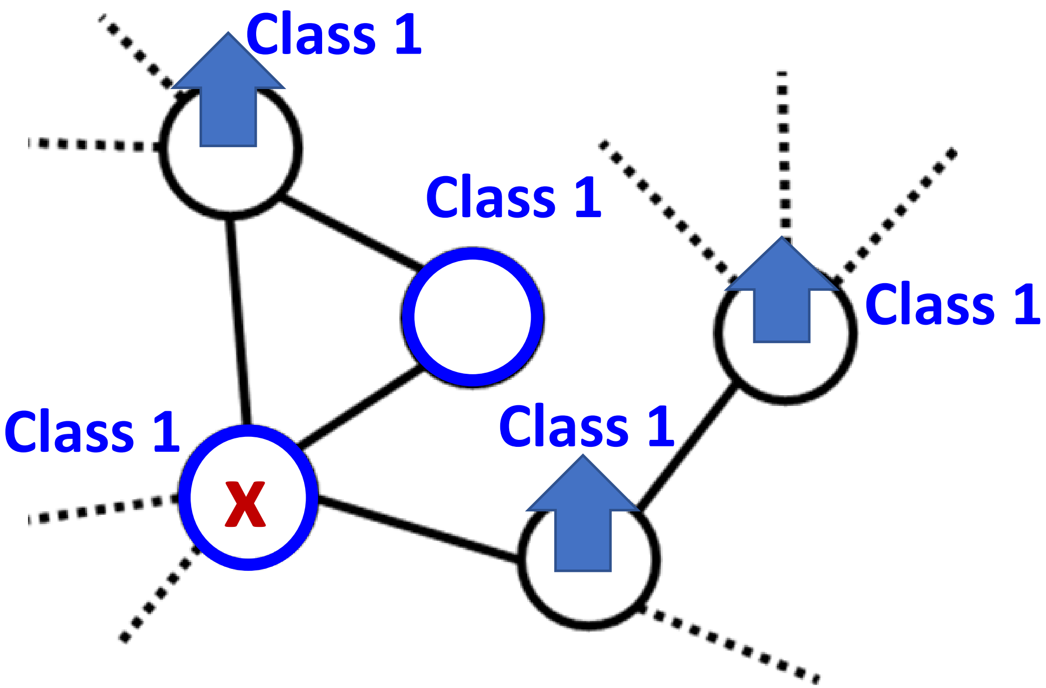


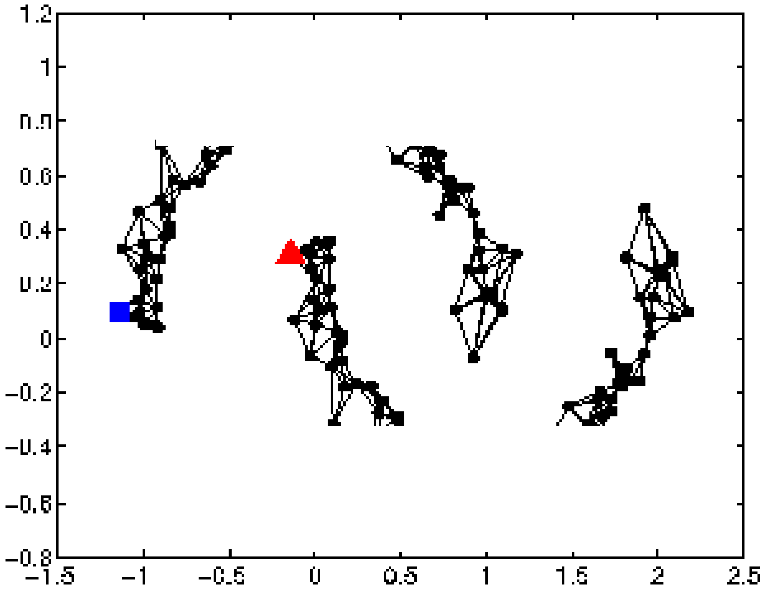
- Edge weight is proportional to $s(x^i, x^j)$
  - ※ Gaussian Radial Basis Function:
  - ※ $s(x^i, x^j) = exp\left(-\gamma \left\| x^i - x^j \right\|^2\right)$

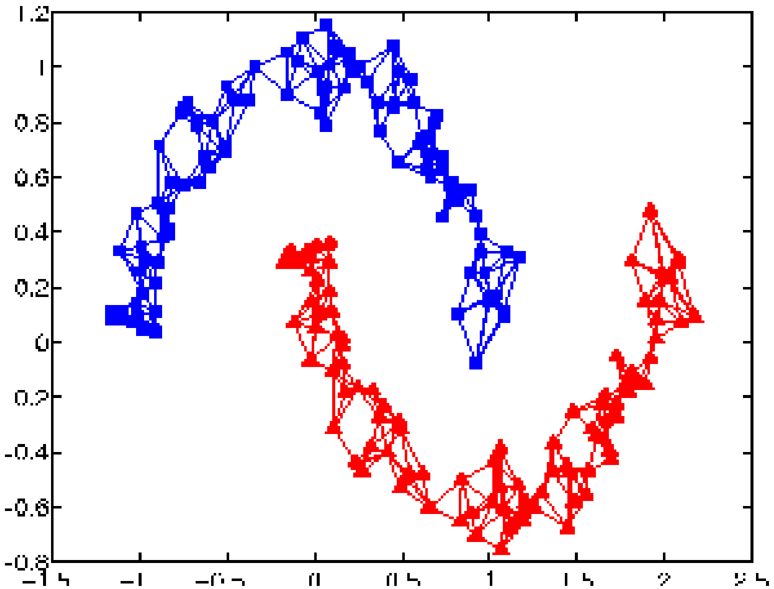


The images are from the tutorial slides of Amarnag Subramanya and Partha Pratim Talukdar

Smoothness Assumption







- The labelled data influence their neighbors.

※ Propagate through the graph

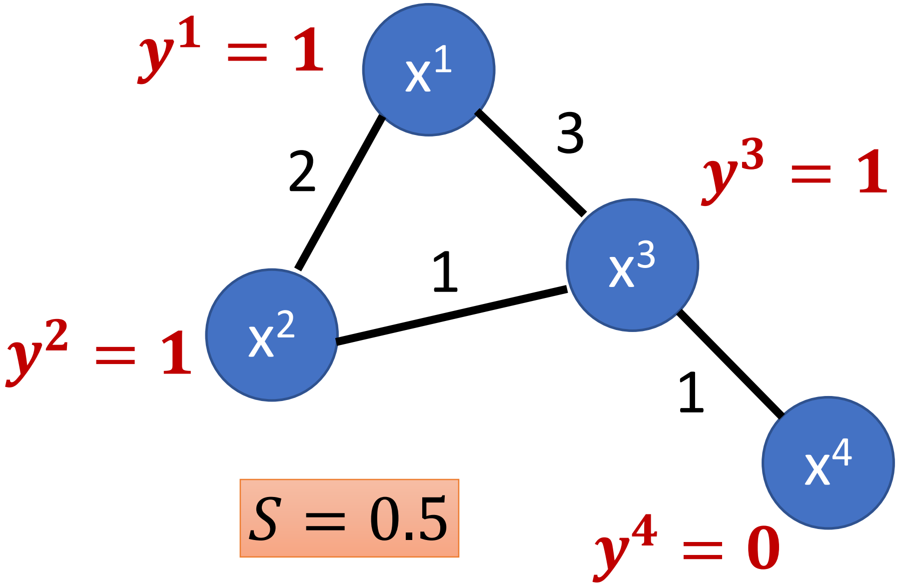
Graph-based Approach

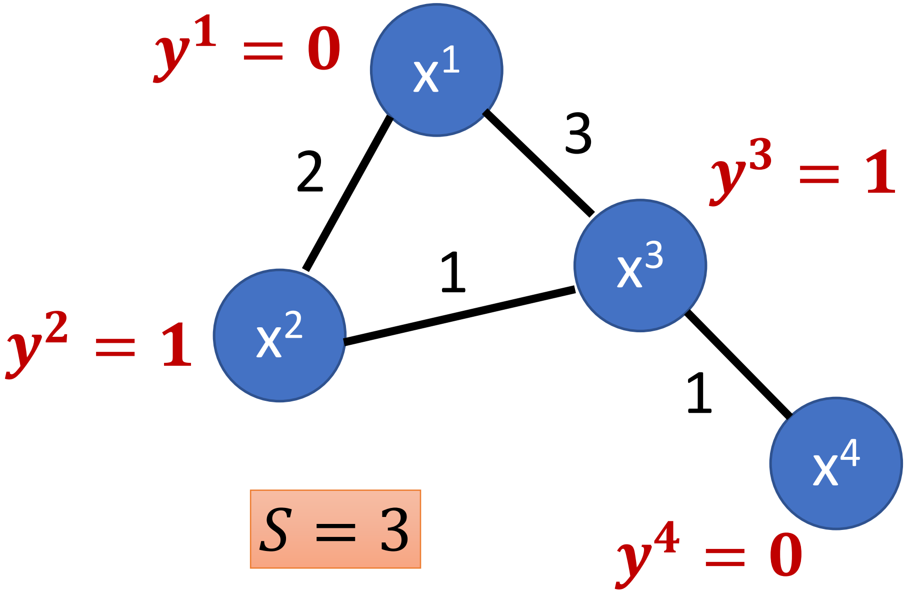
- Define the smoothness of the labels on the graph

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

For all data (no matter labelled or not)

Smaller means smoother

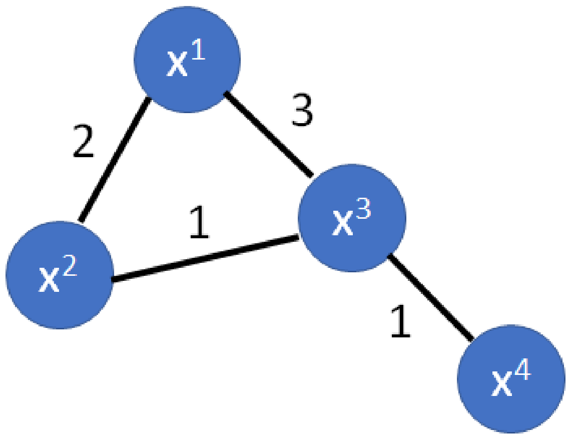




Graph-based Approach

- Define the smoothness of the labels on the graph

$$S = \frac{1}{2}\sum_{i,j} w_{i,j}\left(y^i - y^j\right)^2 = \boldsymbol{y}^T L \boldsymbol{y}$$



※ y: (R+U)-dim vector : $\boldsymbol{y} = \left[\cdots y^i \cdots y^j \cdots\right]^{\boldsymbol{T}}$

※ L: (R+U) x (R+U) matrix -- Graph Laplacian

※ $L = D - W$

$$W = \begin{bmatrix} 0 & 2 & 3 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} \boxed{5} & 0 & 0 & 0 \\ 0 & \boxed{3} & 0 & 0 \\ 0 & 0 & \boxed{5} & 0 \\ 0 & 0 & 0 & \boxed{1} \end{bmatrix}$$

Graph-based Approach

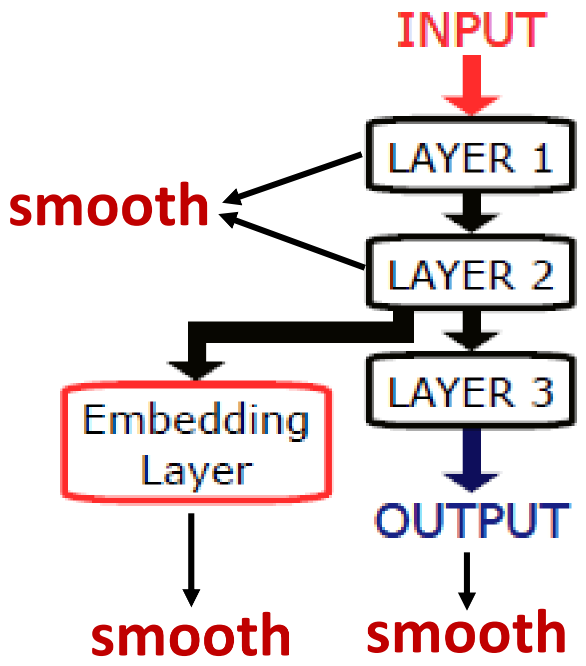
- Define the smoothness of the labels on the graph

$$S = \frac{1}{2}\sum_{i,j} w_{i,j}\left(y^i - y^j\right)^2 = \boldsymbol{y}^T L \boldsymbol{y}$$

Depending on model parameters

$$L = \sum_{x^r} C(y^r, \hat{y}^r) \boxed{+\lambda S}$$

As a regularization term



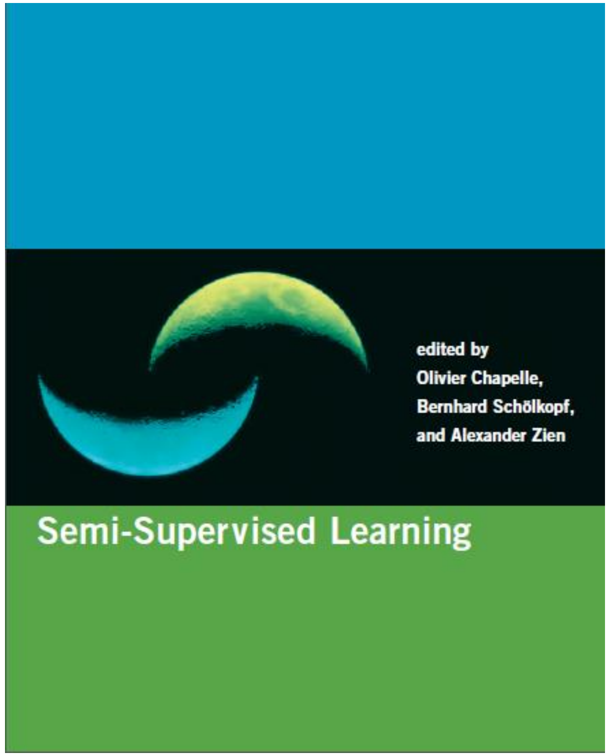
J. Weston, F. Ratle, and R. Collobert, "Deep learning via semi-supervised embedding," ICML, 2008

10.5 Better Representation

Looking for Better Representation

- Find a better (simpler) representations from the unlabeled data
  - ※ Find the latent factors behind the observation
  - ※ The latent factors (usually simpler) are better representations

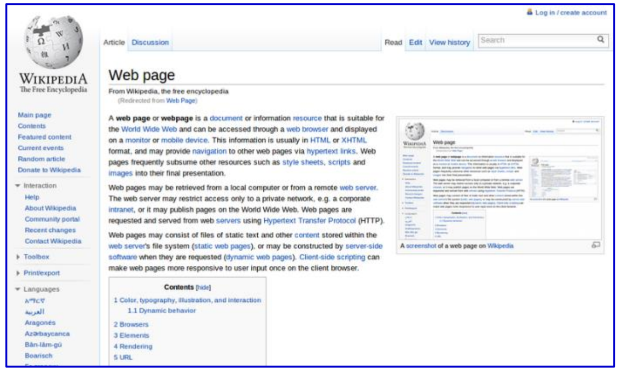
- Reference
  - ※ Semi-supervised learning
  - ※ http://olivier.chapelle.cc/ssl-book/



Multi-view learning

Multi-view learning

- Sometimes, an observation can be represented by
  - ※ two independent sets of features or ‘**views**’.
  - ※ For example a webpage can be characterized by
    - ➢ its content but also by the links’ text pointing to it.
  - ※ This view redundance can be used for semi-supervised learning!



- Multi-view learning
  - ※ Conventional algorithms ‘concatenate’ all views.
  - ※ This approach might cause overfitting with small training sets.
  - ※ Not physically meaningful since each view has specific statistical properties.

Multi-view learning

- Multi-view learning takes advantage of all views
  - ※ to jointly optimize and exploit the redundant views
  - ※ of the same input data to improve performance.

- Co-Training (Blum, A., & Mitchell, T.)
  - ※ is a type of semi-supervised algorithm.
  - ※ Two classifiers work together to enlarge the training set L and increase performance.

Co-training algorithm

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

Randomly choose $2p + 2n$ examples from U to replenish U'

Some implementations use independent L for each view.

Co-training algorithm

- Assumptions
  ※ A feature split into two views exists.
  ※ Each feature split (view) is sufficient to train a good classifier.
  ※ The views are conditionally independent given the class.

- How to combine the results?
  ※ Multiply output probabilities.
  ※ Choose the class with maximum probability among the two models.
  ※ Train a single model after the last iteration.

Multi-view learning

- 协同训练过程虽简单, 但令人惊讶的是:
  - ※ 若两个视图充分且条件独立, 则可利用未标记样本
  - ※ 通过协同训练将弱分类器的泛化性能提升到任意高
  - ※ 理论证明参见：[Blum and Mitchell, 1998].
- Multi-view的条件独立性在现实任务中通常很难满足
  - ※ 因此性能提升幅度不会那么大
  - ※ 虽然如此，协同训练仍可有效地提升弱分类器的性能
- 总体说来：理论基础相对坚实、适用范围较为广泛

Next chapter: Deep Learning