與本次作業（CNN）相關的內容為self-training

Semi-supervised Learning

Introduction

- Supervised learning: $\{(x^r, \hat{y}^r)\}_{r=1}^{R}$
  - E.g. x^r: image, $\hat{y}^r$: class labels
- Semi-supervised learning: $\{(x^r, \hat{y}^r)\}_{r=1}^{R}$, $\{x^u\}_{u=R}^{R+U}$
  - A set of unlabeled data, usually U >> R
  - Transductive learning: unlabeled data is the testing data
  - Inductive learning: unlabeled data is not the testing data

使用testing data當unlabeled data

- Why semi-supervised learning?
  - Collecting data is easy, but collecting "labelled" data is expensive
  - We do semi-supervised learning in our lives

Why semi-supervised learning helps?

Labelled data

cat

dog

Unlabeled data



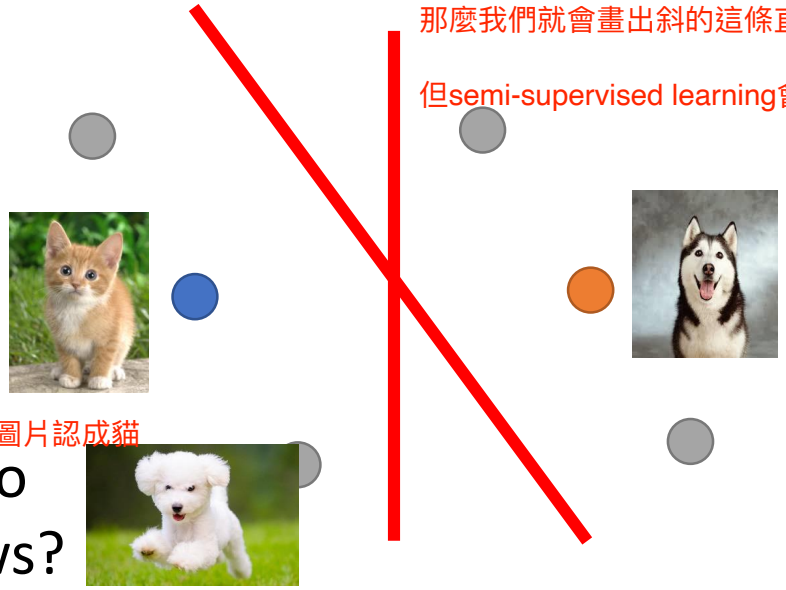

(Image of cats and dogs without labeling)

Why semi-supervised learning helps?

假若只有藍色與橘色這種labeled data
那我們畫出來的線可能是直的

但若多了unlabeled data，就可以從中得知更多data分佈的資訊
那如果我們假設狗的照片會聚集在一群，貓的照片也會聚集在一群
那麼我們就會畫出斜的這條直線

但semi-supervised learning會不會work就取決於假設精不精確



例如semi-supervised就有可能把這張圖片認成貓
但其實是狗，
而導致預測方向更不精確

Who knows?

The distribution of the unlabeled data tell us ***something***.

Usually with some assumptions

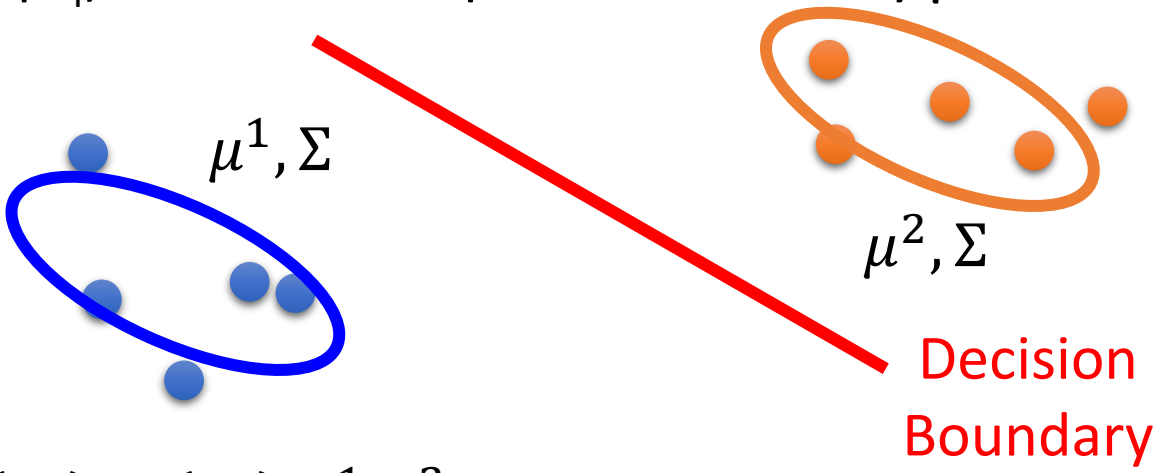
Outline

- Semi-supervised Learning for Generative Model
- Low-density Separation Assumption
- Smoothness Assumption
- Better Representation

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

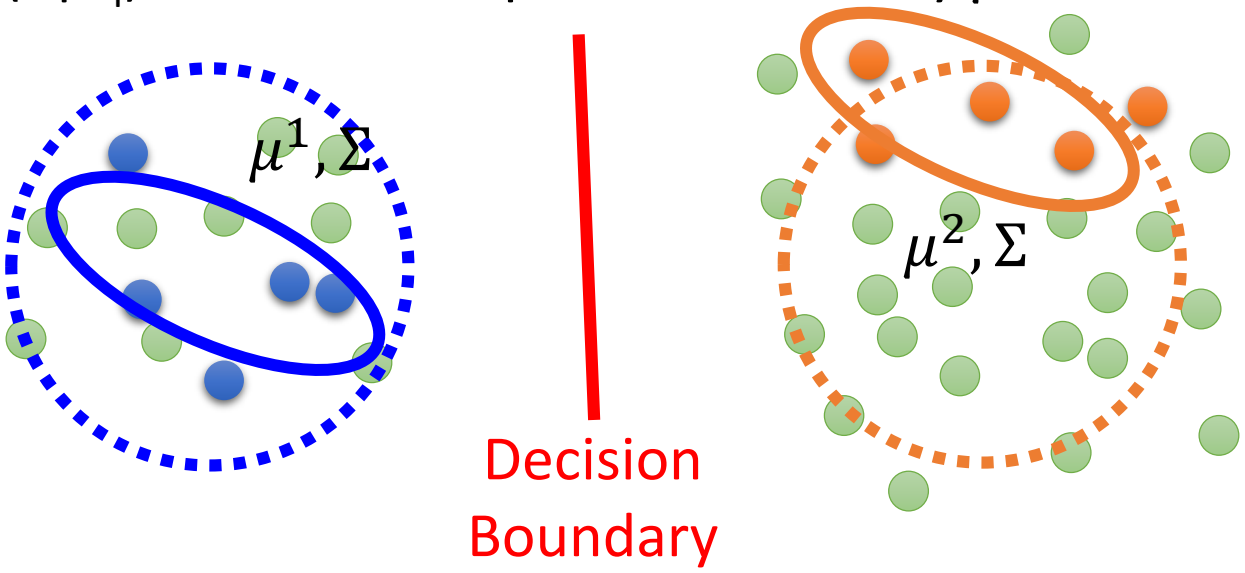


With $P(C_1)$, $P(C_2)$,μ^1,μ^2, Σ

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

Semi-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

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

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

E

- Step 1: compute the posterior probability of unlabeled data

$P_\theta(C_1|x^u)$ Depending on model θ

Back to step 1

M

- Step 2: update model

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

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

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

Why?

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

- Maximum likelihood with labelled data — Closed-form solution

$$logL(\theta) = \sum_{x^r} logP_\theta(x^r, \hat{y}^r)$$

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

- Maximum likelihood with labelled + unlabeled data — Solved iteratively

$$logL(\theta) = \sum_{x^r} logP_\theta(x^r, \hat{y}^r) + \sum_{x^u} logP_\theta(x^u)$$

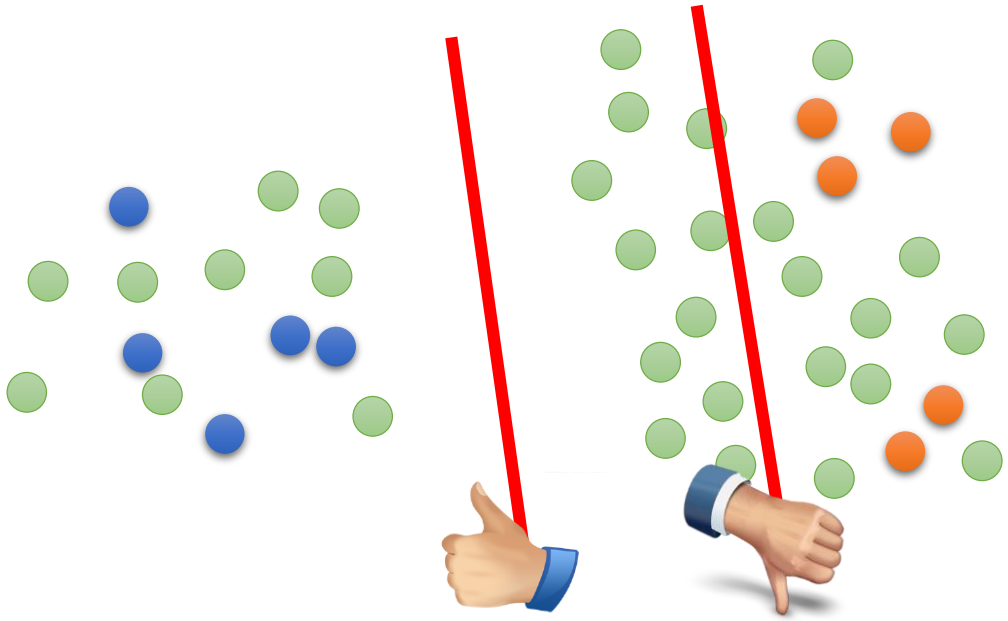
$$P_\theta(x^u) = P_\theta(x^u|C_1)P(C_1) + P_\theta(x^u|C_2)P(C_2)$$

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

Semi-supervised Learning
Low-density Separation

非黑即白

"Black-or-white"



Self-training

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

    Independent to the model

    Regression?

    regression用self-training不會有幫助
  - Apply f^* to the unlabeled data set
    - Obtain $\{(x^u, y^u)\}_{u=l}^{R+U}$ Pseudo-label
  - 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

    You can also provide a weight to each data.

    有些pseudo label比較confidence, weight就比較大

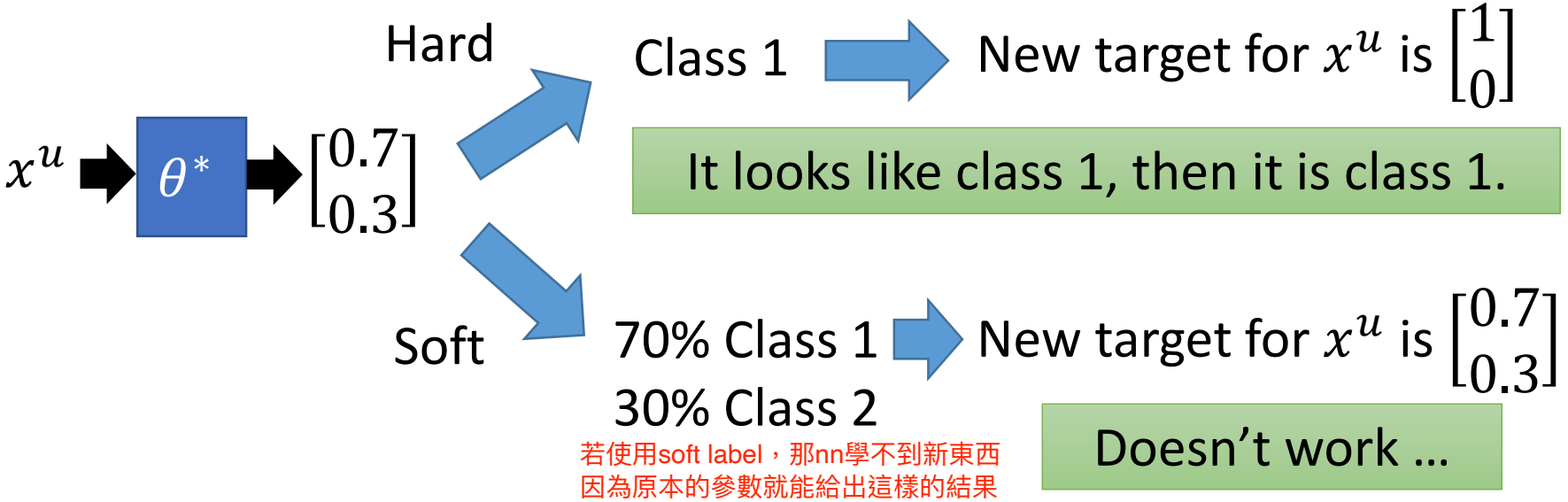
Self-training

- Similar to semi-supervised learning for generative model
- Hard label v.s. Soft label

self-training是hard label, 而前面講的generated是使用soft label(有機率)

Considering using neural network

θ^* (network parameter) from labelled data



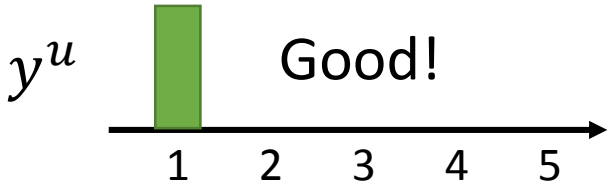
若使用soft label，那nn學不到新東西
因為原本的參數就能給出這樣的結果

但若覺得hard label太果斷，可以使用entropy-based regularization

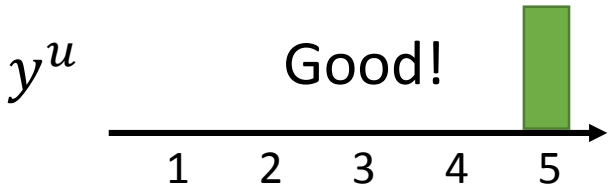
Entropy-based Regularization

Distribution

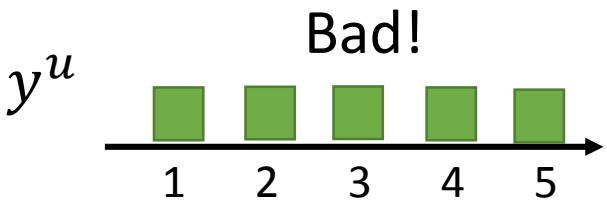
Entropy of y^u :
Evaluate how concentrate the distribution y^u is



$E(y^u) = 0$



$E(y^u) = 0$



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

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

As small as possible

將loss function加上unlabeled data的entropy當作regularization

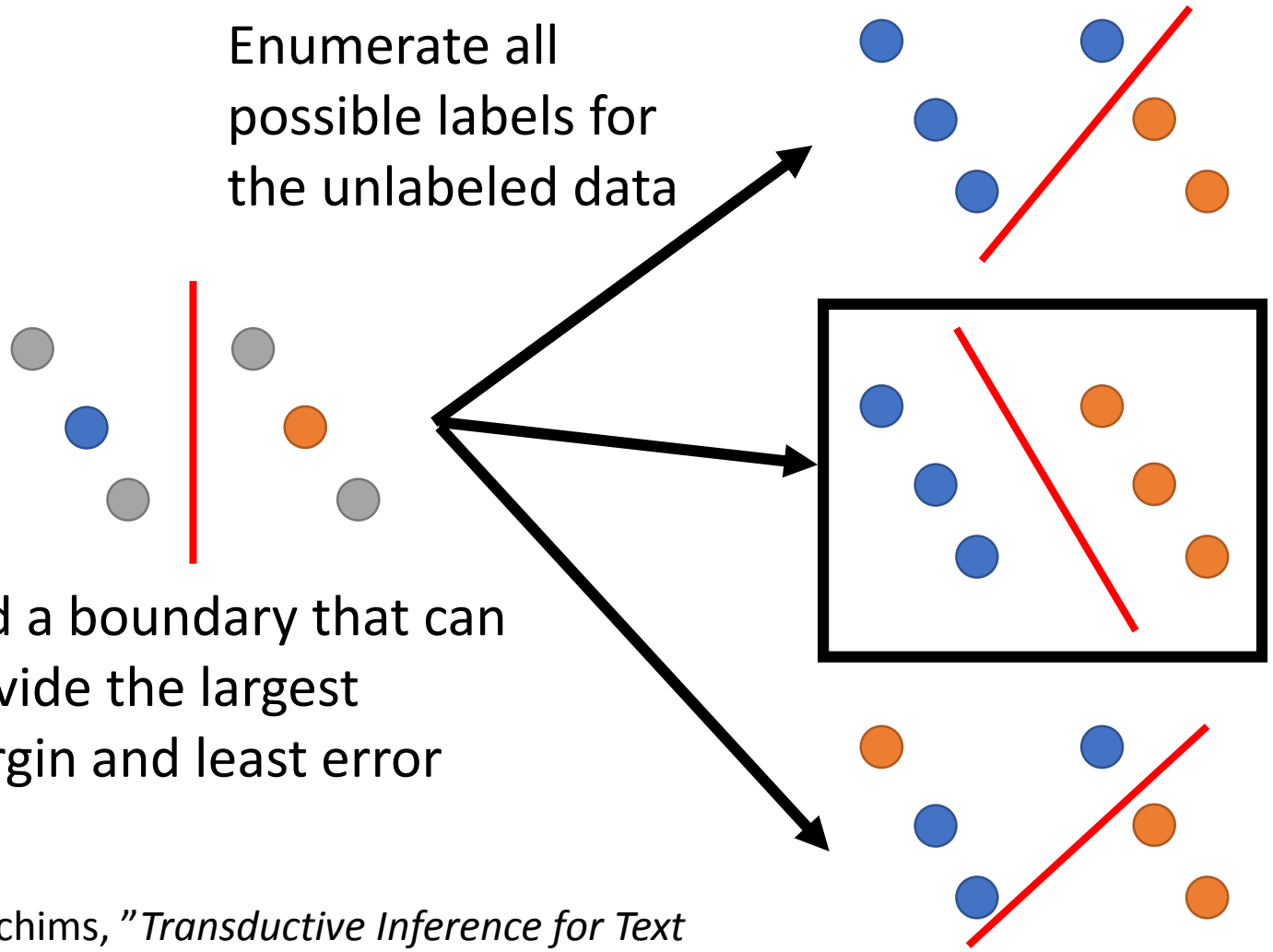
$$L = \sum_{x^r} C(y^r, \hat{y}^r) + \lambda \sum_{x^u} E(y^u)$$

labelled data

unlabeled data

如果model對unlabeled data的預測結果是很平均的話，就不適合拿來做self-training

Outlook: Semi-supervised SVM



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

Semi-supervised Learning
Smoothness Assumption

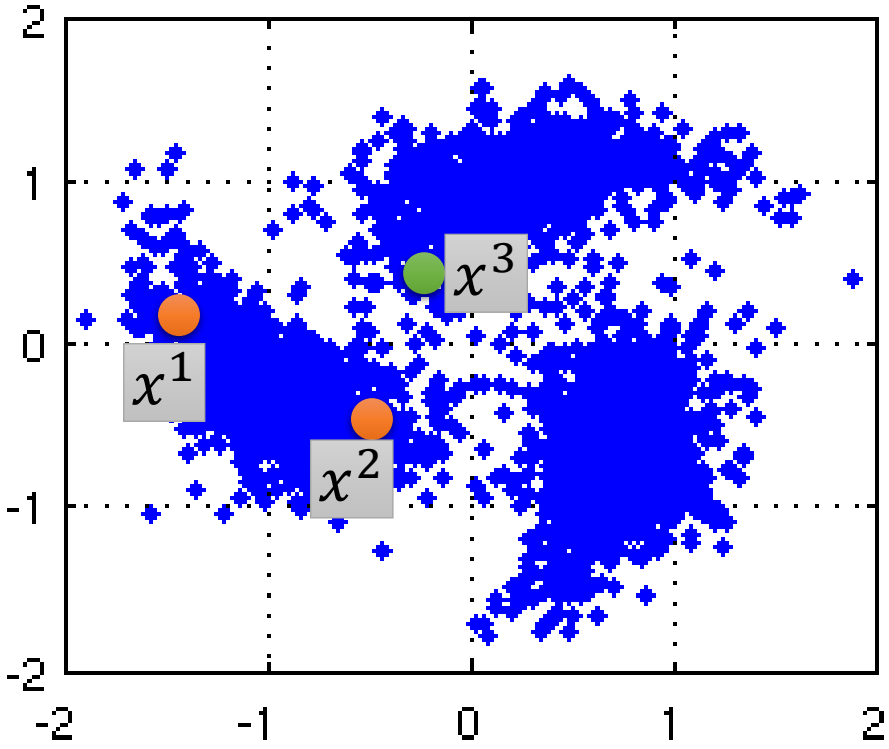
近朱者赤，近墨者黑

“You are known by the company you keep”

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

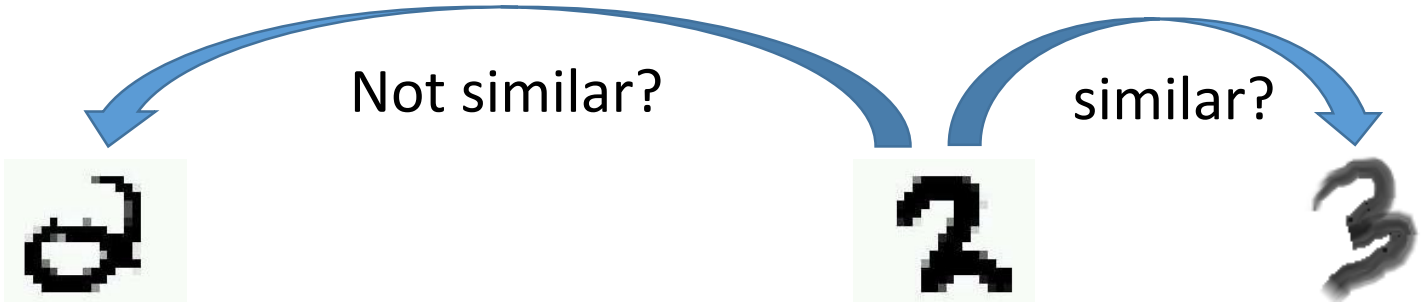


x^1 and x^2 have the same label

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

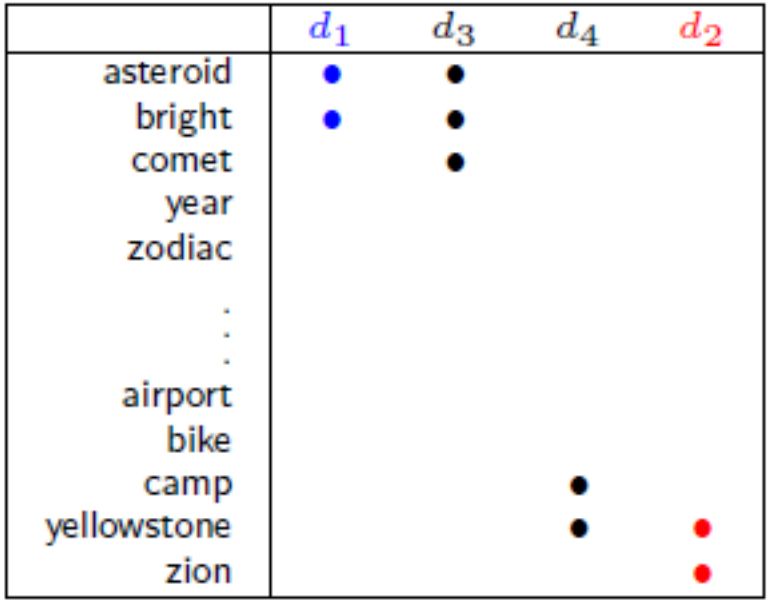
(The example is from the tutorial slides of Xiaojin Zhu.)

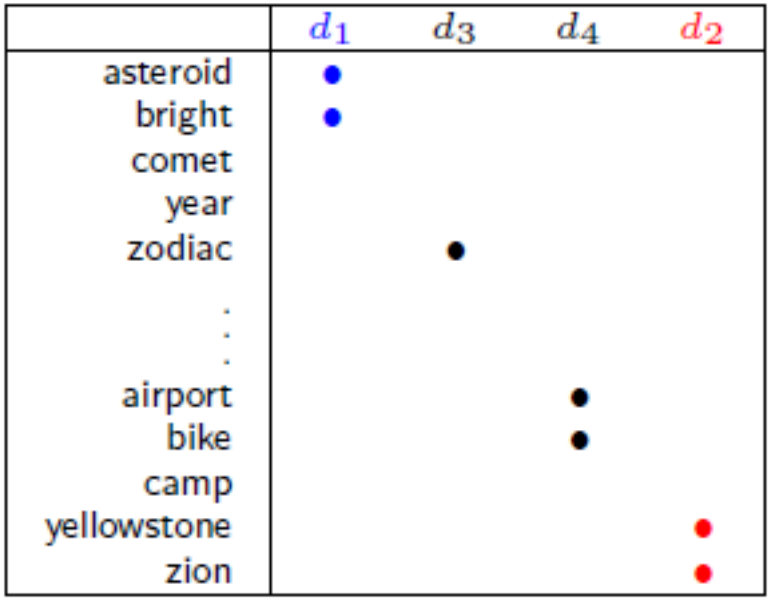
Source of image: http://www.moehui.com/5833.html/5/

Smoothness Assumption

在文章分類裡
smoothness assumption特別有用
（黑色是unlabeled data）

- Classify astronomy vs. travel articles



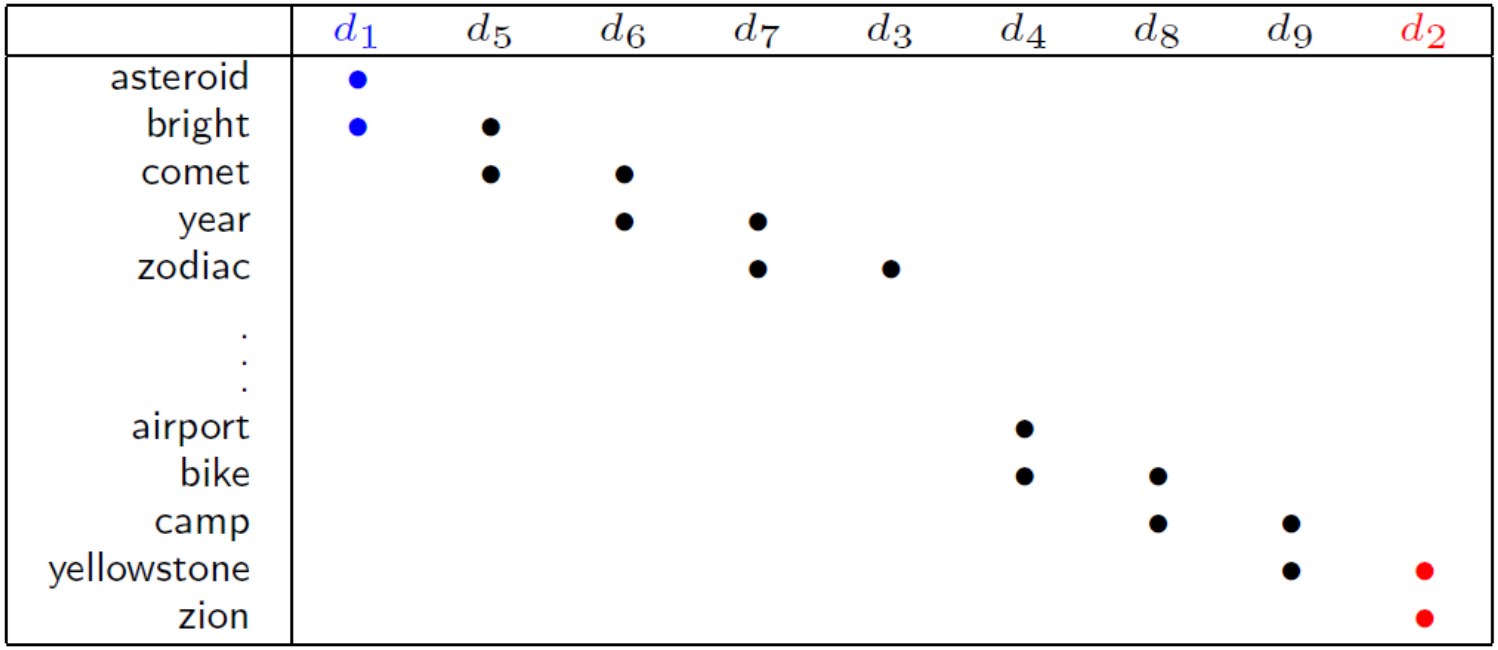


(The example is from the tutorial slides of Xiaojin Zhu.)

Smoothness Assumption

可以知道d1和d3可能是同一類的
而d2和d4可能是同一類的

- Classify astronomy vs. travel articles



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

Cluster and then Label

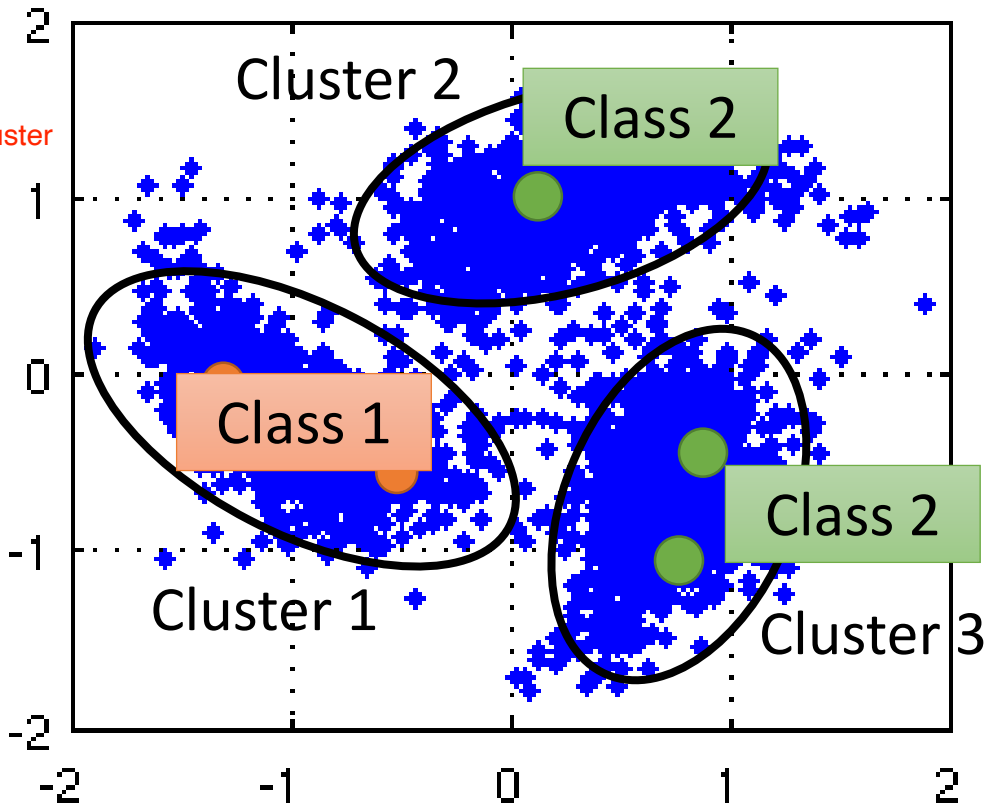
利用cluster把unlabeled data進行分類
如下圖就可以分成cluster1, 2, 3
接著對每一個cluster去看比較像哪個class
就可以進行分類

對image你若直接對pixel做cluster幾乎不可能做出好的結果
因為同樣的背景，但是可能class為狗或貓
或是不同的背景，但都是狗，而且有些旋轉平移
這些都會導致pixel clustering難度上升

但是可以使用deep autoencoder抽feature再做cluster

Class 1

Class 2



Using all the data to learn a classifier as usual

Graph-based Approach

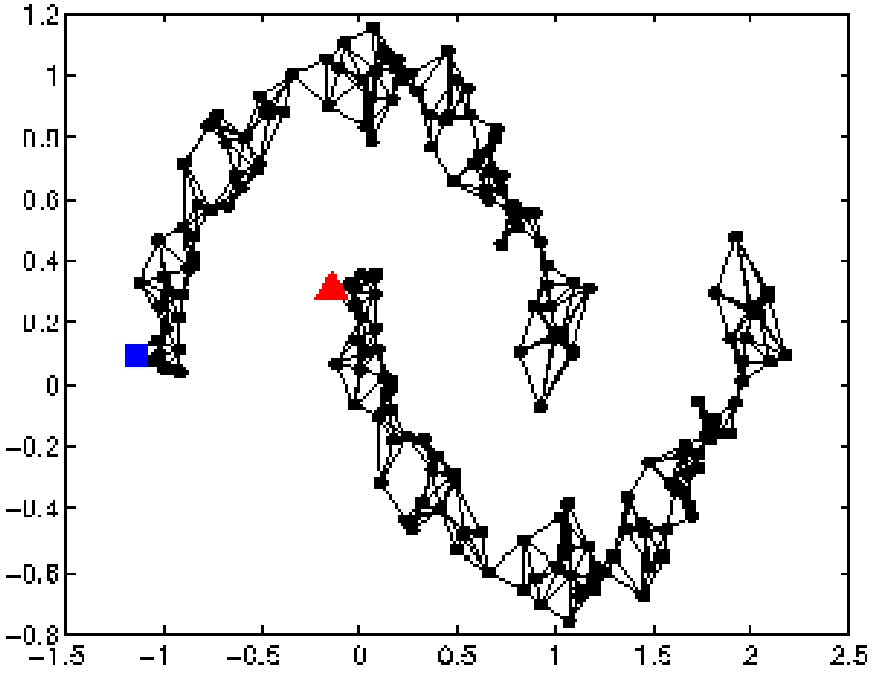
- How to know x^1 and x^2 are close in a high density region (connected by a high density path)

Represented the data points as a ***graph***

Graph representation is nature sometimes.

E.g. Hyperlink of webpages, citation of papers

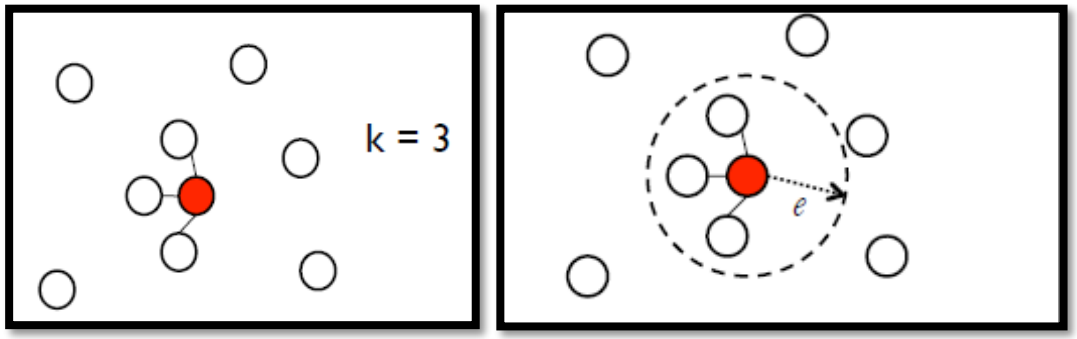
Sometimes you have to construct the graph yourself.



Graph-based Approach - Graph Construction

The image is from the tutorial slides of Amarnag Subramanya and Partha Pratim Talukdar

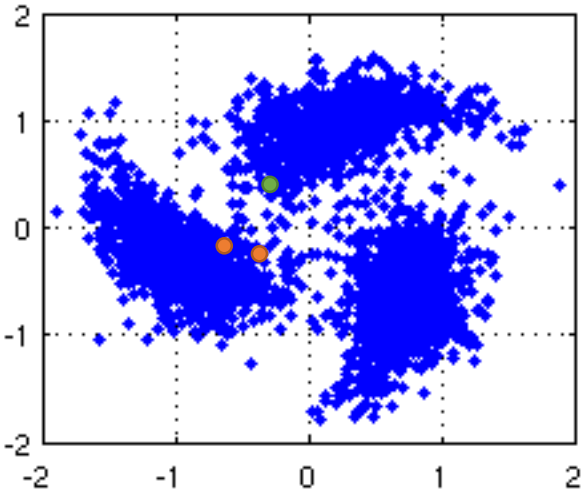
- Define the similarity $s(x^i, x^j)$ between x^i and x^j
- Add edge:
  - K Nearest Neighbor
  - e-Neighborhood



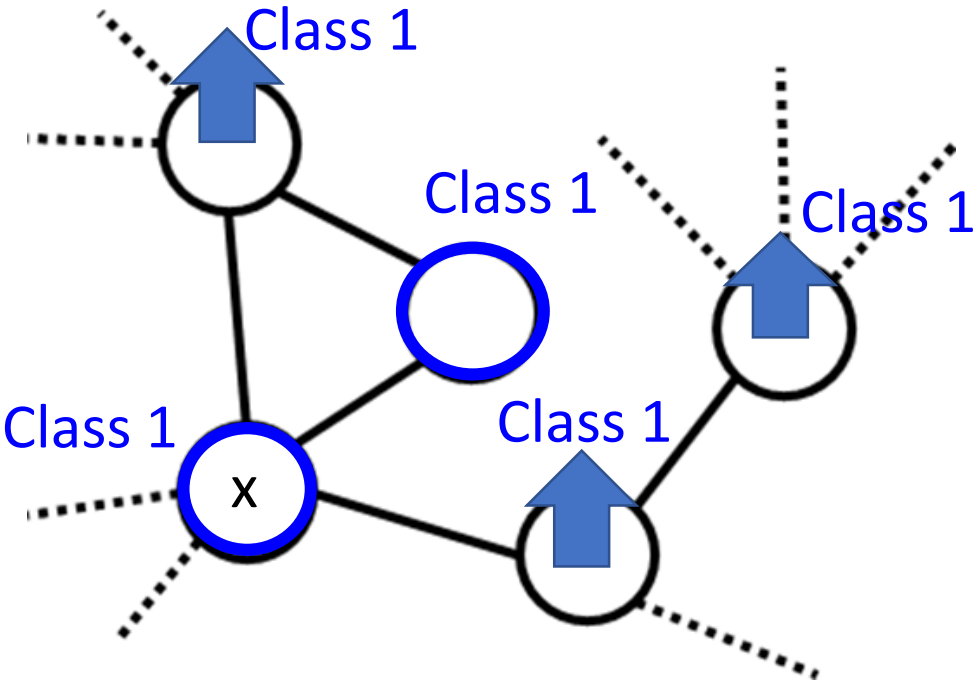
- Edge weight is proportional to $s(x^i, x^j)$

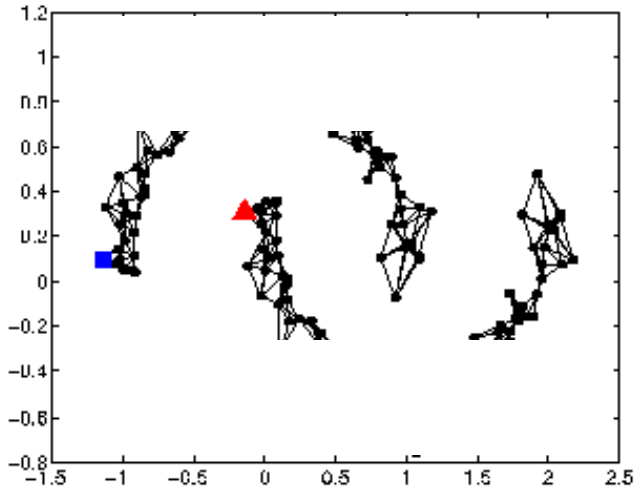
Gaussian Radial Basis Function:

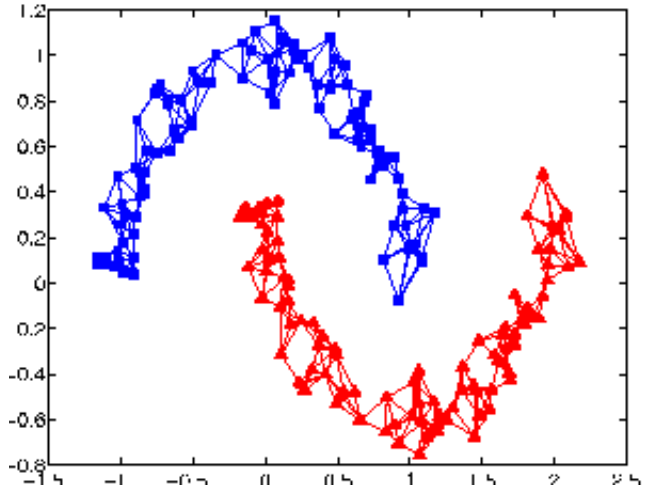
$$s(x^i, x^j) = exp\left(-\gamma \left\| x^i - x^j \right\|^2\right)$$



Graph-based Approach







The labelled data influence their neighbors.

Propagate through the graph

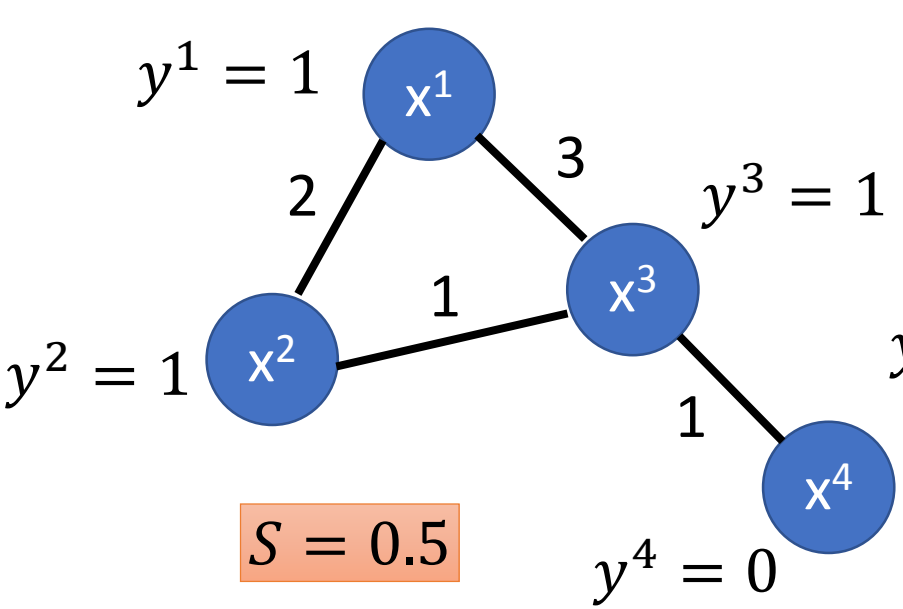
Graph-based Approach

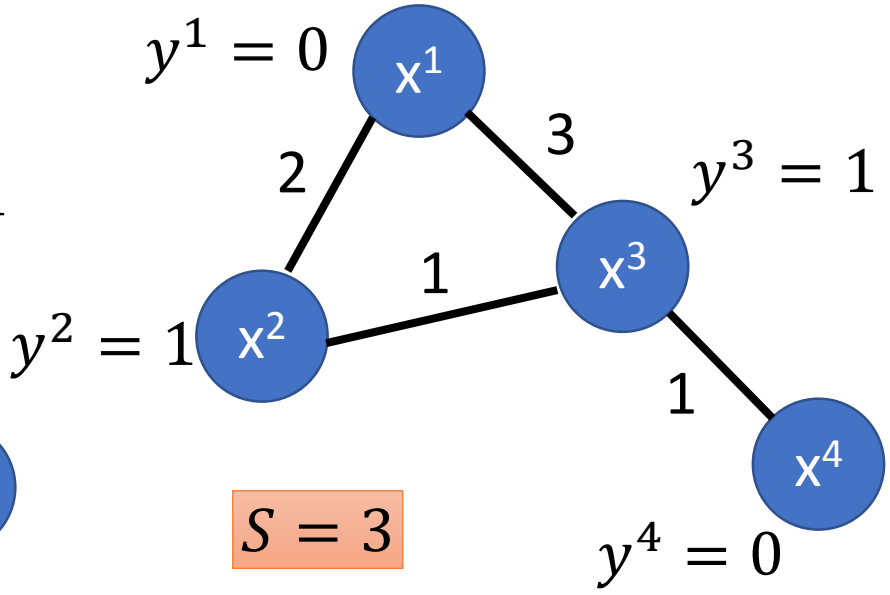
- Define the smoothness of the labels on the graph

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

For all data (no matter labelled or not)

Smaller means smoother

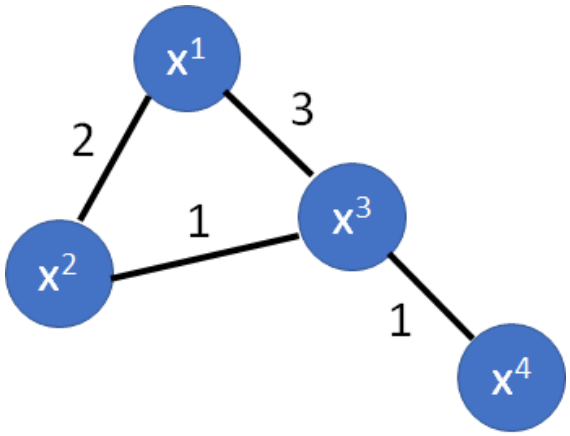




Graph-based Approach

- 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

$$L = D - W$$

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

Graph-based Approach

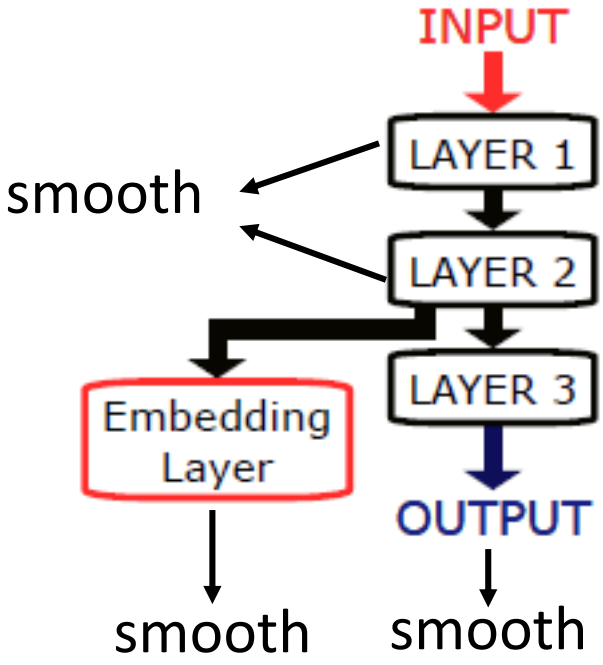
- Define the smoothness of the labels on the graph

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

Depending on network parameters

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

As a regularization term



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

Semi-supervised Learning
Better Representation

去蕪存菁，化繁為簡

Looking for Better Representation

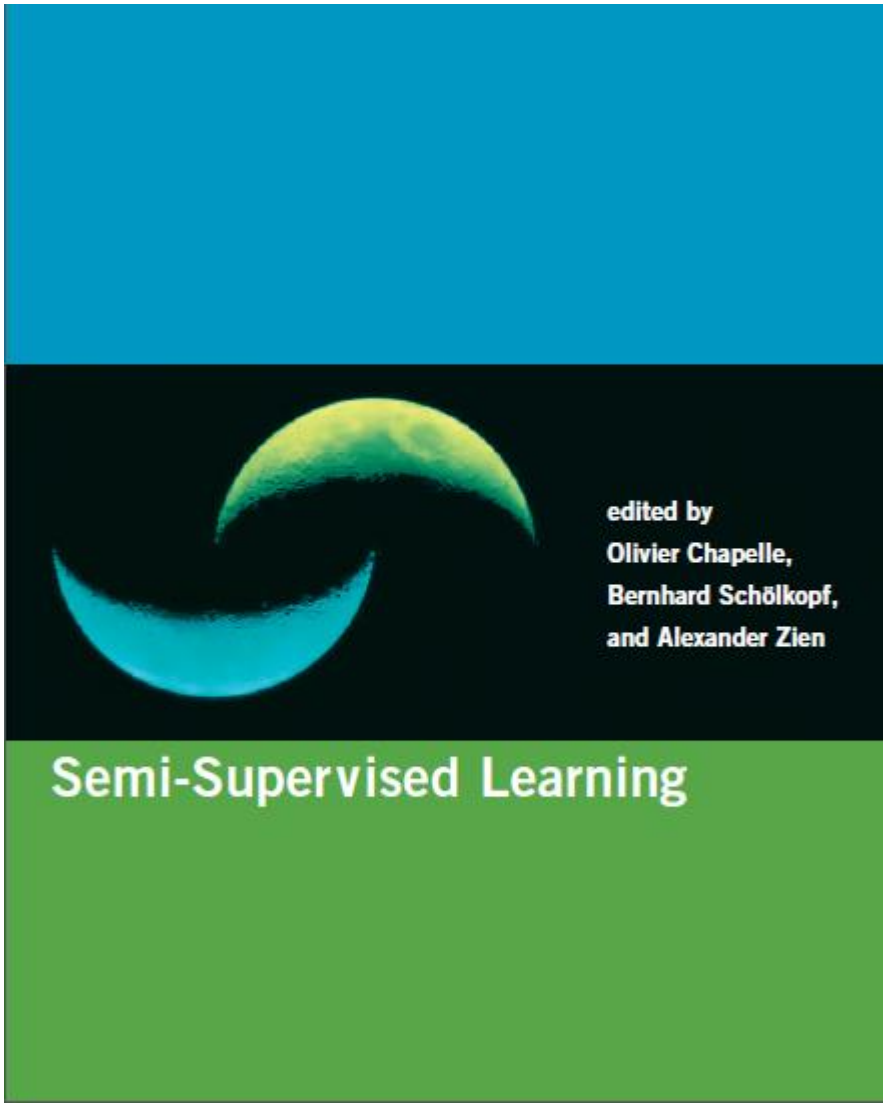
- Find the latent factors behind the observation
- The latent factors (usually simpler) are better representations

observation

Better representation
(Latent factor)

(In unsupervised learning part)

Reference



http://olivier.chapelle.cc/ssl-book/

Acknowledgement

- 感謝 劉議隆 同學指出投影片上的錯字
- 感謝 丁勃雄 同學指出投影片上的錯字