Introduction of Structured Learning

Hung-yi Lee

Structured Learning

- We need a more powerful function *f*
  - Input and output are both objects with structures
  - *Object*: sequence, list, tree, bounding box …

$$f : X \rightarrow Y$$

X is the space of one kind of object

Y is the space of another kind of object

In the previous lectures, the input and output are both vectors.

Introduction of Structured Learning

Unified Framework

Unified Framework

Training

- Find a function F

$$F: X \times Y \rightarrow R$$

- F(x,y): evaluate how compatible the objects x and y is

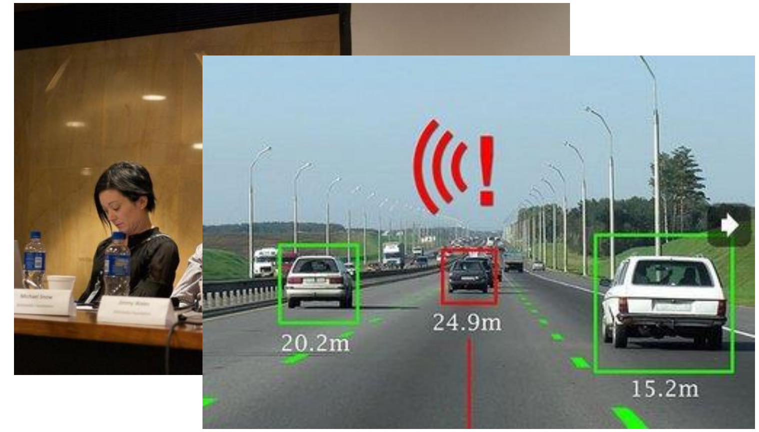
Inference (Testing)

- Given an object x

$$\tilde{y} = \arg\max_{y \in Y} F(x, y)$$

$$f: X \rightarrow Y \quad \Rightarrow \quad f(x) = \tilde{y} = \arg\max_{y \in Y} F(x, y)$$

Unified Framework – Object Detection



- Task description
  - Using a bounding box to highlight the position of a certain object in an image
  - E.g. A detector of Haruhi

X : Image ⟶ Y : Bounding Box

Haruhi

(the girl with yellow ribbon)

Unified Framework – Object Detection

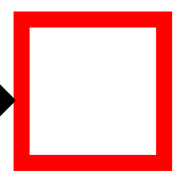
Training

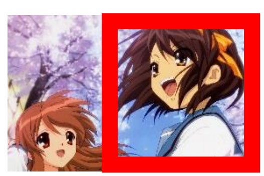
- Find a function F

$$F: X \times Y \rightarrow R$$

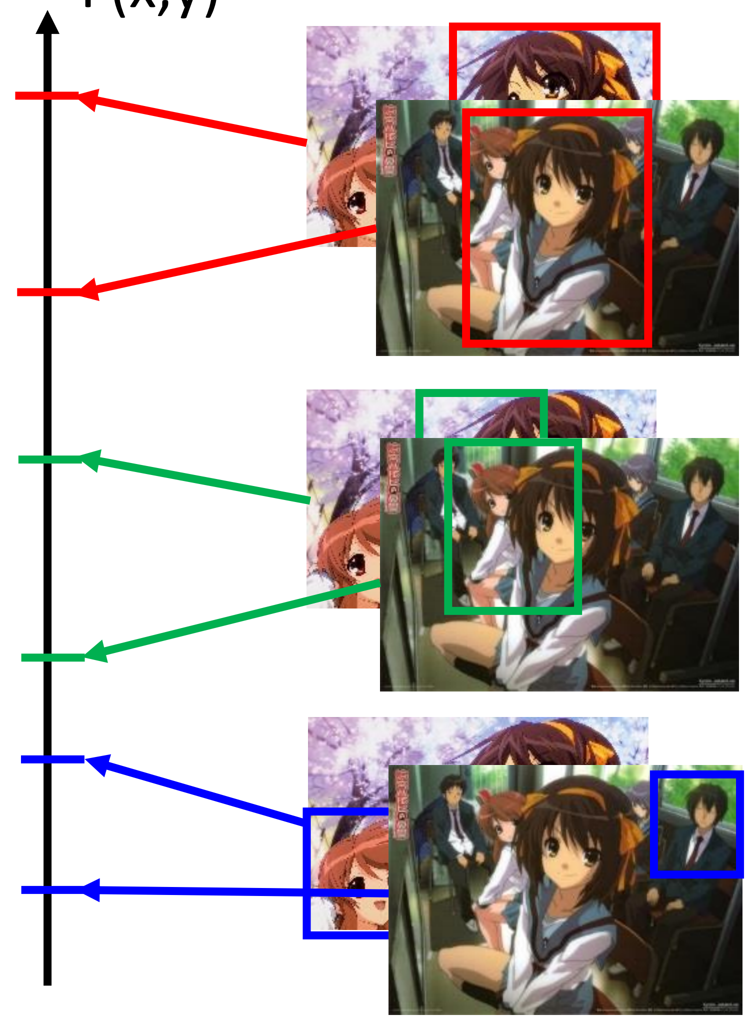
- F(x,y): evaluate how compatible the objects x and y is

x: Image 

y: Bounding Box 

F(x,y) → F()

the correctness of taking range of y in x as "Haruhi"



Unified Framework – Object Detection

Training

- Find a function F

$$F : X \times Y \rightarrow R$$

- F(x,y): evaluate how compatible the objects x and y is

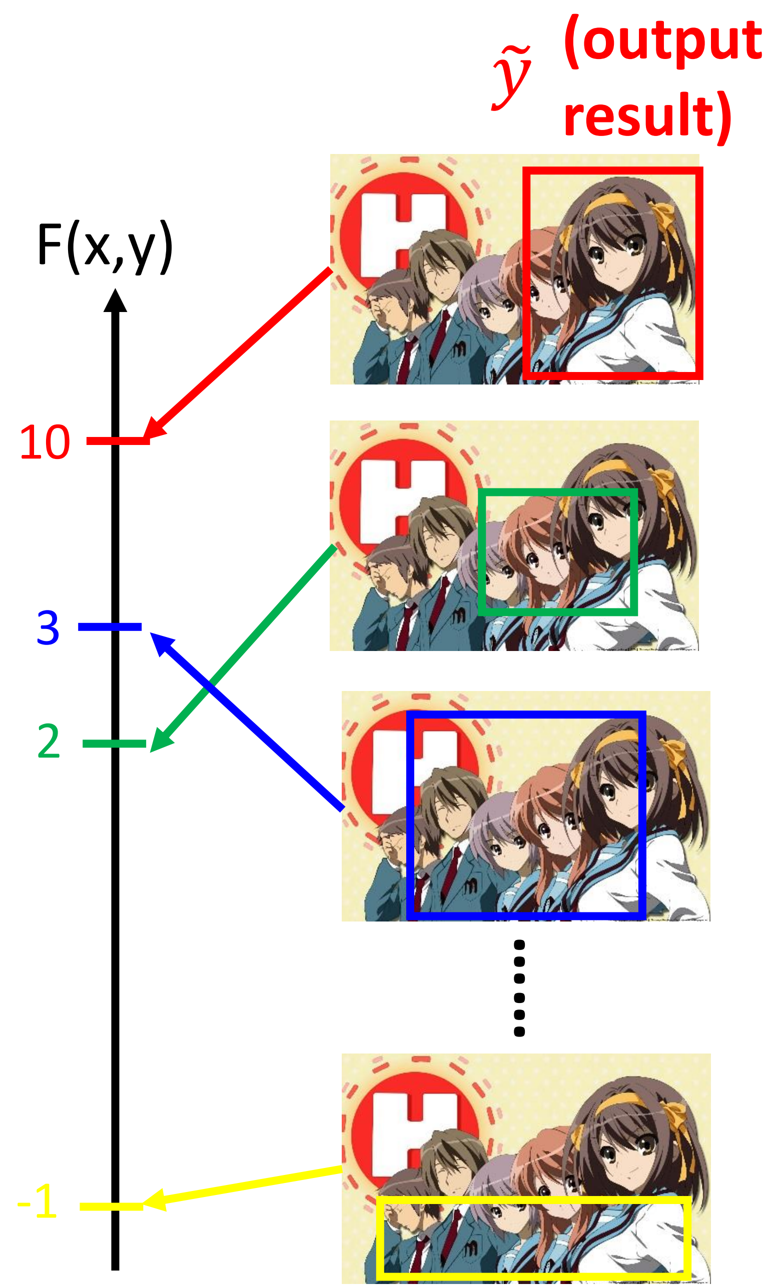
Inference (Testing)

- Given an object x

$$\tilde{y} = \arg\max_{y \in Y} F(x, y)$$

input x = 

Enumerate all possible bounding box y



Unified Framework
- Summarization

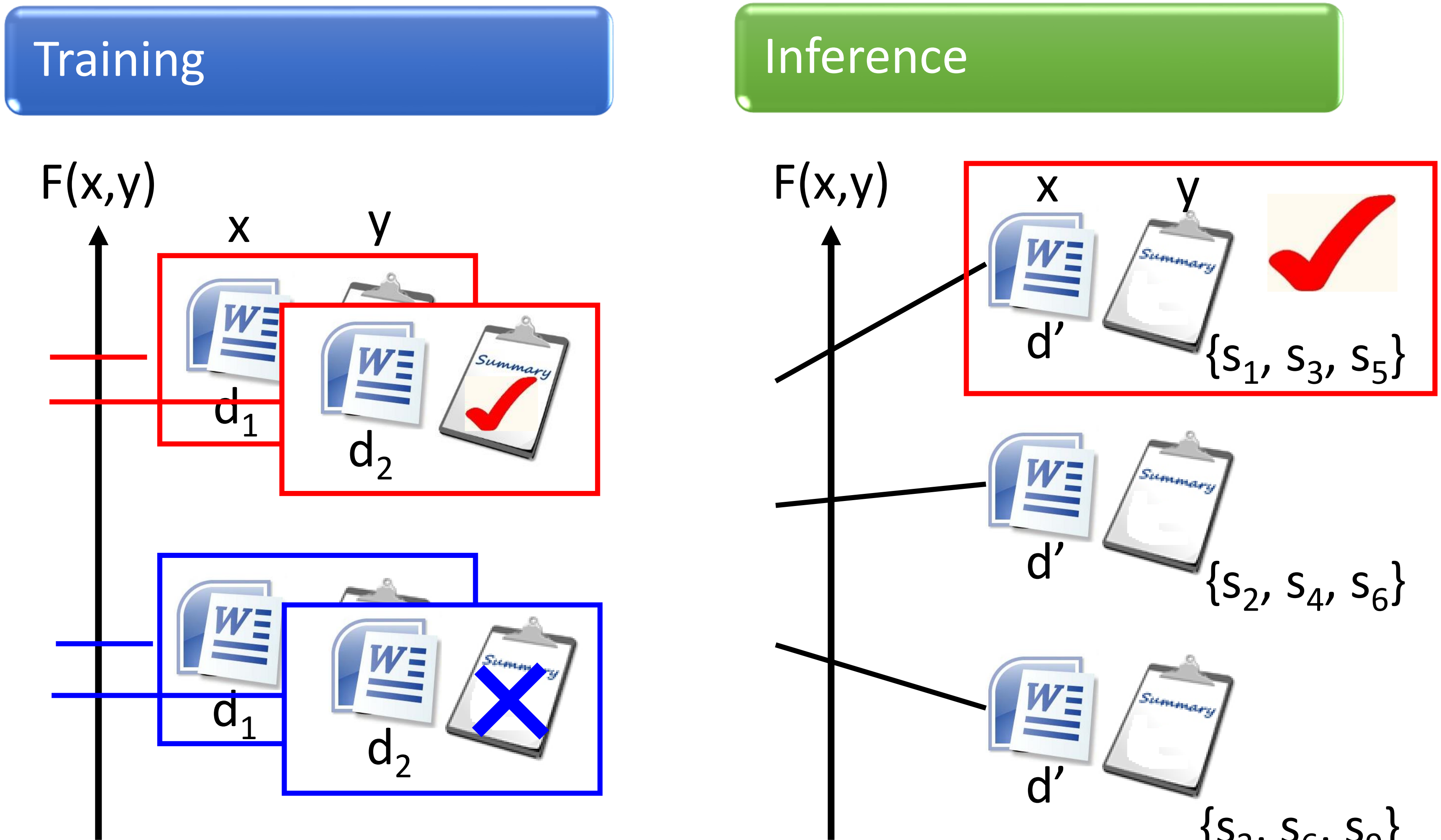
- Task description
  - Given a long document
  - Select a set of sentences from the document, and cascade the sentences to form a short paragraph

X $\quad$ Y

long document
$=\{s_1, s_2, s_3, \ldots\ldots s_i \ldots\}$
s_i: the i^{th} sentence

summary
$=\{s_1, s_3, s_5\}$

Unified Framework
- Summarization



Unified Framework
- Retrieval

- Task description
  - User input a keyword Q
  - System returns a ***list*** of web pages

X $\rightarrow$ Y

"Obama"
(keyword)

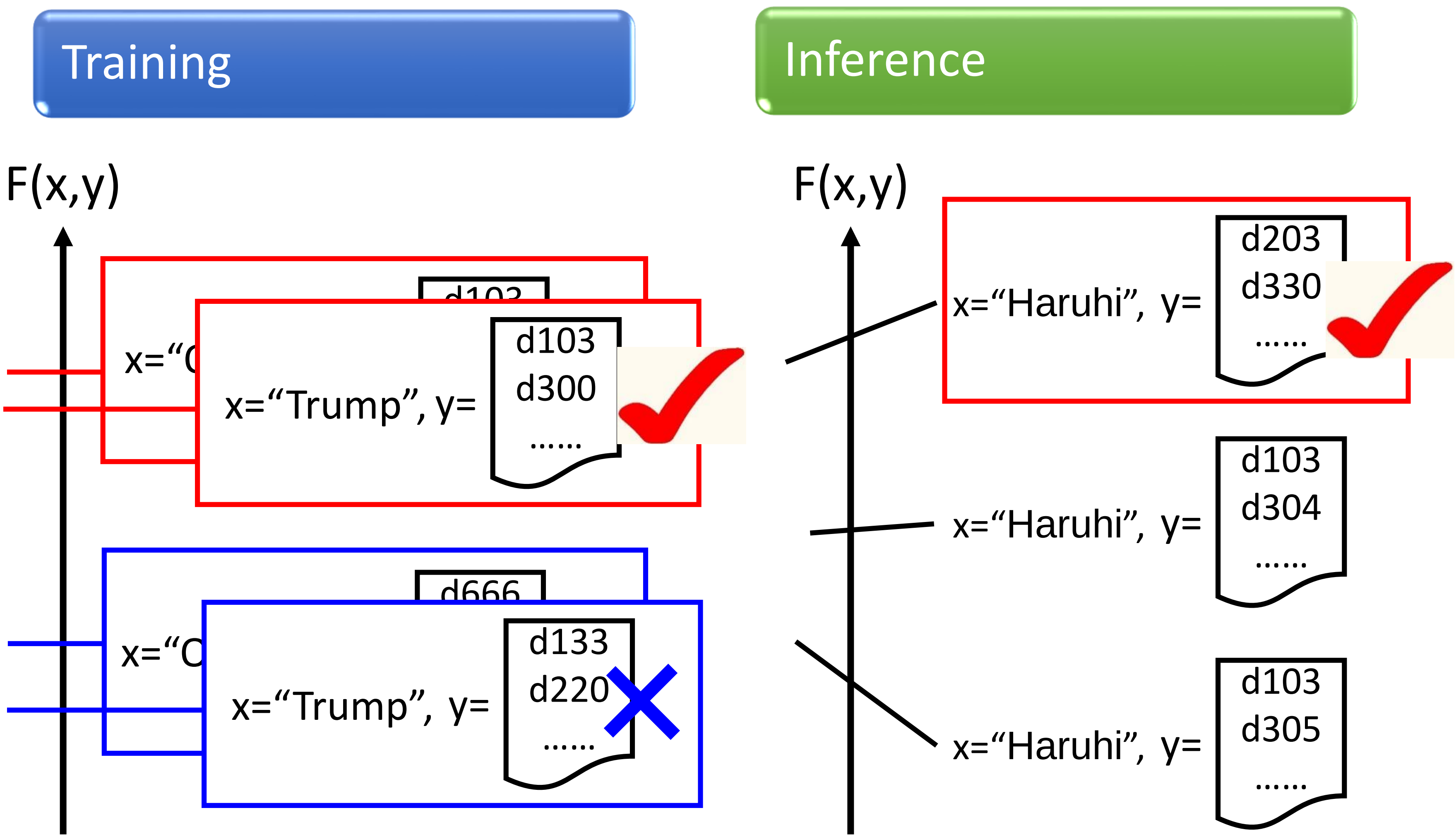
d10011

d98776

⋮

A list of web pages (Search Result)

Unified Framework - Retrieval



Statistics

Unified Framework

Training

- Find a function F

$$F: X \times Y \rightarrow R$$

- F(x,y): evaluate how compatible the objects x and y is

Inference

- Given an object x

$$\tilde{y} = \arg\max_{y \in Y} F(x, y)$$

$$F(x, y) = P(x, y)?$$

Training

- Estimate the probability P(x,y)

$$P: X \times Y \rightarrow [0,1]$$

Inference

- Given an object x

$$\tilde{y} = \arg\max_{y \in Y} P(y \mid x)$$

$$= \arg\max_{y \in Y} \frac{P(x, y)}{P(x)}$$

$$= \arg\max_{y \in Y} P(x, y)$$

Statistics

Unified Framework

$$F(x, y) = P(x, y)?$$

Drawback for probability

- Probability cannot explain everything
- 0-1 constraint is not necessary

Strength for probability

- Meaningful

Energy-based Model: http://www.cs.nyu.edu/~yann/research/ebm/

Training

- Estimate the probability P(x,y)

$$P : X \times Y \to [0,1]$$

Inference

- Given an object x

$$\tilde{y} = \arg\max_{y \in Y} P(y \mid x)$$

$$= \arg\max_{y \in Y} \frac{P(x, y)}{P(x)}$$

$$= \arg\max_{y \in Y} P(x, y)$$

Unified Framework

That’s it!?

Training

- Find a function F

$$F: X \times Y \rightarrow R$$

- F(x,y): evaluate how compatible the objects x and y is

Inference (Testing)

- Given an object x

$$\tilde{y} = \arg\max_{y \in Y} F(x, y)$$

There are three problems in this framework.

Problem 1

- ***Evaluation***: What does F(x,y) look like?
  - How F(x,y) compute the "compatibility" of objects x and y

Object Detection: F(x=  , y=)

Summarization: F(x=  , y= )

(a long document) (a short paragraph)

Retrieval: F(x= "Obama" , y= )

Barack Obama
https://www.barackobama.com/ 翻譯這個網頁
2015年1月6日 - President Obama is fighting hard to make change in his final two years in office. It's never been more important to take a stand, and join this ...

Barack Obama - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Barack_Obama
Barack Hussein Obama II is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, ...
Barack Obama, Sr. - Early life and career of Barack - Ann Dunham - Michelle Obama

(keyword) (Search Result)

Problem 2

- **Inference**: How to solve the "arg max" problem

$$y = \arg\max_{y \in Y} F(x, y)$$

The space *Y* can be extremely large!

Object Detection: *Y*=All possible bounding box (maybe tractable)

Summarization: *Y*=All combination of sentence set in a document …

Retrieval: *Y*=All possible webpage ranking ….

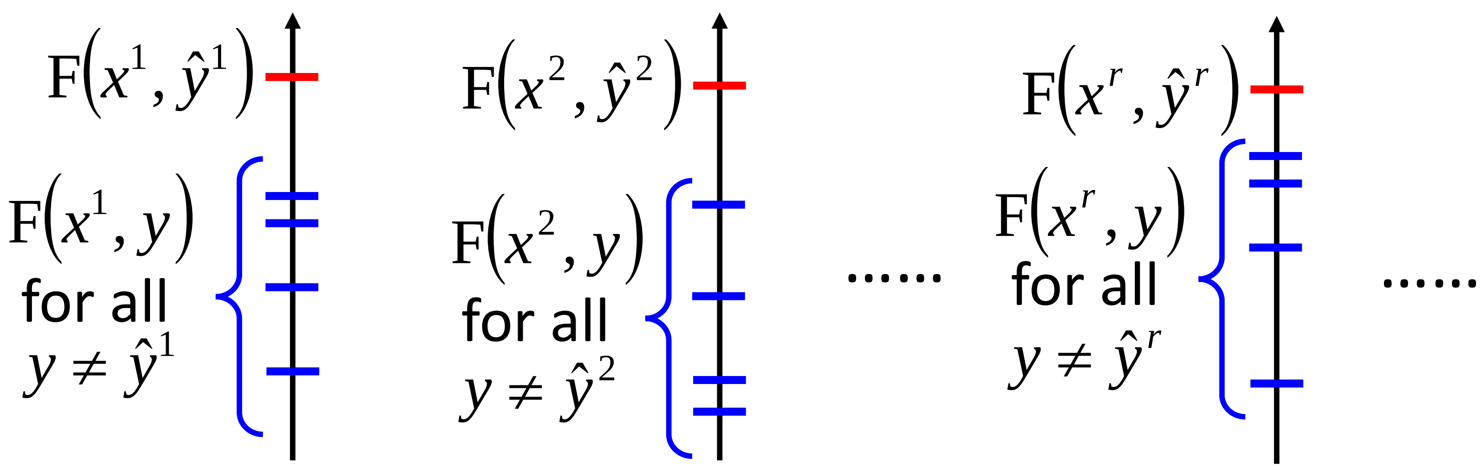
Problem 3

- ***Training***: Given training data, how to find F(x,y)

Principle

Training data: $\left\{\left(x^1,\hat{y}^1\right),\left(x^2,\hat{y}^2\right),\ldots,\left(x^r,\hat{y}^r\right),\ldots\right\}$

We should find F(x,y) such that ……



Three Problems

Problem 1: Evaluation

- What does F(x,y) look like?

Problem 2: Inference

- How to solve the “arg max” problem

$$y = \arg\max_{y \in Y} F(x, y)$$

Problem 3: Training

- Given training data, how to find F(x,y)

Three Problems

Problem 1: Evaluation

- What does F(x,y) look like?

Problem 2: Inference

- How to solve the “arg max

$y = \arg m$

Problem 3: Training

- Given training data, how t

Have you heard the three problems elsewhere?

Hidden Markov Model

- **Three Basic Problems for HMMs**

Given an observation sequence $\overline{O}=(o_1,o_2,\ldots,o_T)$, and an HMM $\lambda=(A,B,\pi)$

- Problem *1* :
  How to *efficiently* compute $P(\overline{O}|\lambda)$?
  ⇨ *Evaluation problem*
- Problem *2* :
  How to choose an optimal state sequence $\mathbf{q}=(q_1,q_2,\ldots\ldots, q_T)$?
  ⇨ *Decoding Problem*
- Problem *3* :
  Given some observations $\overline{O}$ for the HMM λ, how to adjust the model parameter $\lambda=(A,B,\pi)$ to maximize $P(\overline{O}|\lambda)$?
  ⇨ *Learning /Training Problem*

From 數位語音處理

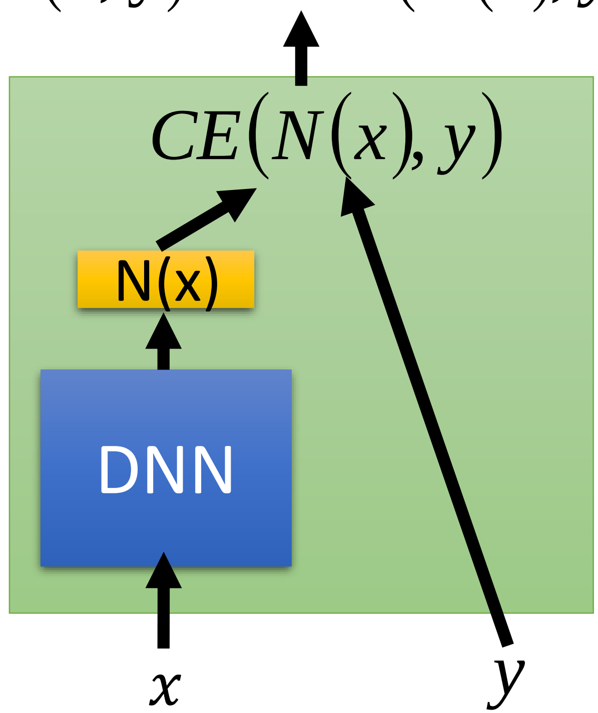
Link to DNN?

The same as what we have learned.

Training

$$F: X \times Y \rightarrow R$$

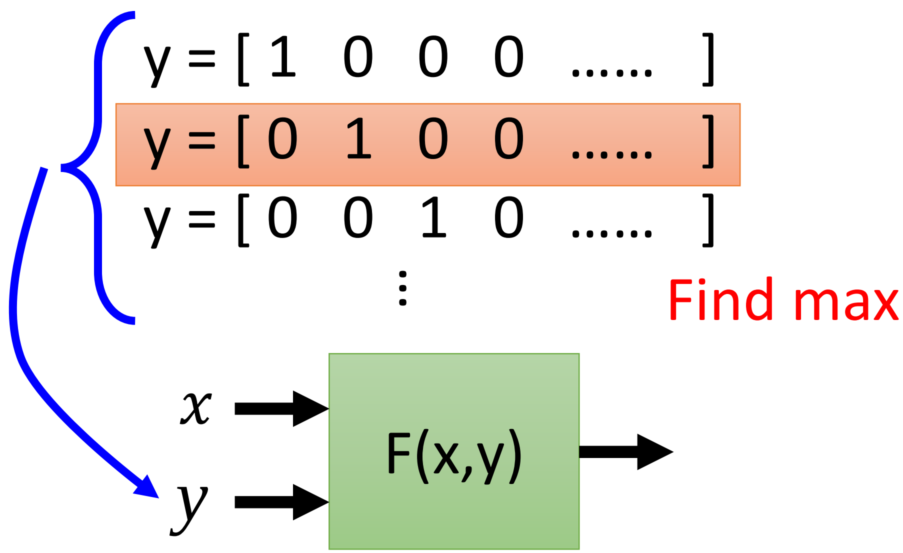
$$F(x, y) = -CE(N(x), y)$$



Inference

$$\tilde{y} = \arg\max_{y \in Y} F(x, y)$$

In handwriting digit classification, there are only 10 possible y.



Introduction of Structured Learning

Linear Model

Structured Linear Model

Problem 1: Evaluation

- What does F(x,y) look like? ➡ in a specific form

Problem 2: Inference

- How to solve the "arg max" problem

$$y = \arg\max_{y \in Y} F(x, y)$$

~~**Problem 3: Training**~~

- ~~Given training data, how to find F(x,y)~~

Structured Linear Model: Problem 1

- Evaluation: What does F(x,y) look like?

Characteristics

$$\phi_1(x,y)$$
$$\phi_2(x,y)$$
$$\phi_3(x,y)$$
$$\vdots$$

$$\mathrm{F}(x,y) = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \phi_1(x,y) \\ \phi_2(x,y) \\ \phi_3(x,y) \\ \vdots \end{bmatrix}$$

w $\quad$ $\phi(x,y)$

$$\mathrm{F}(x,y) = w_1 \cdot \phi_1(x,y) + w_2 \cdot \phi_2(x,y) + w_3 \cdot \phi_3(x,y) \ldots$$

Learning from data

$$\mathrm{F}(x,y) = w \cdot \phi(x,y)$$

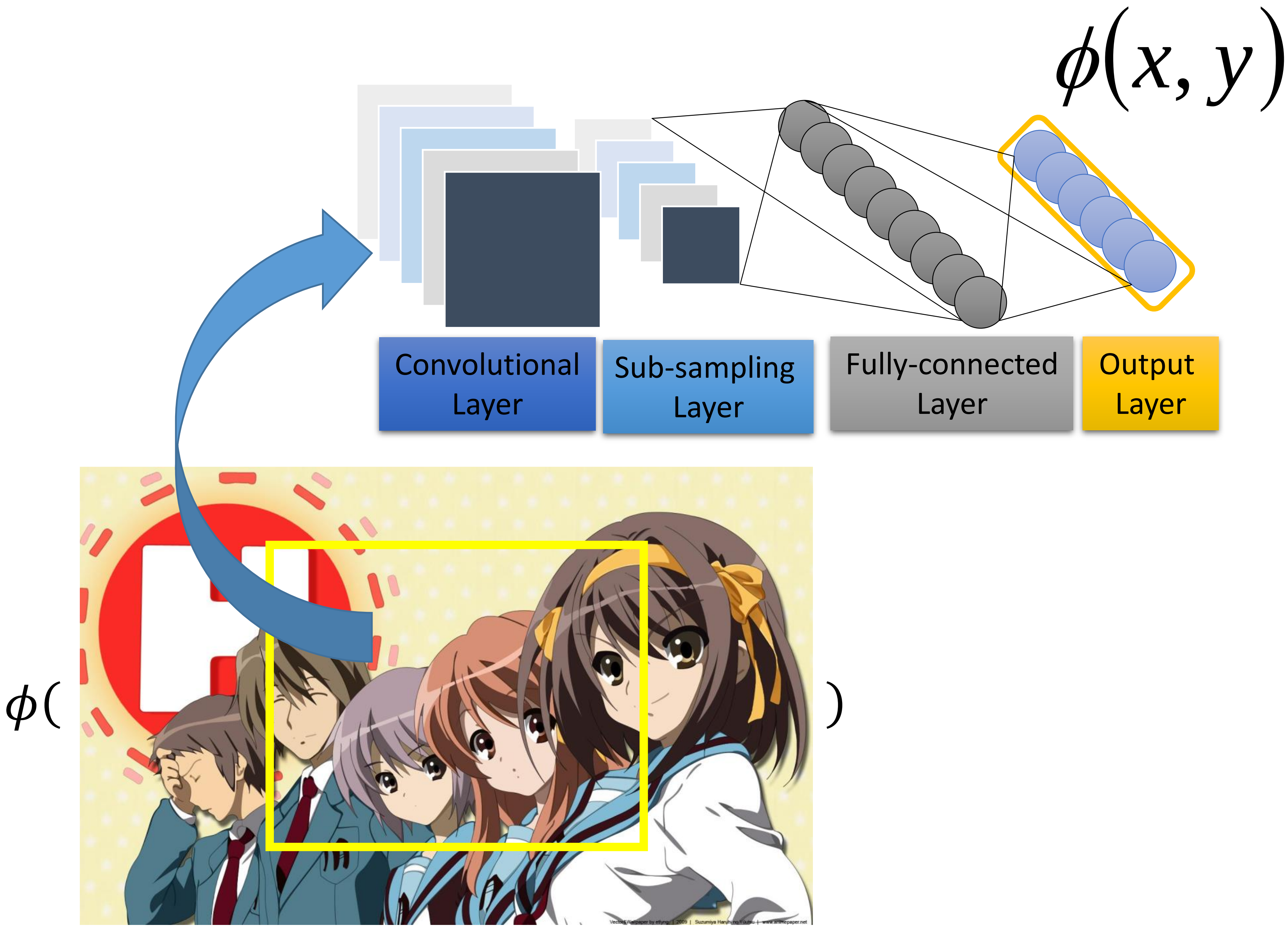
Structured Linear Model: Problem 1

- Evaluation: What does F(x,y) look like?
- Example: ***Object Detection***

x y

$\phi($  $) =$

$$\begin{bmatrix} \text{percentage of color red in box y} \\ \text{percentage of color green in box y} \\ \text{percentage of color blue in box y} \\ \text{percentage of color red out of box y} \\ \cdots\cdots \\ \text{area of box y} \\ \text{number of specific patterns in box y} \\ \cdots\cdots \end{bmatrix}$$


$\phi(x, y)$
Convolutional Layer
Sub-sampling Layer
Fully-connected Layer
Output Layer
$\phi($ $)$

Structured Linear Model: Problem 2

- **Inference**: How to solve the "arg max" problem

$$y = \arg\max_{y \in Y} \mathrm{F}(x, y)$$

$$\mathrm{F}(x, y) = w \cdot \phi(x, y) \Rightarrow y = \arg\max_{y \in Y} \mathrm{w} \cdot \phi(x, y)$$

- Assume we have solved this question.

Structured Linear Model: Problem 3

- Training: Given training data, how to learn F(x,y)
  - F(x,y) = w·φ(x,y), so what we have to learn is w

Training data: $\left\{\left(x^1, \hat{y}^1\right), \left(x^2, \hat{y}^2\right), \ldots, \left(x^r, \hat{y}^r\right), \ldots\right\}$

We should find w such that

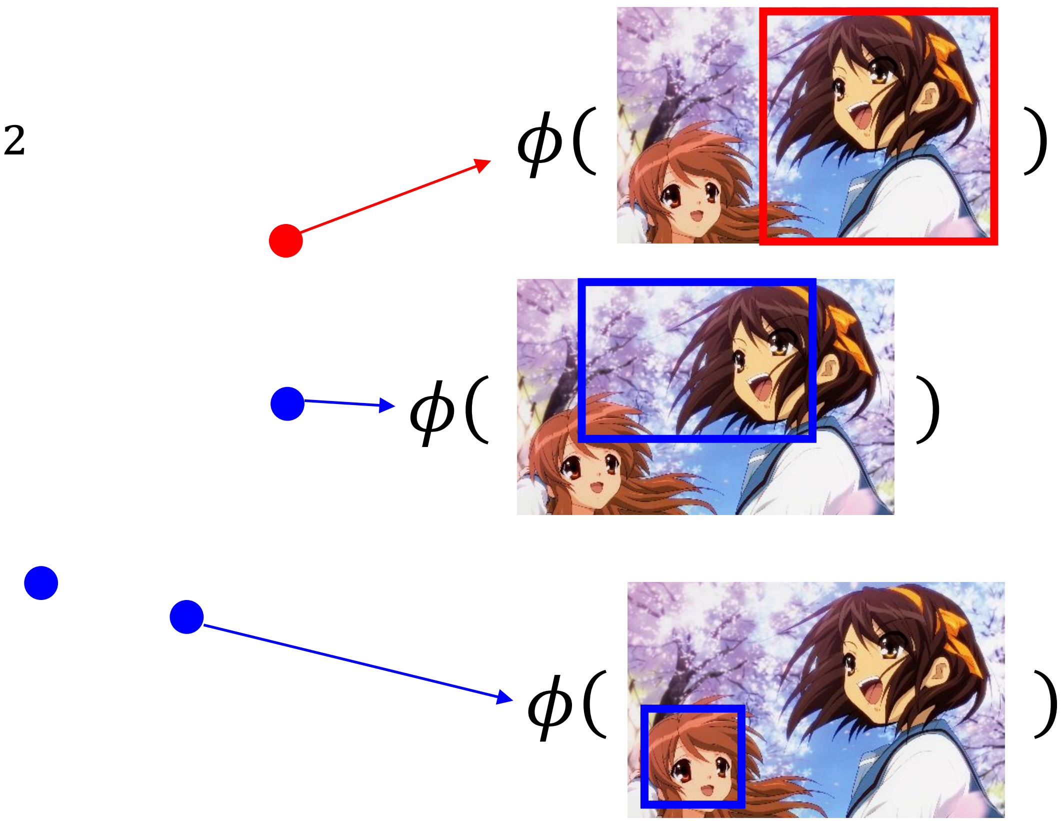
$\forall r$ (All training examples)

$\forall y \in Y - \{\hat{y}^r\}$ (All incorrect label for r-th example)

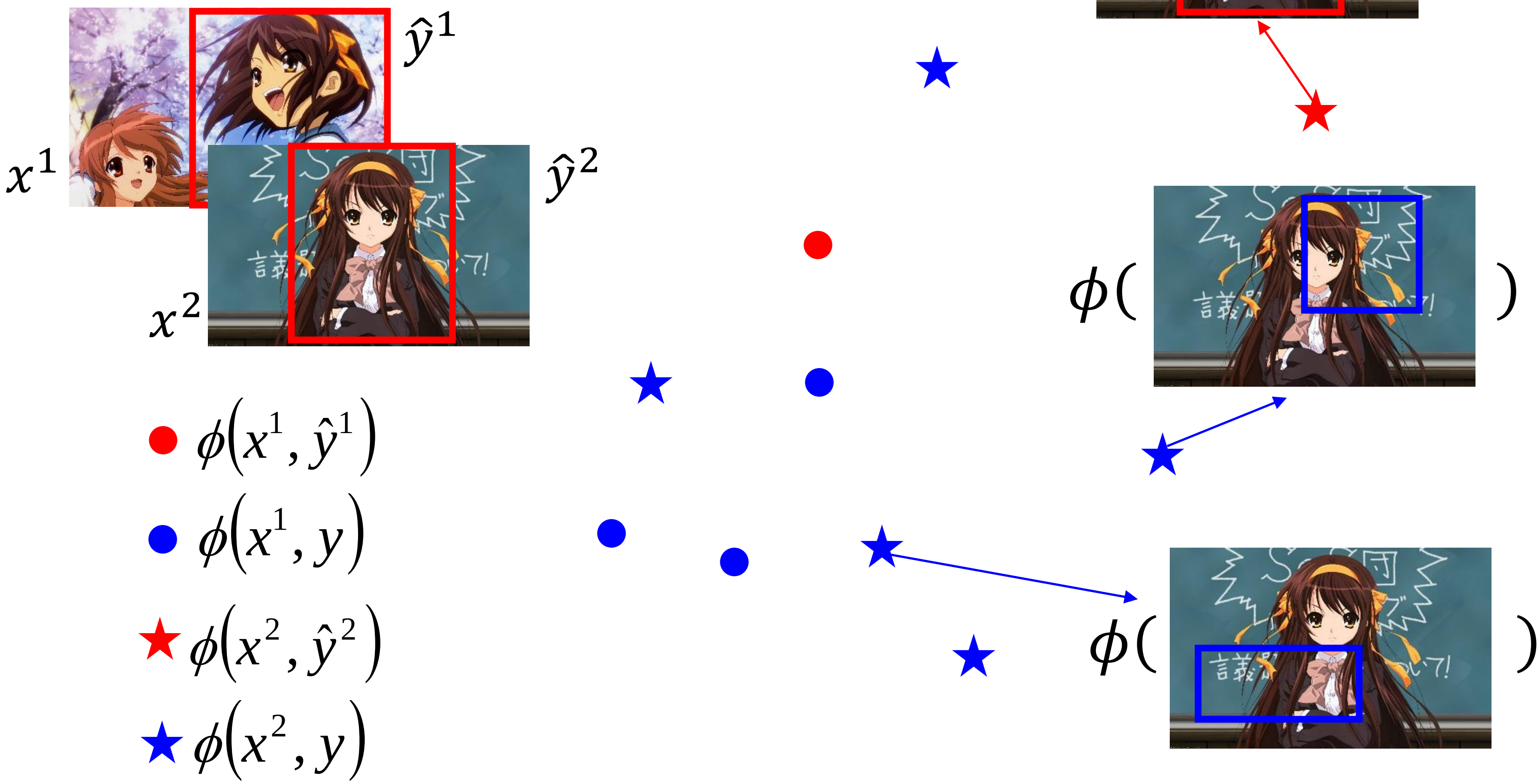
$$w \cdot \phi\left(x^r, \hat{y}^r\right) > w \cdot \phi\left(x^r, y\right)$$

Structured Linear Model: Problem 3

- $\phi(x^1, \hat{y}^1)$
- $\phi(x^1, y)$

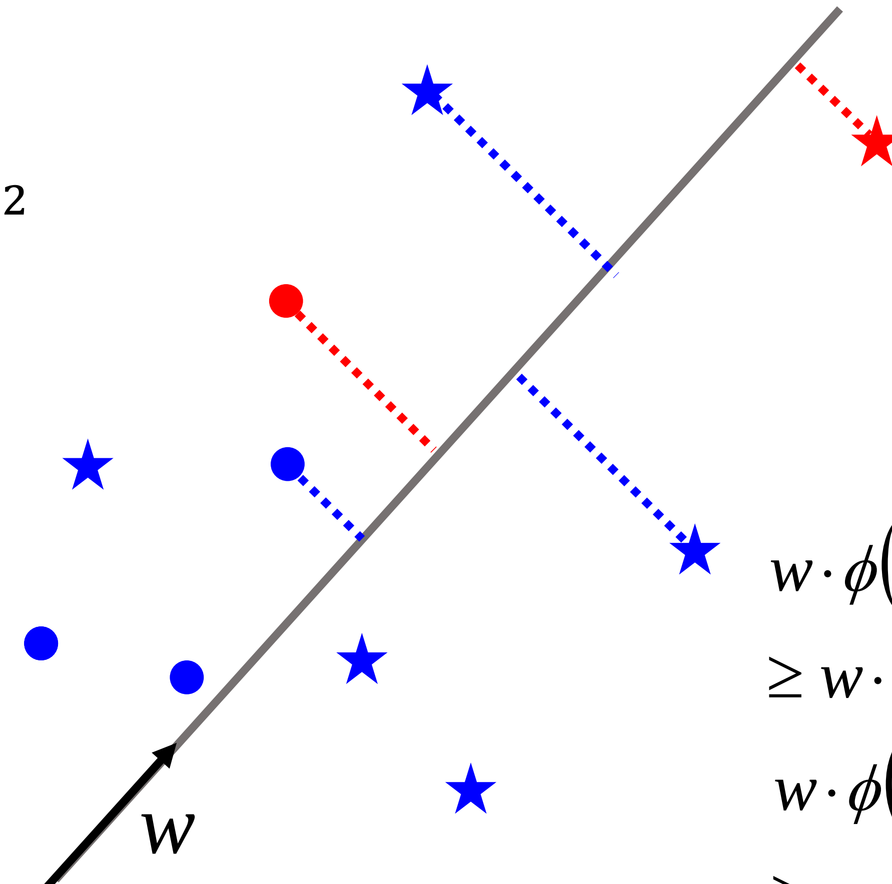


Structured Linear Model: Problem 3



Structured Linear Model: Problem 3

- ● (red) $\phi(x^1, \hat{y}^1)$
- ● (blue) $\phi(x^1, y)$
- ★ (red) $\phi(x^2, \hat{y}^2)$
- ★ (blue) $\phi(x^2, y)$



$$w \cdot \phi(x^1, \hat{y}^1) \geq w \cdot \phi(x^1, y)$$

$$w \cdot \phi(x^2, \hat{y}^2) \geq w \cdot \phi(x^2, y)$$

Solution of Problem 3

Difficult?

Not as difficult as expected

Algorithm

Will it terminate?

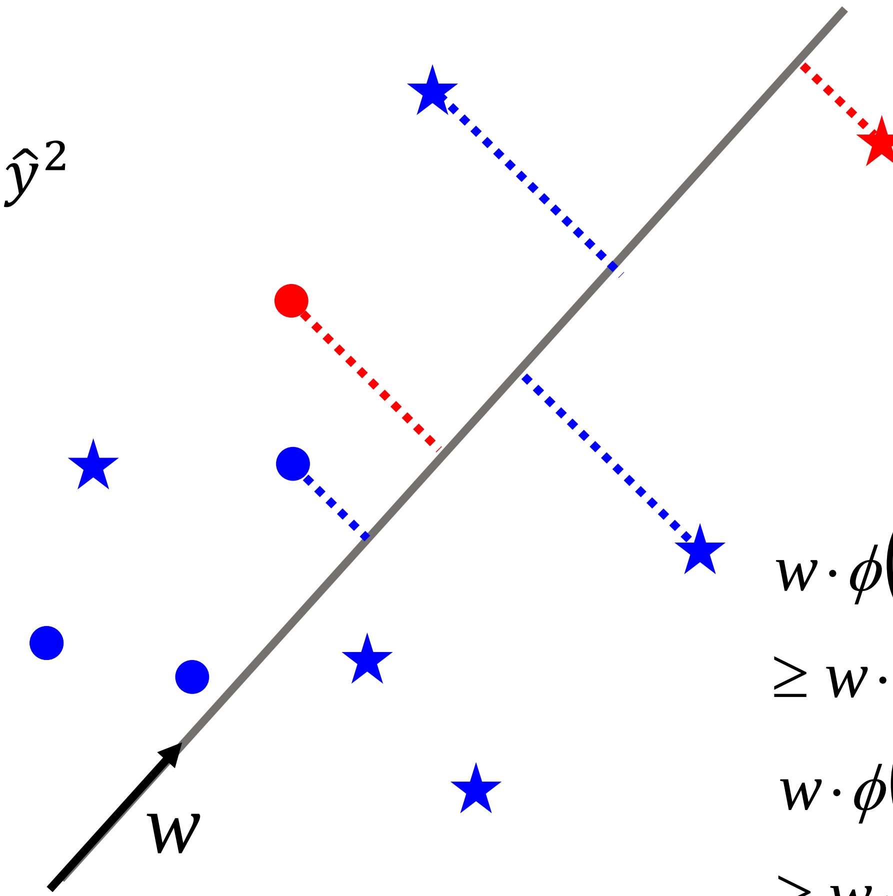
- **Input**: training data set $\{(x^1,\hat{y}^1),(x^2,\hat{y}^2),\ldots,(x^r,\hat{y}^r),\ldots\}$
- **Output**: weight vector w
- **Algorithm**: Initialize w = 0
  - do
    - For each pair of training example $(x^r,\hat{y}^r)$
      - Find the label $\tilde{y}^r$ maximizing $w \cdot \phi(x^r, y)$

        $$\tilde{y}^r = \arg\max_{y \in Y} w \cdot \phi(x^r, y)$$ (question 2)
      - If $\tilde{y}^r \neq \hat{y}^r$, update w

        $$w \rightarrow w + \phi(x^r,\hat{y}^r) - \phi(x^r,\tilde{y}^r)$$
  - until w is not updated ➡ We are done!

Algorithm - Example

- ● (red) $\phi(x^1, \hat{y}^1)$
- ● (blue) $\phi(x^1, y)$
- ★ (red) $\phi(x^2, \hat{y}^2)$
- ★ (blue) $\phi(x^2, y)$



$$w \cdot \phi(x^1, \hat{y}^1) \geq w \cdot \phi(x^1, y)$$

$$w \cdot \phi(x^2, \hat{y}^2) \geq w \cdot \phi(x^2, y)$$

Algorithm - Example

- ● (red) $\phi\left(x^1, \hat{y}^1\right)$
- ● (blue) $\phi\left(x^1, y\right)$
- ★ (red) $\phi\left(x^2, \hat{y}^2\right)$
- ★ (blue) $\phi\left(x^2, y\right)$

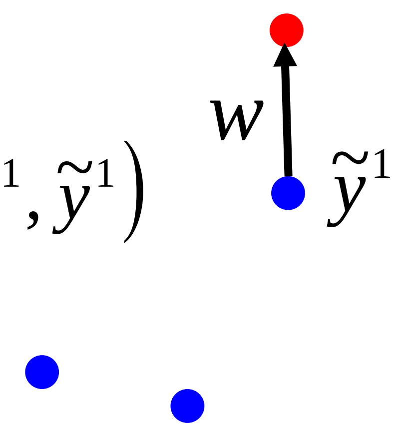
Initialize w = 0

pick $\left(x^1, \hat{y}^1\right)$

$$\tilde{y}^1 = \arg\max_{y \in Y} w \cdot \phi\left(x^1, y\right)$$

If $\tilde{y}^1 \neq \hat{y}^1$, update w

$$w \rightarrow w + \phi\left(x^1, \hat{y}^1\right) - \phi\left(x^1, \tilde{y}^1\right)$$



Because w=0 at this time, $\phi(x^1, y)$ always 0

➡ Random pick one point as $\tilde{y}^r$

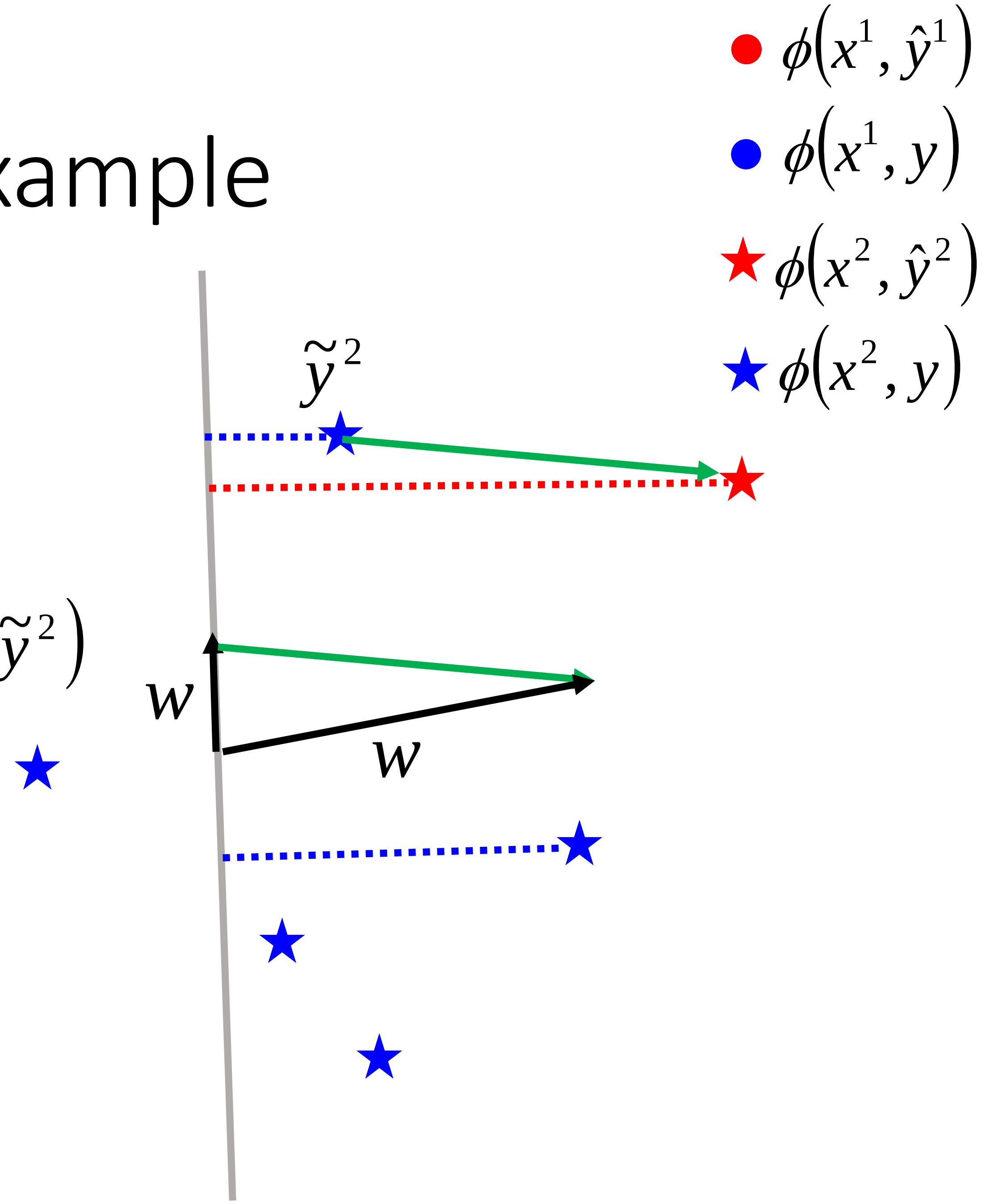
Algorithm - Example

pick $\left(x^2, \hat{y}^2\right)$

$$\tilde{y}^2 = \arg\max_{y \in Y} w \cdot \phi\left(x^2, y\right)$$

If $\tilde{y}^2 \neq \hat{y}^2$, update w

$$w \rightarrow w + \phi\left(x^2, \hat{y}^2\right) - \phi\left(x^2, \tilde{y}^2\right)$$



Algorithm - Example

$\bullet\ \phi(x^1, \hat{y}^1)$

$\bullet\ \phi(x^1, y)$

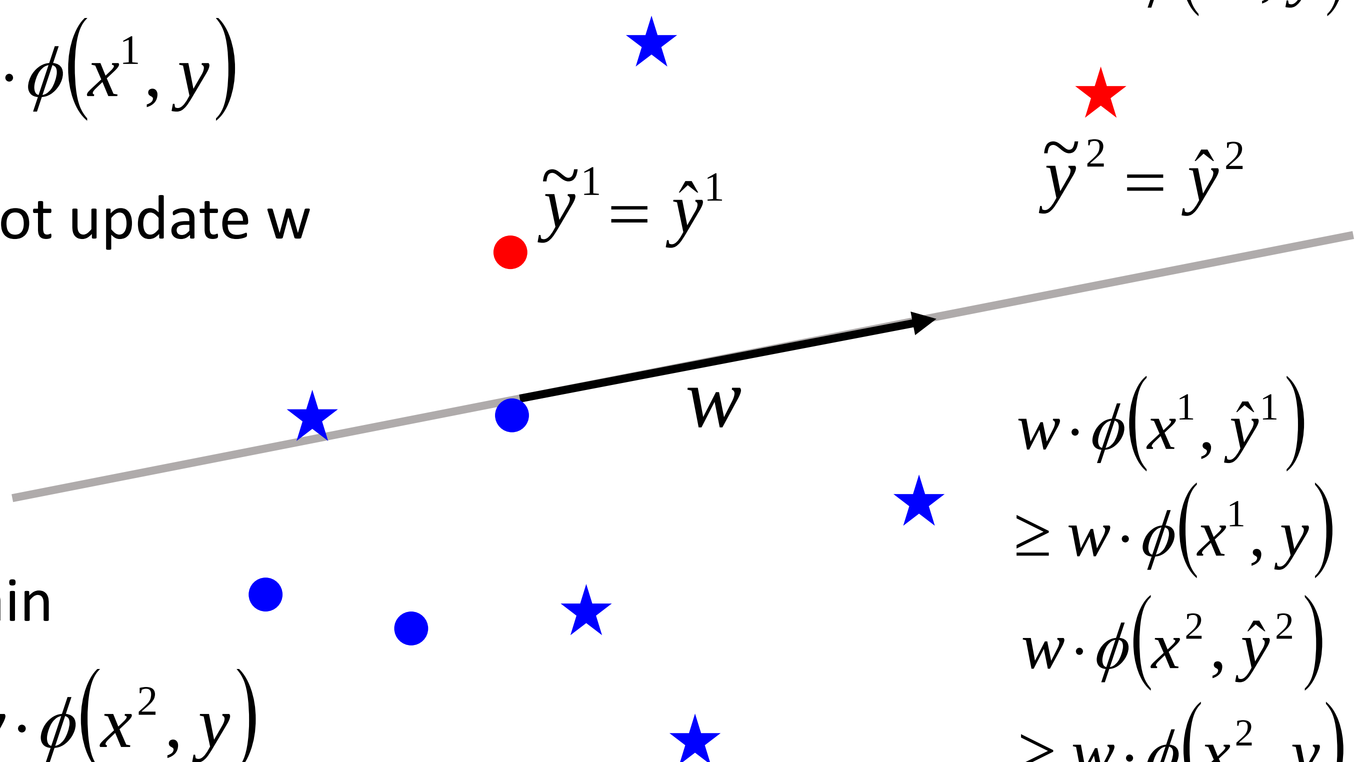
$\star\ \phi(x^2, \hat{y}^2)$

$\star\ \phi(x^2, y)$

pick $(x^1, \hat{y}^1)$ again

$$\tilde{y}^1 = \arg\max_{y \in Y} w \cdot \phi(x^1, y)$$

$\tilde{y}^1 = \hat{y}^1$ ➡ do not update w



$w \cdot \phi(x^1, \hat{y}^1)$
$\geq w \cdot \phi(x^1, y)$

$w \cdot \phi(x^2, \hat{y}^2)$
$\geq w \cdot \phi(x^2, y)$

pick $(x^2, \hat{y}^2)$ again

$$\tilde{y}^2 = \arg\max_{y \in Y} w \cdot \phi(x^2, y)$$

$\tilde{y}^2 = \hat{y}^2$ ➡ do not update w

So we are done

Assumption: Separable

- There exists a weight vector $\widehat{w}$ $\qquad \|\hat{w}\| = 1$

$\forall r$ (All training examples)

$\forall y \in Y - \{\hat{y}^r\}$ (All incorrect label for an example)

$$\hat{w} \cdot \phi(x^r, \hat{y}^r) \geq \hat{w} \cdot \phi(x^r, y)$$ (The target exists)

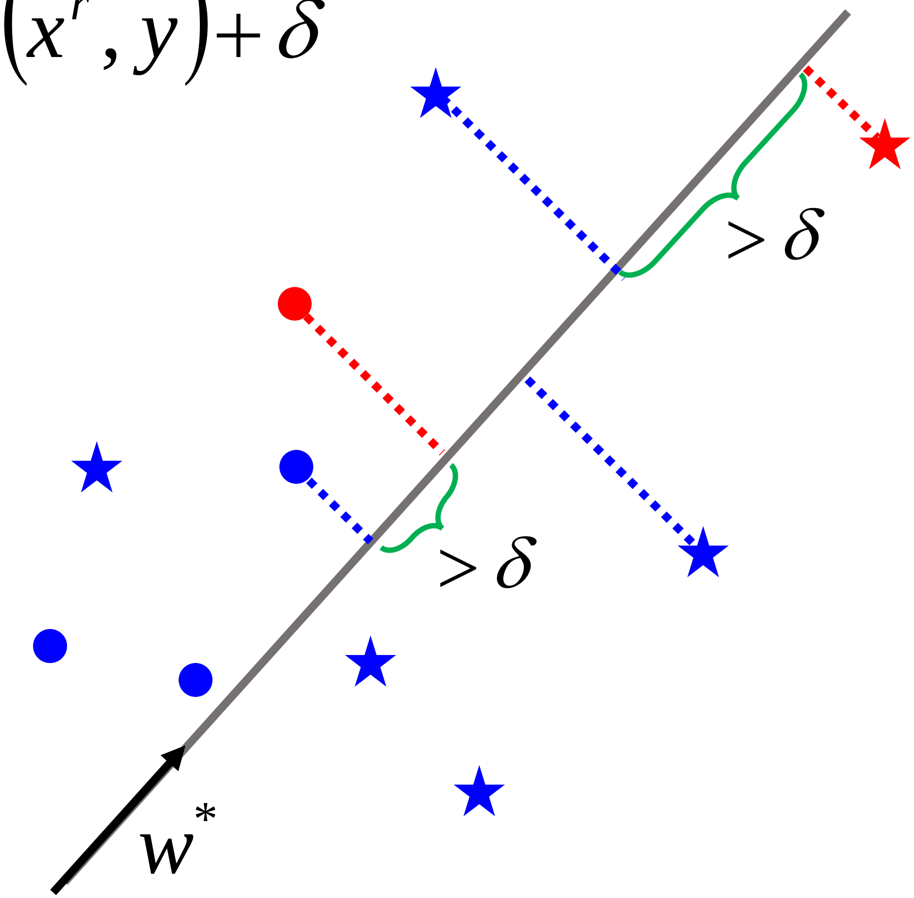
$$\hat{w} \cdot \phi(x^r, \hat{y}^r) \geq \hat{w} \cdot \phi(x^r, y) + \delta$$

Assumption: Separable

$$\hat{w} \cdot \phi\left(x^r, \hat{y}^r\right) \geq \hat{w} \cdot \phi\left(x^r, y\right) + \delta$$

- ● $\phi\left(x^1, \hat{y}^1\right)$
- ● $\phi\left(x^1, y\right)$
- ★ $\phi\left(x^2, \hat{y}^2\right)$
- ★ $\phi\left(x^2, y\right)$

……



Proof of Termination

w is updated once it sees a mistake

$$w^0 = 0 \rightarrow w^1 \rightarrow w^2 \rightarrow \ldots\ldots \rightarrow w^k \rightarrow w^{k+1} \rightarrow \ldots\ldots$$

$$w^k = w^{k-1} + \phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)$$ (the relation of w^k and w^{k-1})

Proof that: The angle ρ_k between $\hat{w}$ and w_k is smaller as k increases

Analysis $\cos\rho_k$ (larger and larger?) $\cos\rho_k = \frac{\hat{w}}{\|\hat{w}\|} \cdot \frac{w^k}{\|w^k\|}$

$$\hat{w} \cdot w^k = \hat{w} \cdot \left(w^{k-1} + \phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)\right)$$

$$= \hat{w} \cdot w^{k-1} + \underline{\hat{w} \cdot \phi(x^n, \hat{y}^n) - \hat{w} \cdot \phi(x^n, \tilde{y}^n)} \geq \hat{w} \cdot w^{k-1} + \delta$$

$\geq \delta$ (Separable)

Proof of Termination

w is updated once it sees a mistake

$$w^0 = 0 \rightarrow w^1 \rightarrow w^2 \rightarrow \ldots\ldots \rightarrow w^k \rightarrow w^{k+1} \rightarrow \ldots\ldots$$

$$w^k = w^{k-1} + \phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)$$ (the relation of w^k and w^{k-1})

Proof that: The angle ρ_k between $\hat{w}$ and w_k is smaller as k increases

Analysis $\cos \rho_k$ (larger and larger?)

$$\cos \rho_k = \frac{\hat{w}}{\|\hat{w}\|} \cdot \frac{w^k}{\|w^k\|}$$

$$\hat{w} \cdot w^k \geq \hat{w} \cdot w^{k-1} + \delta$$

=0 ≥δ

$$\hat{w} \cdot w^1 \geq \hat{w} \cdot w^0 + \delta \qquad \hat{w} \cdot w^2 \geq \hat{w} \cdot w^1 + \delta \quad \ldots\ldots$$

$$\hat{w} \cdot w^1 \geq \delta \qquad \hat{w} \cdot w^2 \geq 2\delta \quad \ldots\ldots$$

$$\hat{w} \cdot w^k \geq k\delta$$

(so what)

Proof of Termination

$$\cos\rho_k = \frac{\hat{w}}{\|\hat{w}\|} \cdot \frac{w^k}{\|w^k\|} \qquad w^k = w^{k-1} + \phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)$$

$$\|w^k\|^2 = \left\|w^{k-1} + \phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)\right\|^2$$

$$= \|w^{k-1}\|^2 + \underbrace{\left\|\phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)\right\|^2}_{>0} + \underbrace{2w^{k-1} \cdot \left(\phi(x^n, \hat{y}^n) - \phi(x^n, \tilde{y}^n)\right)}_{?\ <0 \text{ (mistake)}}$$

Assume the distance between any two feature vector is smaller than R

$$\le \|w^{k-1}\| + R^2$$

$$\|w^1\|^2 \le \|w^0\|^2 + R^2 = R^2$$

$$\|w^2\|^2 \le \|w^1\|^2 + R^2 \le 2R^2$$

$$\cdots$$

$$\|w^k\|^2 \le kR^2$$

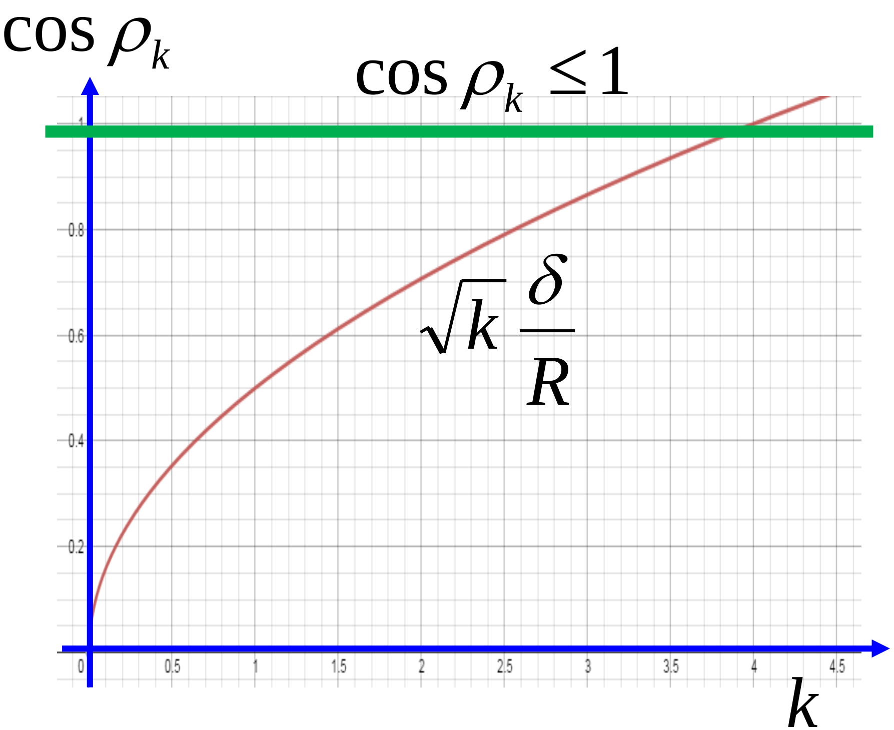
Proof of Termination

$$\cos\rho_k = \frac{\hat{w}}{\|\hat{w}\|}\cdot\frac{w^k}{\|w^k\|} \qquad \hat{w}\cdot w^k \ge k\delta \qquad \|w^k\|^2 \le k\mathrm{R}^2$$

$$\ge \frac{k\delta}{\sqrt{kR^2}} = \sqrt{k}\,\frac{\delta}{R}$$

$$\sqrt{k}\,\frac{\delta}{R} \le 1$$

$$k \le \left(\frac{R}{\delta}\right)^2$$

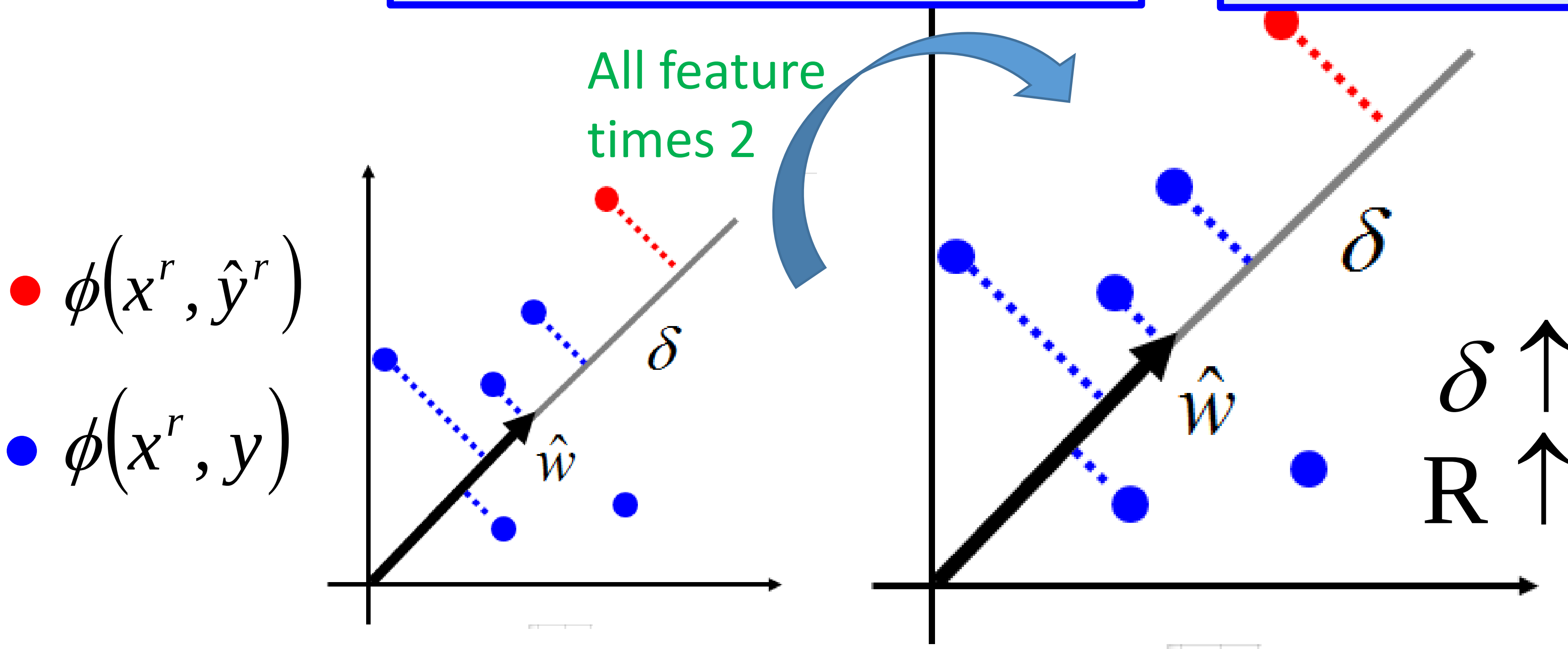


Proof of Termination

$$k \leq \left(\frac{R}{\delta} \right)^2$$

R → The largest distances between features → Normalization

δ → Margin: Is it easy to separable red points from the blue ones → Larger margin, less update



Structured Linear Model: Reduce 3 Problems to 2

Problem 1: Evaluation

- How to define $F(x,y)$

Problem 2: Inference

- How to find the y with the largest $F(x,y)$

Problem 3: Training

- How to learn $F(x,y)$

$F(x,y)=w\cdot\phi(x,y)$

Problem A: Feature

- How to define $\phi(x,y)$

Problem B: Inference

- How to find the y with the largest $w\cdot\phi(x,y)$

Graphical Model

A language which describes the evaluation function

Structured Learning

We also know how to involve hidden information.

Problem 1: Evaluation

- What does F(x,y) look like? $F(x,y) = w \cdot \phi(x,y)$

Problem 2: Inference

- How to solve the "arg max" problem

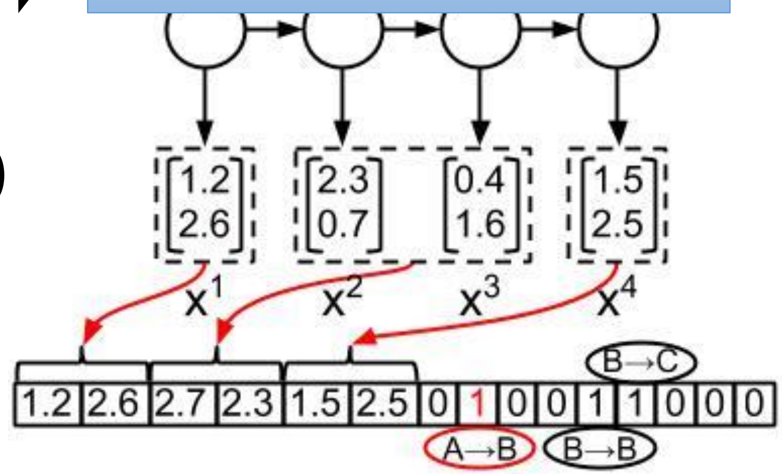
$$y = \arg\max_{y \in Y} F(x,y)$$

Problem 3: Training

- Given training data, how to find F(x,y) Structured SVM, etc.

Difficulties

Difficulty 1. Evaluation → Graphical Model



$$F(x, y) = w \cdot \phi(x, y)$$

$$\phi(x, y)$$

Hard to figure out? Hard to interpret the meaning?

Difficulty 2. Inference → Gibbs Sampling

We can use Viterbi algorithm to deal with sequence labeling. How about other cases?

Graphical Model

$F(x, y)$ ⟷ Graph

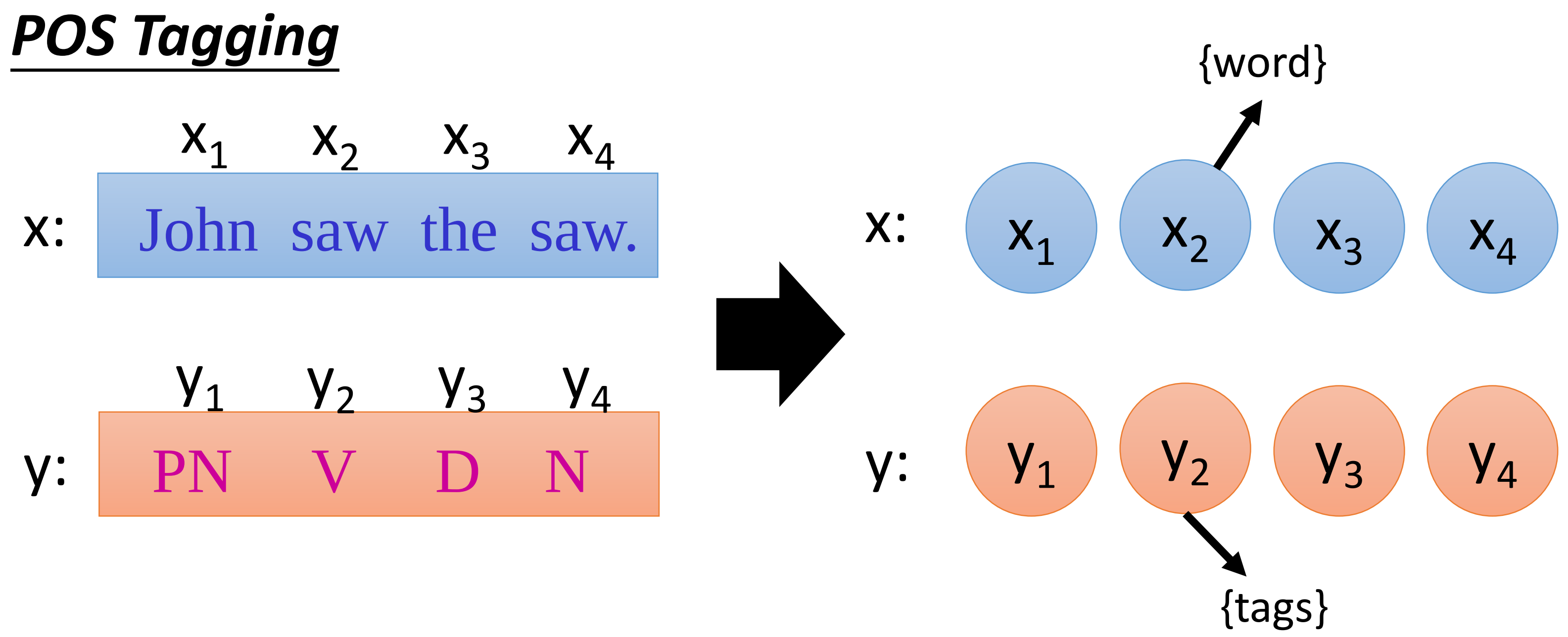
- Define and describe your evaluation function F(x,y) by a graph
- There are three kinds of graphical model.
  - *Factor graph*, *Markov Random Field* (MRF) and *Bayesian Network* (BN)
  - Only *factor graph* and *MRF* will be briefly mentioned today.

Decompose F(x,y)

- $F(x, y)$ is originally a ***global*** function
  - Define over the whole x and y
- Based on graphical model, $F(x, y)$ is the composition of some ***local*** functions
  - x and y are decomposed into smaller components
  - Each local function defines on only a few related components in x and y
  - Which components are related → defined by Graphical model

Decomposable x and y

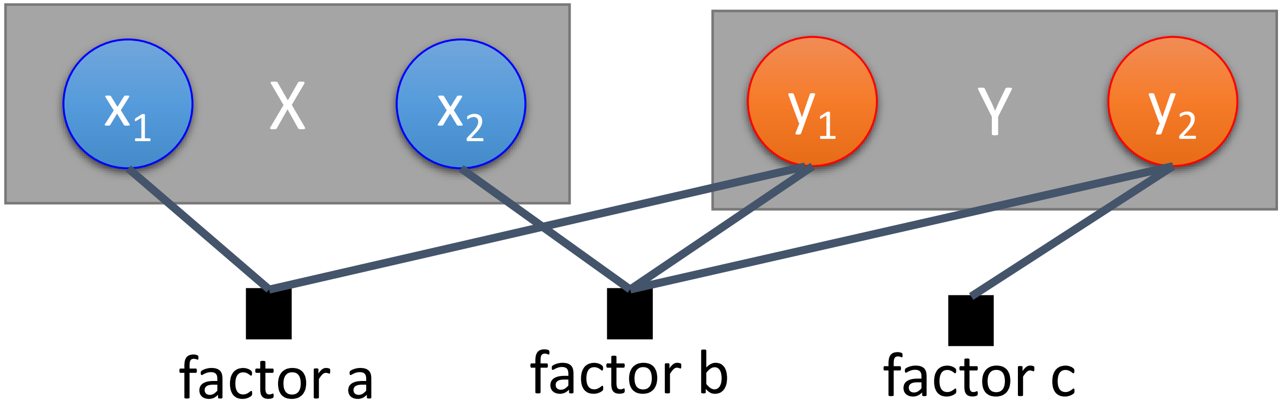
- x and y are decomposed into smaller components



Factor Graph

Each factor influences some components.

Each factor corresponds to a local function.



$f_a(x_1, y_1)$ $f_b(x_2, y_1, y_2)$ $f_d(y_2)$

Larger value means more compatible.

$$F(x, y) = f_a(x_1, y_1) + f_b(x_2, y_1, y_2) + f_c(y_2)$$

You only have to define the factors.

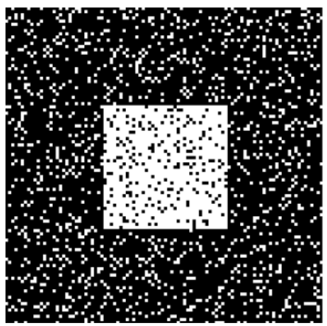
The local functions of the factors are learned from data.

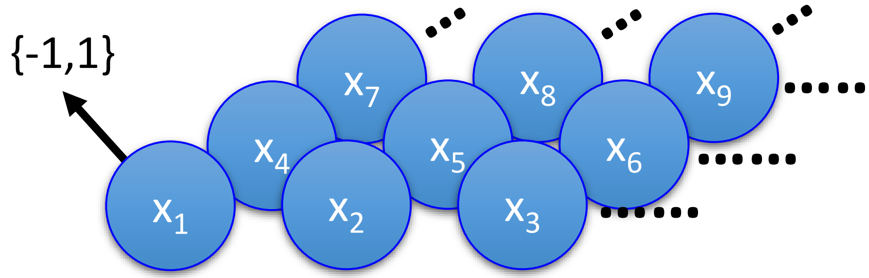
Factor Graph - Example

- ***Image De-noising***

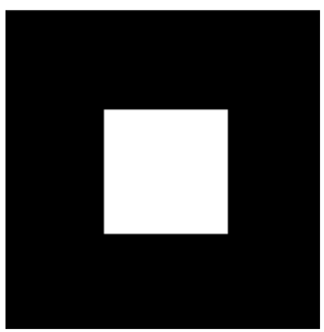
Each pixel is one component

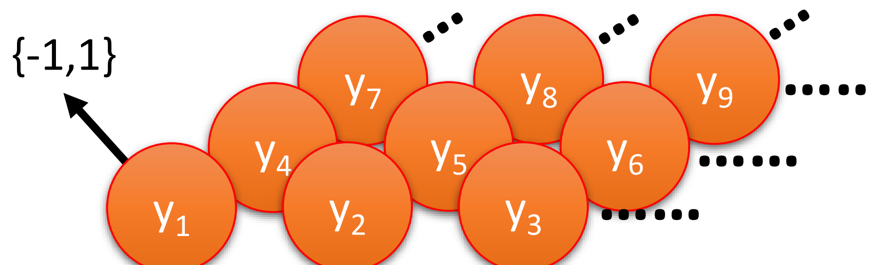
Noisy image
x





Clean image
y





http://cs.stanford.edu/people/karpathy/visml/ising_example.html

Factor Graph - Example

Factor:

Noisy and clean images are related

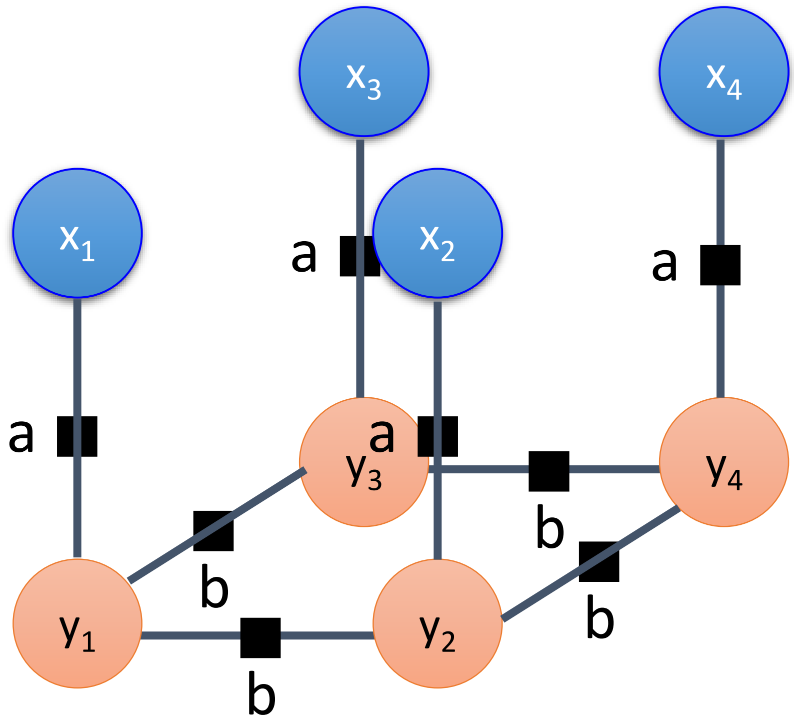
- **a**: the values of x_i and y_i

The colors in the clean image is smooth.

- **b**: the values of the neighboring y_i

noisy image

cleaned image



$$f_a(x_i, y_i) = \begin{cases} 1 & x_i = y_i \\ -1 & x_i \neq y_i \end{cases}$$

$$f_b(y_i, y_j) = \begin{cases} 2 & y_i = y_j \\ -2 & y_i \neq y_j \end{cases}$$

The weights can be learned from data.

Factor Graph - Example

Factor:

Noisy and clean images are related

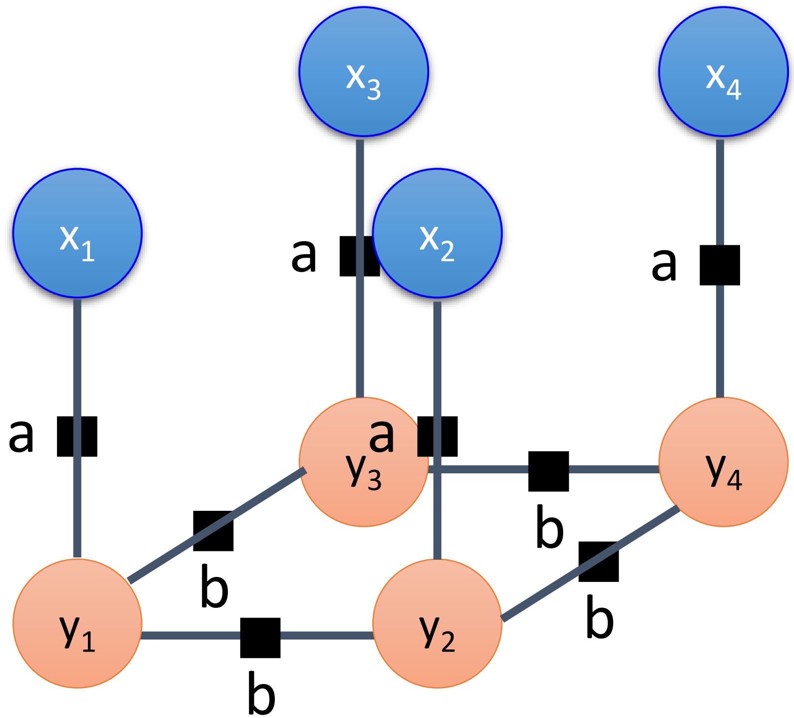
- **a**: the values of x_i and y_i

The colors in the clean image is smooth.

- **b**: the values of the neighboring y_i

noisy image

cleaned image



Realize $F(x,y)$ easily from the factor graph

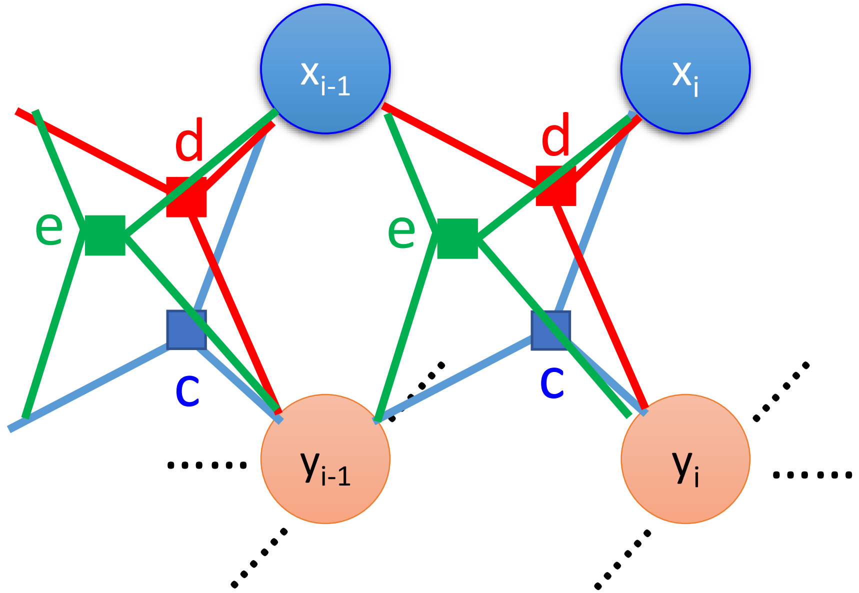
$$F(x,y) = \sum_{i=1}^{4} f_a(x_i, y_i)$$

$$+ f_b(x_1, y_2) + f_b(x_1, y_3)$$
$$+ f_b(x_2, y_4) + f_b(x_3, y_4)$$

Factor Graph - Example

Factor:

- **c**: the values of x_i and the values of the neighboring y_i
- **d**: the values of the neighboring x_i and the values of y_i



$$f_c(x_i, y_i, y_{i-1})$$

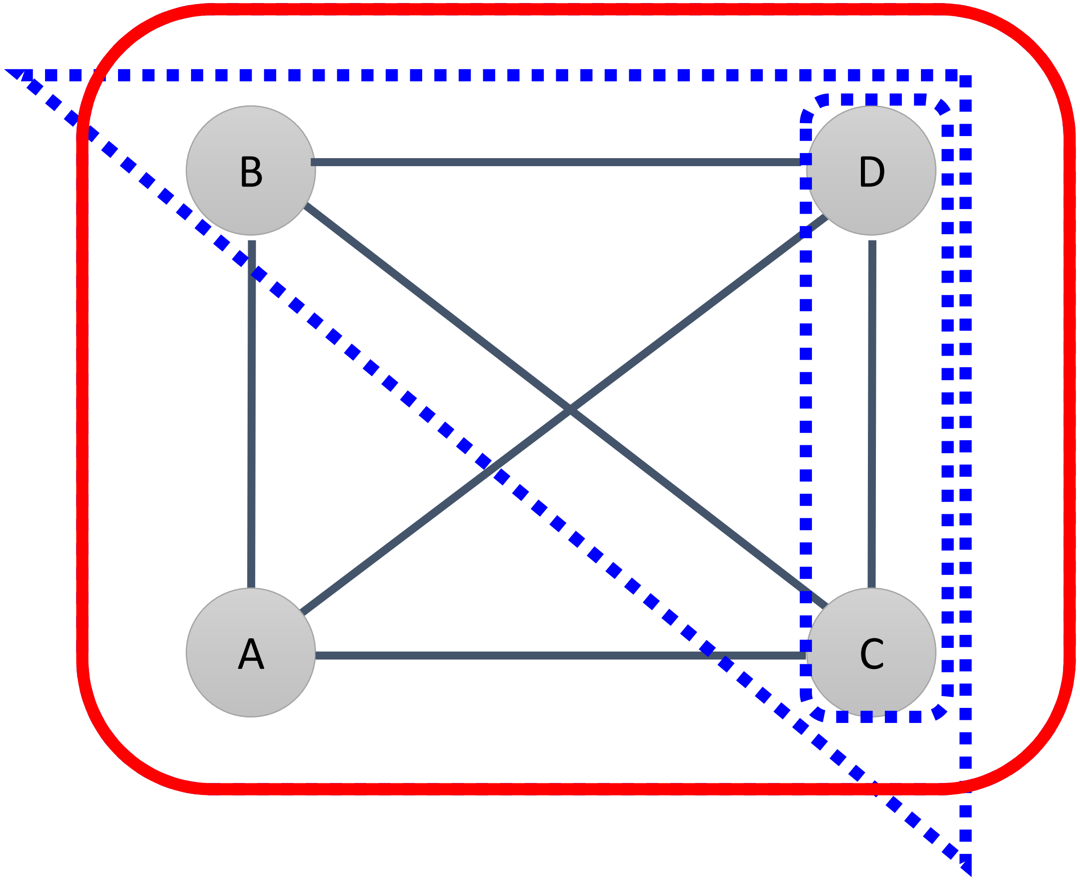
$$f_d(x_i, x_{i-1}, y_i)$$

$$f_e(x_i, x_{i-1}, y_i, y_{i-1})$$

Markov Random Field (MRF)

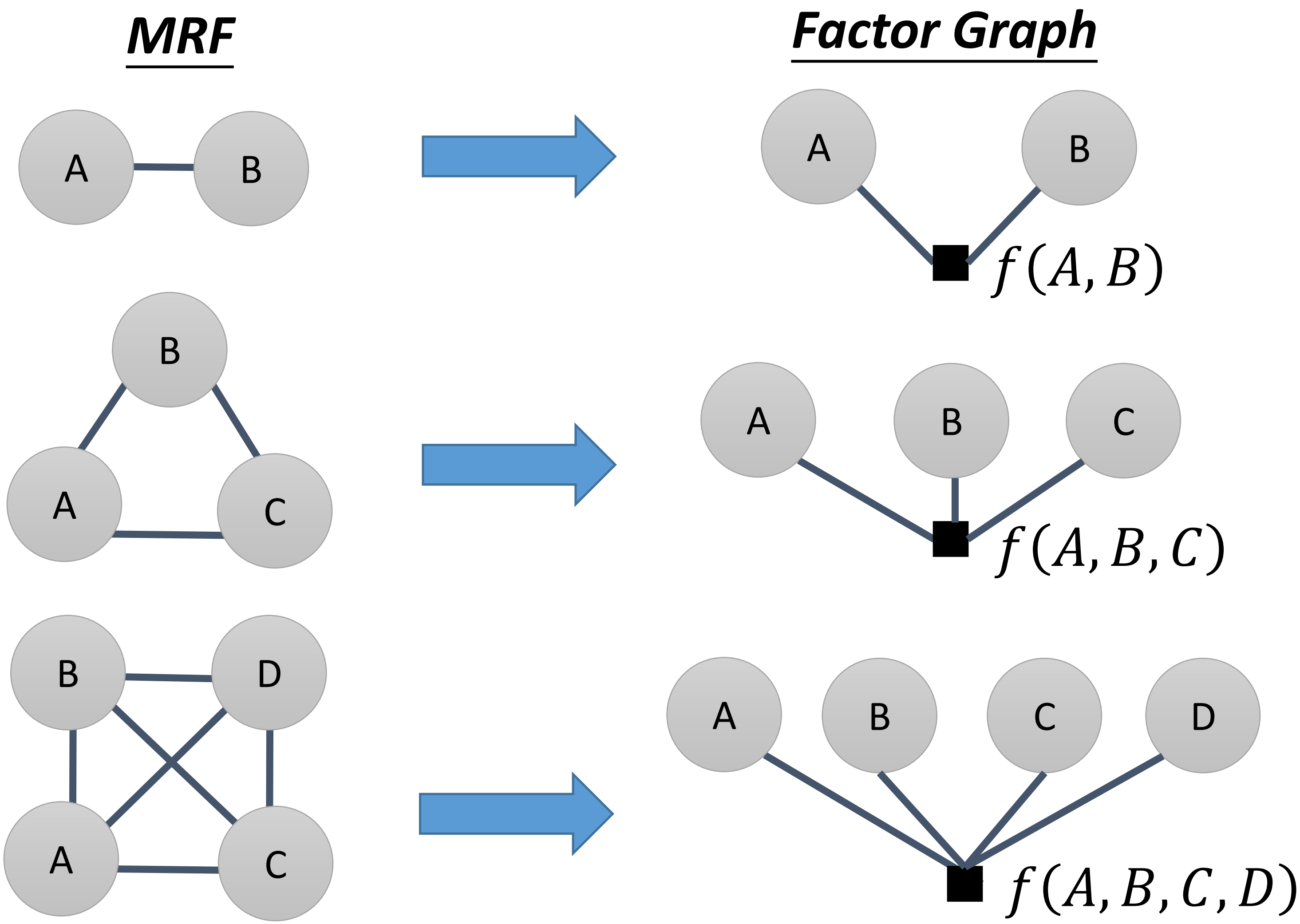
Clique: a set of components connecting to each other

Maximum Clique: a clique that is not included by other cliques

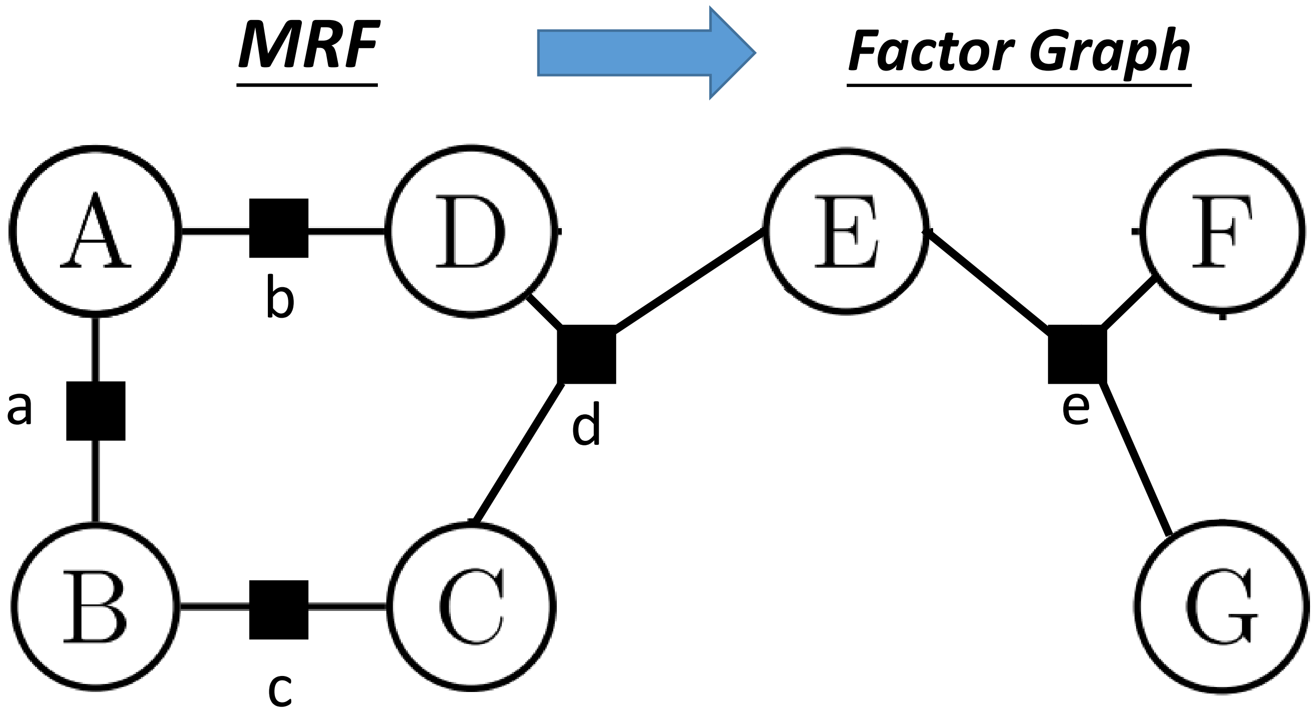


MRF

Each maximum clique on the graph corresponds to a factor



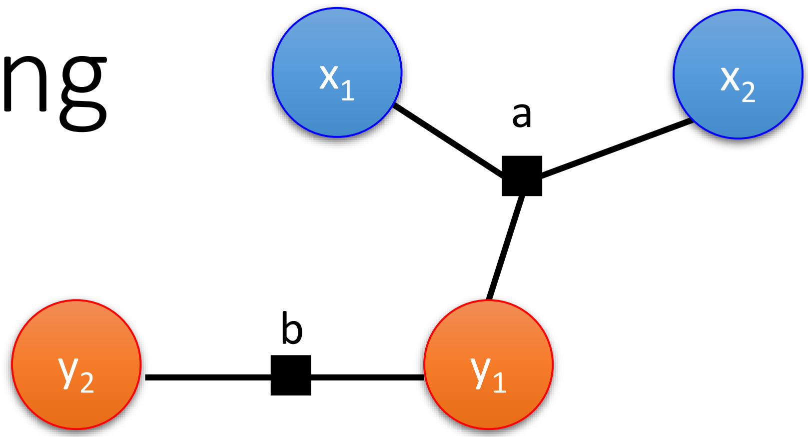
MRF



Evaluation Function

$$f_a(A,B) + f_b(A,D) + f_c(B,C) + f_d(C,D,E) + f_e(E,F,G)$$

Training



$$F(x,y) = f_a(x_1,x_2,y_1) + f_b(y_1,y_2)$$

$$= w_a \cdot \phi_a(x_1,x_2,y_1) + w_b \cdot \phi_b(y_1,y_2)$$

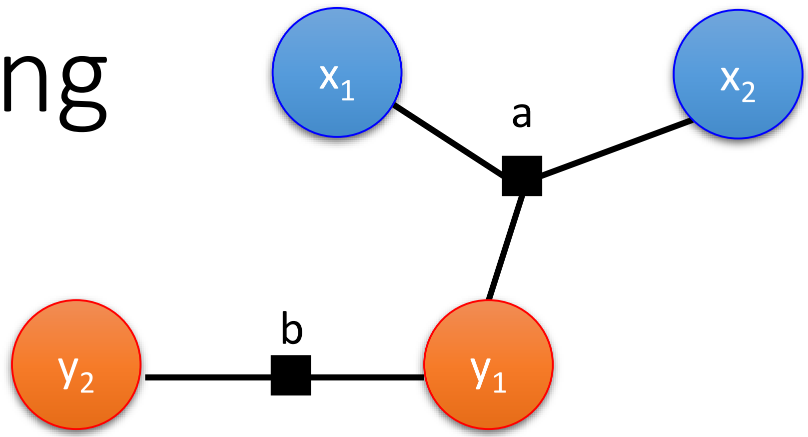
$$= \begin{bmatrix} w_a \\ w_b \end{bmatrix} \begin{bmatrix} \phi_a(x_1,x_2,y_1) \\ \phi_b(y_1,y_2) \end{bmatrix}$$

$$= w \cdot \phi(x,y)$$

Simply training by ***structured perceptron*** ***or structured SVM***

Max-Margin Markov Networks (M3N)

Training



$$F(x,y) = f_a(x_1,x_2,y_1) + f_b(y_1,y_2)$$

$$= w_a \cdot \phi_a(x_1,x_2,y_1) + w_b \cdot \phi_b(y_1,y_2)$$

$$y_1, y_2 \epsilon \{+1,-1\}$$

y_1	y_2	$f_b(y_1,y_2)$
+1	+1	w_1
+1	-1	w_2
-1	+1	w_3
-1	-1	w_4

$$w_b = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\phi_b(+1,+1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

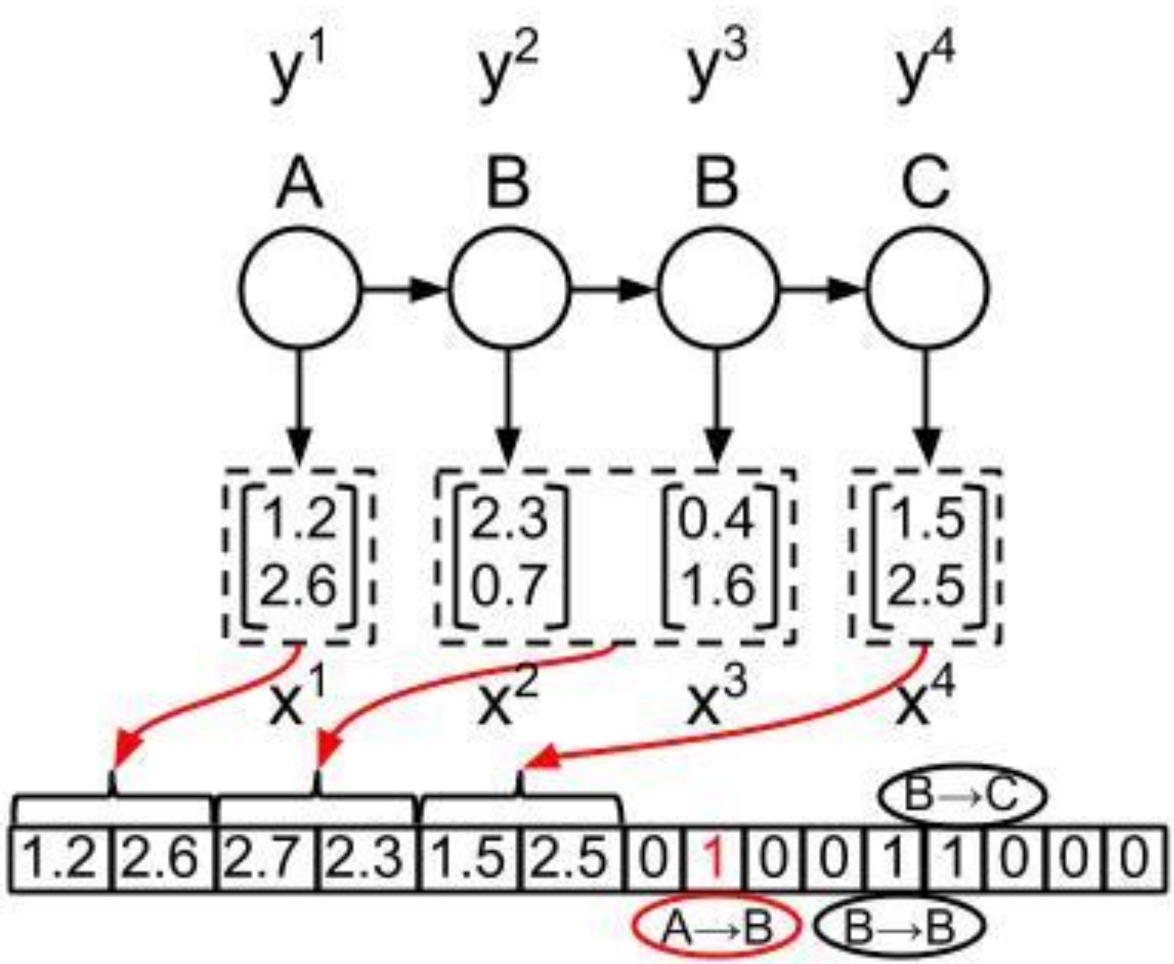
$$\phi_b(+1,-1) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\phi_b(-1,+1) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\phi_b(-1,-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Now can you interpret this?

$\phi(x, y)$



Probability Point of View

- $F(x, y)$ can be any real number
- If you like probability

Between 0 and 1

$$P(x, y) = \frac{e^{F(x,y)}}{\sum_{x',y'} e^{F(x',y')}}$$

$e^{F(x,y)}$ → To be positive

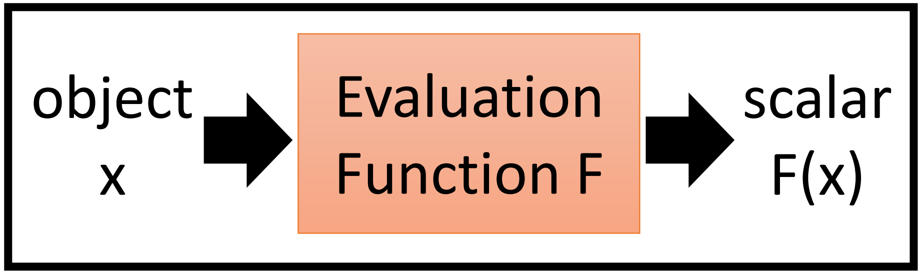
$\sum_{x',y'} e^{F(x',y')}$ → normalization

$$P(y|x) = \frac{P(x, y)}{P(x)}$$

$$= \frac{P(x, y)}{\sum_{y''} P(x, y'')} = \frac{\frac{e^{F(x,y)}}{\sum_{x',y'} e^{F(x',y')}}}{\sum_{y''} \frac{e^{F(x,y'')}}{\sum_{x',y'} e^{F(x',y')}}} = \frac{e^{F(x,y)}}{\sum_{y''} e^{F(x,y'')}}$$

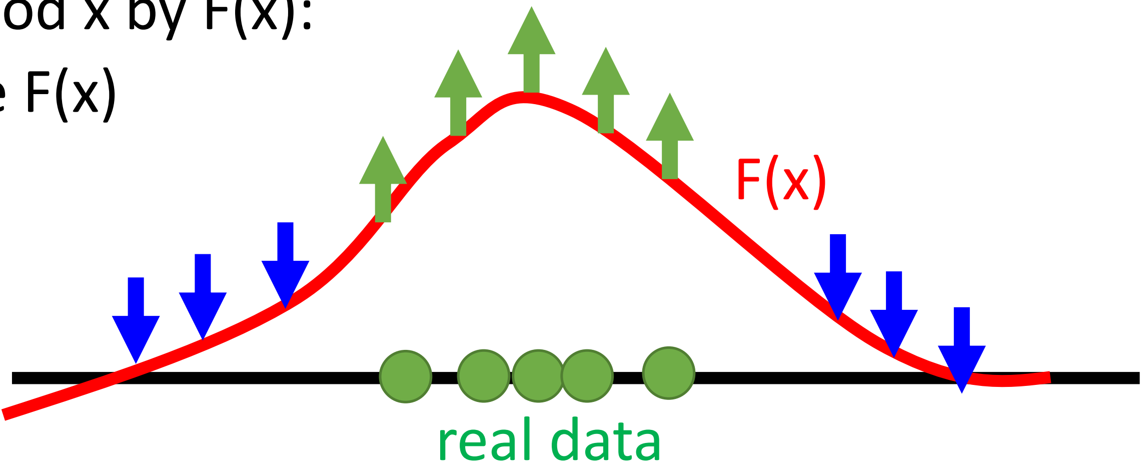
Evaluation Function

- We want to find an evaluation function F(x)
  - Input: object x, output: scalar F(x) (how "good" the object is)
  - E.g. x are images
    - Real x has high F(x)
  - F(x) can be a network



- We can generate good x by F(x):
  - Find x with large F(x)
- How to find F(x)?

In practice, you cannot decrease all the x other than real data.



Evaluation Function
- Structured Perceptron

- **Input**: training data set $\left\{\left(x^1, \hat{y}^1\right), \left(x^2, \hat{y}^2\right), \dots, \left(x^r, \hat{y}^r\right), \dots\right\}$
- **Output**: weight vector w
- **Algorithm**: Initialize w = 0
  - do
    - For each pair of training example $\left(x^r, \hat{y}^r\right)$
      - Find the label $\tilde{y}^r$ maximizing F(x^r, y)
      - If $\tilde{y}^r \neq \hat{y}^r$, update w
  - until w is not updated

$$F(x, y) = w \cdot \phi(x, y)$$

Can be an issue

$$\tilde{y}^r = \arg\max_{y \in Y} F\left(x^r, y\right)$$

Increase $F(x^r, \hat{y}^r)$,
decrease $F(x^r, \tilde{y}^r)$

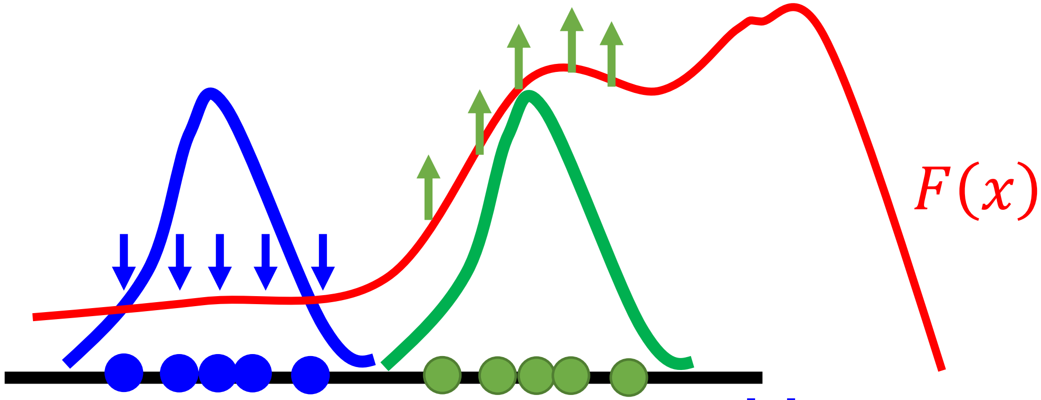
$$w \rightarrow w + \phi\left(x^r, \hat{y}^r\right) - \phi\left(x^r, \tilde{y}^r\right)$$

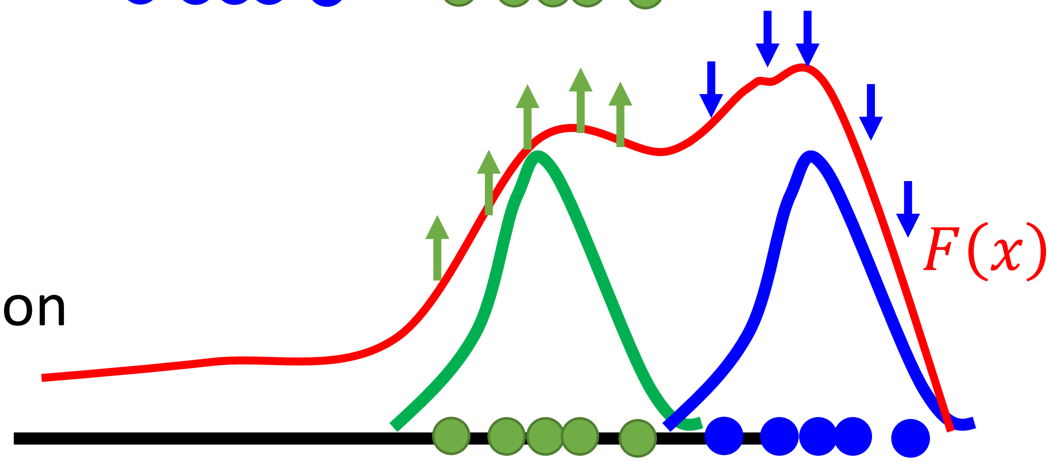
We are done!

How about GAN?

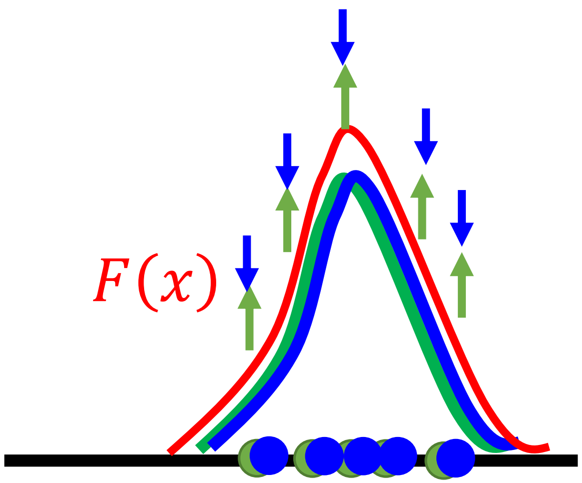
- Generator is an intelligent way to find the negative examples.

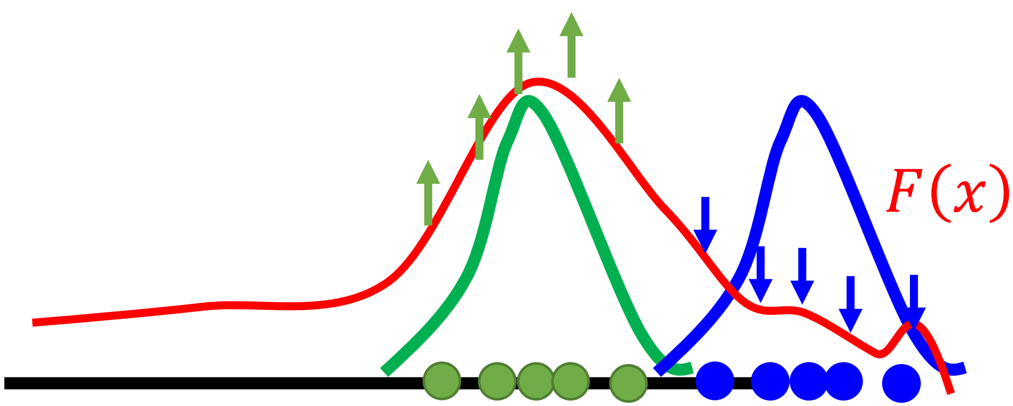
"Experience replay", parameters from last iteration



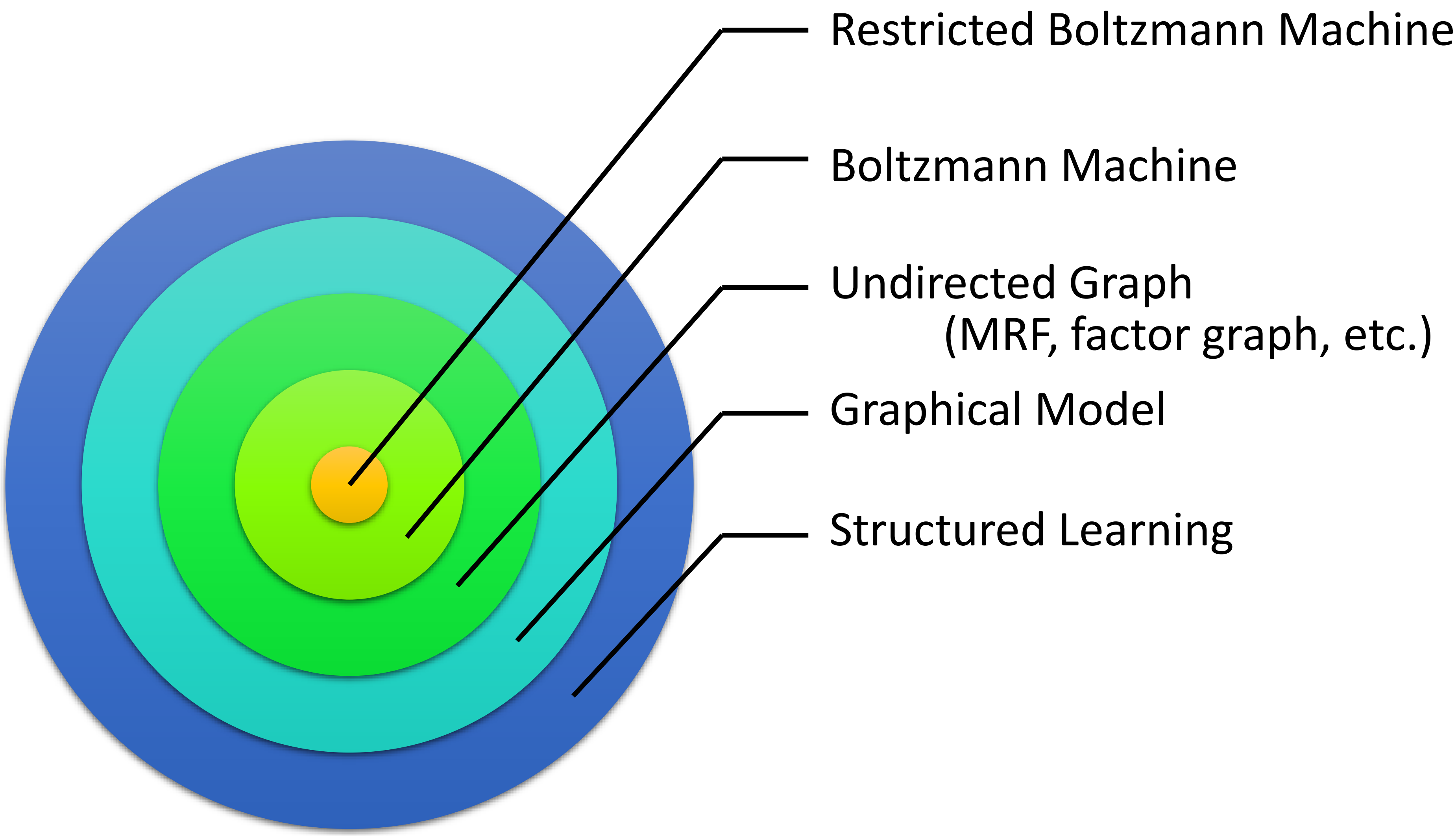


In the end ……





Where are we?


Restricted Boltzmann Machine
Boltzmann Machine
Undirected Graph
(MRF, factor graph, etc.)
Graphical Model
Structured Learning