Learning with Hidden Information

Hung-yi Lee

Different Kinds of Learning

- Supervised Learning
  - Data: $\{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots\}$
- Semi-supervised Learning
  - Data: $\{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots, (x^{N+1}, ?), (x^{N+2}, ?) \cdots\}$
- Unsupervised Learning
  - Data: $\{(x^1, ?), (x^2, ?), \cdots\}$
- Hidden variable learning
  - Data: $\{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots\}$

Some useful information is hidden.

Outline

- Example Applications for Hidden Variable Learning
- General Framework
- Structured SVM with Hidden Information
- Verifying the correctness

Example Applications for Hidden Variable Learning

Example Applications

- ***Sentiment Analysis***: Automatically identify a movie review is positive or negative

Collecting documents about reviewing movies

x^1: 看了這部電影覺得很高興……. → $\hat{y}^1$: Positive (正雷)

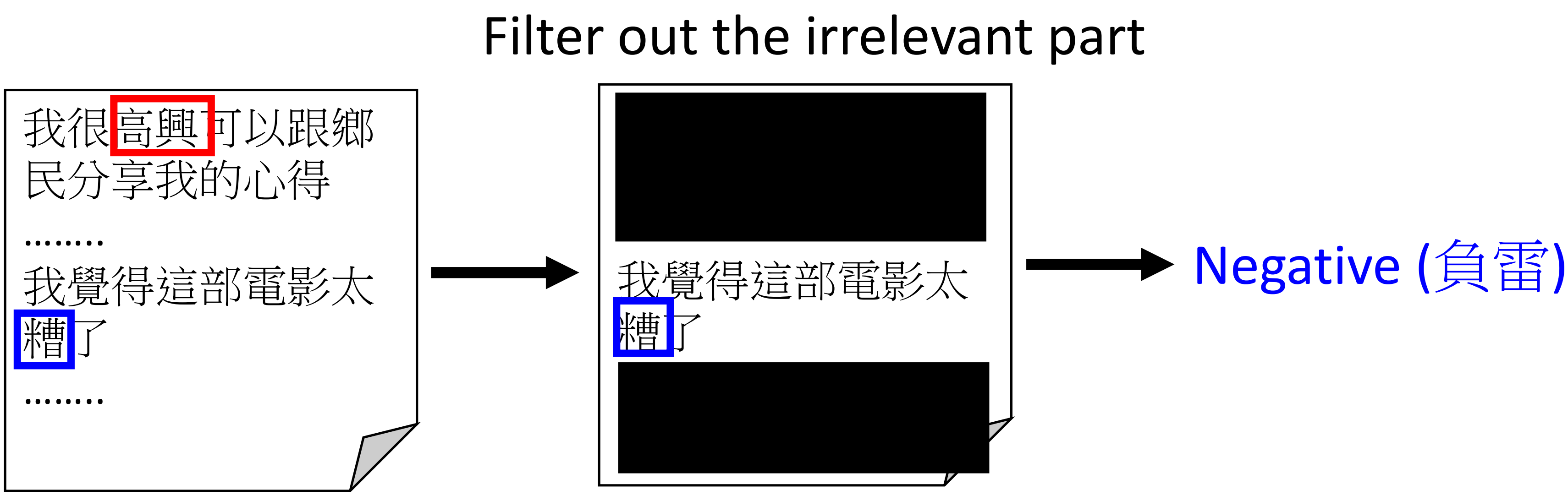
x^2: 這部電影太糟了……. → $\hat{y}^2$: Negative (負雷)

x^3: 這部電影很棒……. → $\hat{y}^3$: Positive (正雷)

This is only an ideal case.

Example Applications

- ***Sentiment Analysis***: Automatically identify a movie review is positive or negative



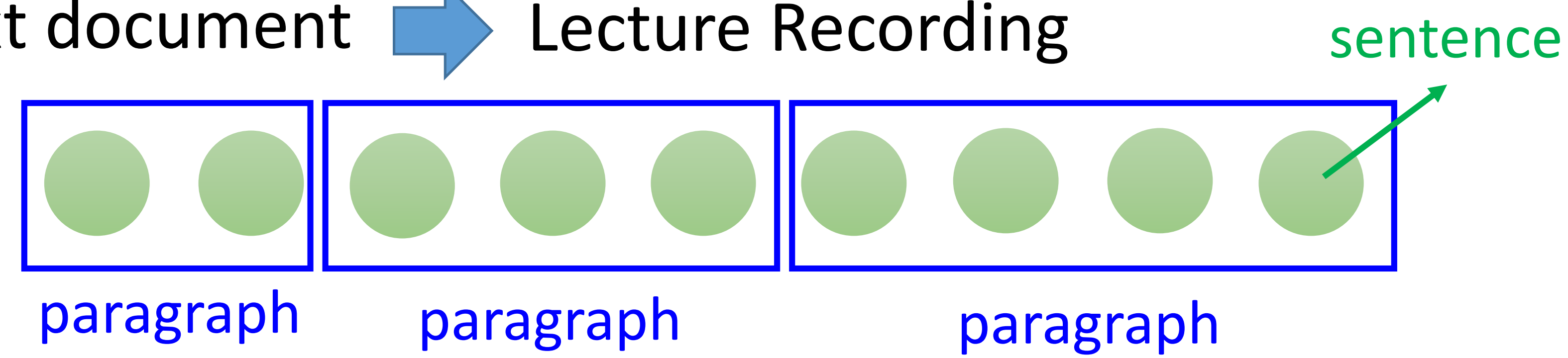
Only part of the document is related to movie review

Which parts are related to movie is hidden information.

Example Applications

- ***Summarization:*** Given a long document, select a set of sentences to form a compact version

Text document ➡ Lecture Recording



Select the whole paragraphs to make readable summaries

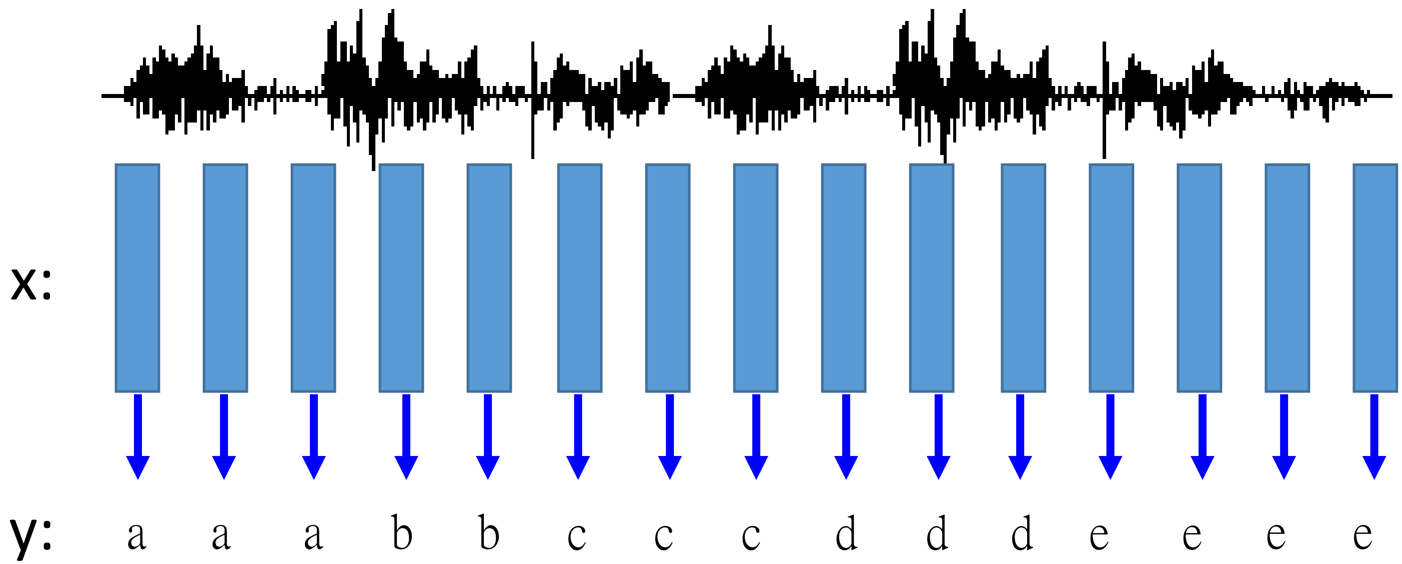
For speech, the paragraph boundaries are hidden.

Example Applications

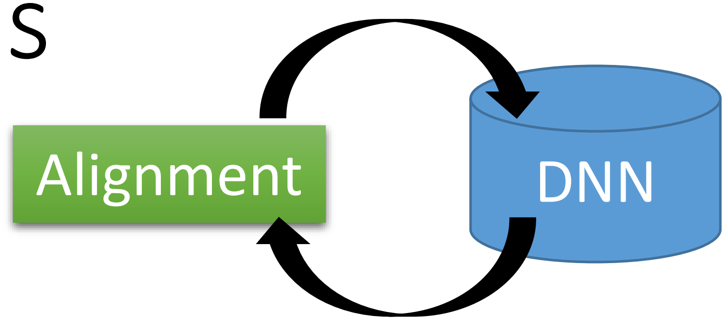
Speech Recognition

The training data in your homework ...

Phoneme or state of each frame is given.



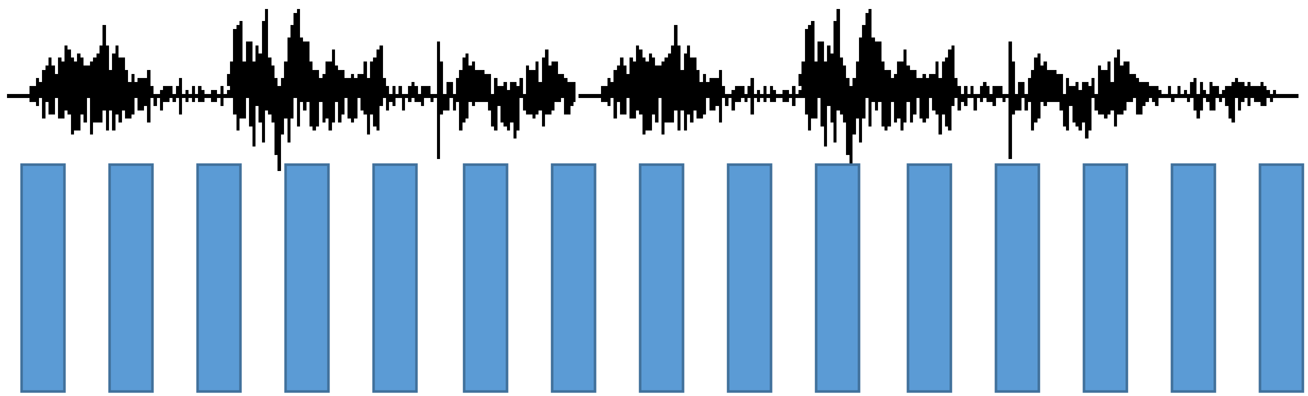
Example Applications



Speech Recognition

In the real world …

x:



y: a b c d e

The alignment between phonemes/states and acoustic features is hidden.

Example Applications

Machine Translation

English French

The word alignment of the sentence pairs is hidden.

https://buffy.eecs.berkeley.edu/PHP/resabs/resabs.php?f_year=2006&f_submit=chapgrp&f_chapter=12

There is a general framework.

Two Steps,
Three Questions

Two Steps

Step 1: Training

- Find function F
  - $F: X \times Y \times H \rightarrow R$
- $F(x, y, h)$ evaluate how compatible x, y and h is

Step2: Inference (Testing)

- Given object x
  - $\tilde{y} = arg \max_{y} \max_{h} F(x, y, h)$
  - $\tilde{y} = arg \max_{y} \sum_{h} F(x, y, h)$

Which one is more reasonable?

Three Problems

- ***Problem 1: Evaluation***
  - What does $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h})$ look like?
  - E.g. $F(x, y, h) = w \cdot \Psi(x, y, h)$
- ***Problem 2: Inference***
  - $\tilde{y} = arg \max_{y} \max_{h} F(x, y, h)$
  - $\tilde{y} = arg \max_{y} \sum_h F(x, y, h)$
- ***Problem 3: Training***
  - Given $\{(x^1, \hat{y}^1), \cdots (x^n, \hat{y}^n), \cdots (x^N, \hat{y}^N)\}$
  - EM-like algorithm

Three Problems - Training

Given Training data: $\{(x^1, \hat{y}^1), \cdots (x^n, \hat{y}^n), \cdots (x^N, \hat{y}^N)\}$

Initialize $F(x, y, h)$

Random?

Way 1. Find the most possible $\tilde{h}^n$

$$\tilde{h}^n = \arg\max_h F(x^n, \hat{y}^n, h)$$

Way 2. Find the probability distribution of h^n

We have $\{(x^1, h^1, \hat{y}^1), \cdots (x^n, h^n, \hat{y}^n), \cdots (x^N, h^N, \hat{y}^N)\}$

We know how to find $F(x, y, h)$ at least when it is linear.

Structured SVM with Hidden Information

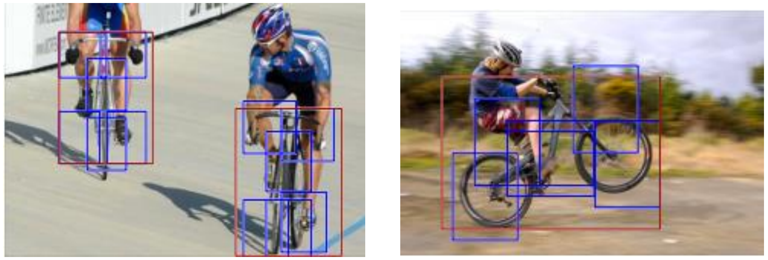
Taking object detection as Example

Motivation

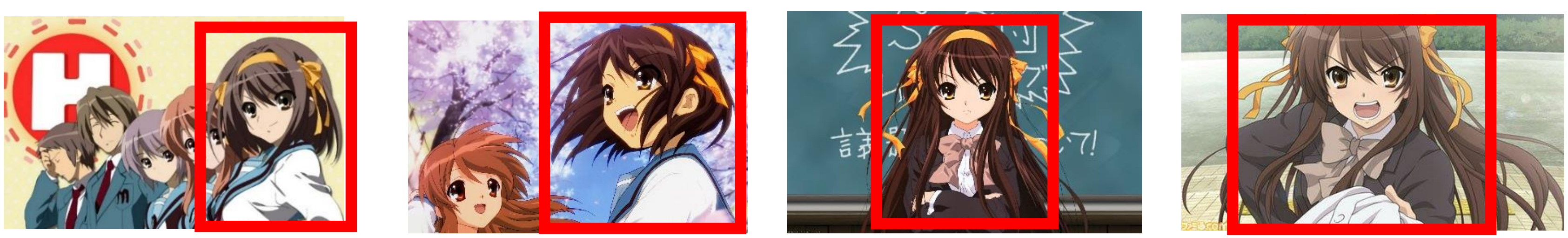
- An object can have more than one types

Train

Bicycle



Haruhi



short hair

long hair

Motivation

Type1. Short hair

Type 2. Long hair

Original Training:

For $\forall y \neq \hat{y}^1$: $w \cdot \phi(x^1, \hat{y}^1) > w \cdot \phi(x^1, y)$

For $\forall y \neq \hat{y}^2$: $w \cdot \phi(x^2, \hat{y}^2) > w \cdot \phi(x^2, y)$

- Because $\phi(x^1, \hat{y}^1)$ and $\phi(x^2, \hat{y}^2)$ can be very different
- It may be hard to use a single w to achieve the above goal

Two Cases

- Involving object types into object detection
- ***<u>Case 1</u>***
  - The useful information is available on training data, only hidden in testing data
  - Not too much difference from original structured SVM, extra efforts for labelling
- ***<u>Case 2</u>***
  - The information is hidden in both training and testing data
  - What we really care about

Case 1: Two kinds of Objects?

- There are two kinds of objects to be detected: **Haruhi_1** and **Haruhi_2**

Haruhi_1

Haruhi_1

Haruhi_2

Haruhi_2

Haruhi_1

Haruhi_1

Haruhi_2

Haruhi_2

Case 1: Two kinds of Objects?

Haruhi_1

Evaluation:

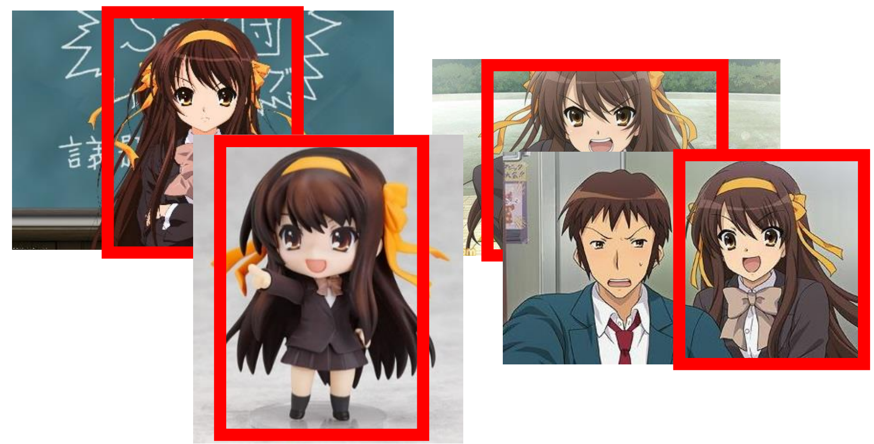
$$F_1(x, y) = w_1 \cdot \phi(x, y)$$

Training Target:

x^n is **Haruhi_1**

$$w_1 \cdot \phi(x^n, \hat{y}^n) > w_1 \cdot \phi(x^n, y)$$

Haruhi_2



Evaluation:

$$F_2(x, y) = w_2 \cdot \phi(x, y)$$

Training Target:

x^n is **Haruhi_2**

$$w_2 \cdot \phi(x^n, \hat{y}^n) > w_2 \cdot \phi(x^n, y)$$

Case 1: Problematic Inference

- Now we have w_1 for Haruhi_1 and w_2 for Haruhi_2
- Inference:

Given an image x

If the Harihu in image is Haruhi_1:

$$\tilde{y}_1 = \arg\max_{y \in \mathbb{Y}} w_1 \cdot \phi(x,y)$$

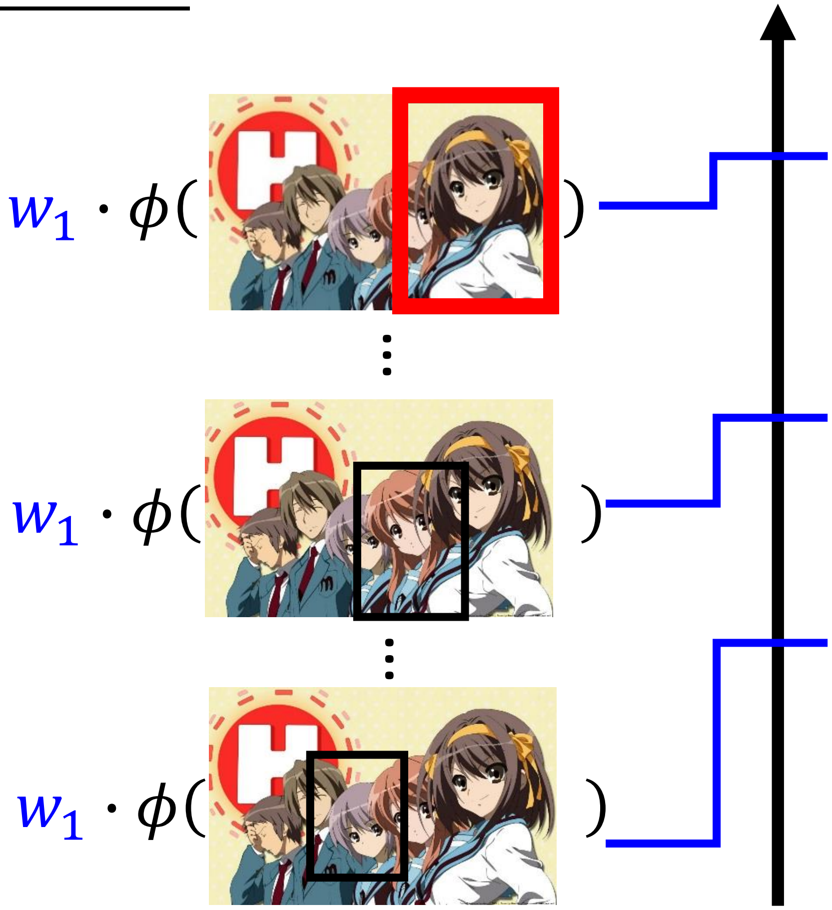
If the Harihu in image is Haruhi_2:

$$\tilde{y}_2 = \arg\max_{y \in \mathbb{Y}} w_2 \cdot \phi(x,y)$$

Critical Problem: Given an input image, we do not know the Haruhi in the image is Haruhi_1 or Haruhi_2

Case 1: Problematic Inference

- ***Inference***

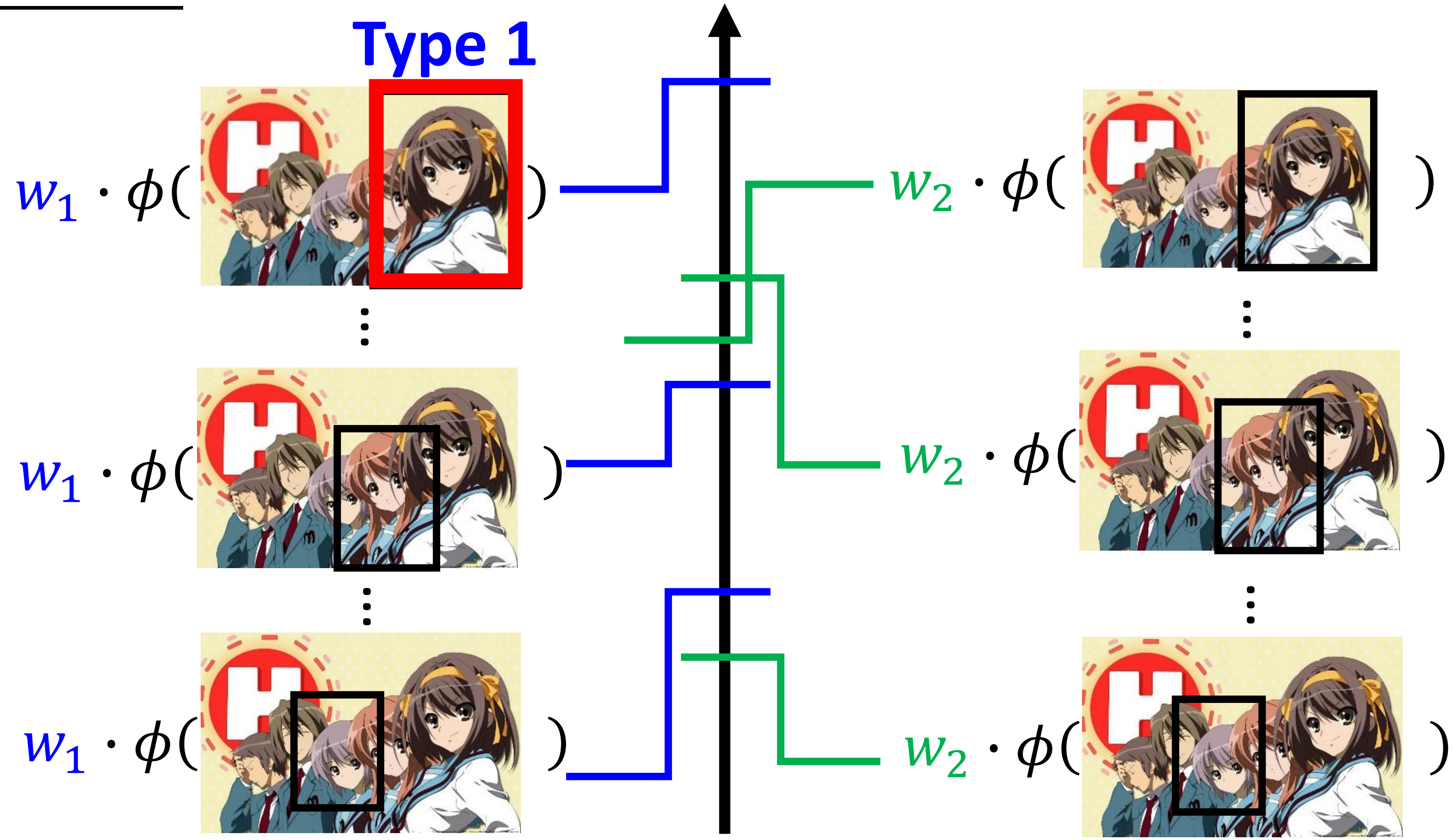


If we know its type 1

We don't know the type of the input image actually.

Case 1: Problematic Inference

- ***Inference***



Case 1: Problematic Inference

- w_1 and w_2 are learned separately

Training Target:

x^n *is* **Haruhi_1**

$$w_1 \cdot \phi(x^n, \hat{y}^n) > w_1 \cdot \phi(x^n, y)$$

0.1 0.09

Training Target:

x^n *is* **Haruhi_2**

$$w_2 \cdot \phi(x^n, \hat{y}^n) > w_2 \cdot \phi(x^n, y)$$

1000000 999999

x^n *is* **Haruhi_1**

$$w_2 \cdot \phi(x^n, y)$$

$$\vee$$

$$w_1 \cdot \phi(x^n, \hat{y}^n)$$

$$\vee$$

$$w_1 \cdot \phi(x^n, y)$$

w_1 and w_2 should be learned jointly

Case 1: Evaluation

For "type 1", $F(x, y) = w_1 \cdot \phi(x, y)$

For "type 2", $F(x, y) = w_2 \cdot \phi(x, y)$

$$\|$$

$$F(x, y, h) = w \cdot \Psi(x, y, h)$$

h: type of Haruhi (type 1 or type 2)

$\Psi(x, y, h)$: a feature vector for x, y and type h

Its length is twice of $\phi(x, y)$

w: a weight vector to be learned

Its length is twice of w_1 or w_2

Case 1: Evaluation

$$F(x,y,h) = \underline{w} \cdot \underline{\Psi(x,y,h)}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\begin{cases} \Psi(x,y,h = \text{"}type\ 1\text{"}) = \begin{bmatrix} \phi(x,y) \\ \mathbf{0} \end{bmatrix} \\ \Psi(x,y,h = \text{"}type\ 2\text{"}) = \begin{bmatrix} \mathbf{0} \\ \phi(x,y) \end{bmatrix} \end{cases}$$

For “type 1”, $\mathrm{F}(x,y,h) = w_1 \cdot \phi(x,y) + w_2 \cdot \mathbf{0}$

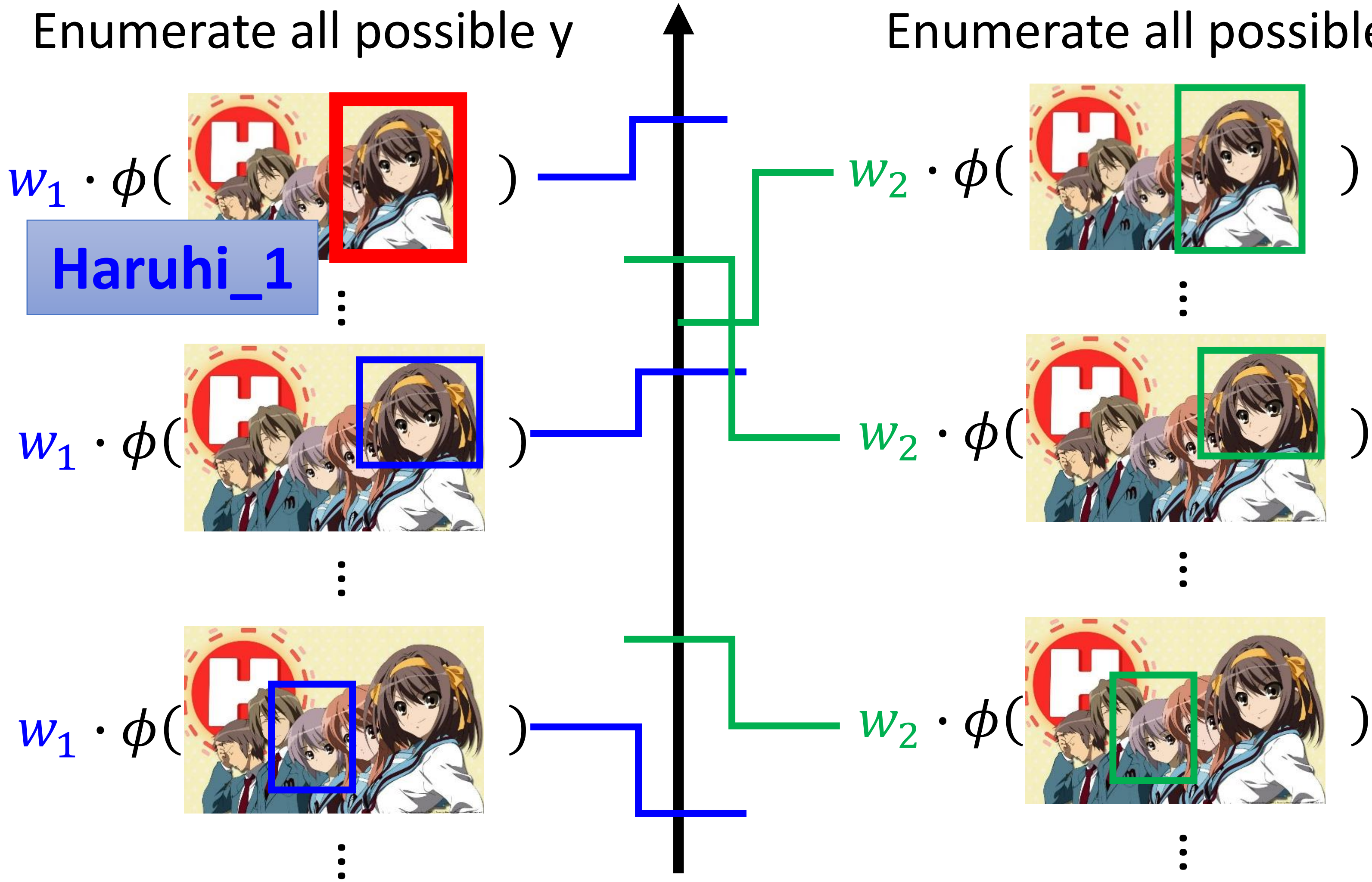
For “type 2”, $\mathrm{F}(x,y,h) = w_1 \cdot \mathbf{0} + w_2 \cdot \phi(x,y)$

Case 1: Inference

$$\tilde{y} = \arg\max_{y} \max_{h} w \cdot \Psi(x, y, h)$$

Enumerate all possible y

Enumerate all possible y



Case 1: Training

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

$$C^n = \max_{y}[w \cdot \phi(x^n, y)] - w \cdot \phi(x^n, \hat{y}^n)$$

$$C^n = \max_{y}[\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)] - w \cdot \phi(x^n, \hat{y}^n)$$

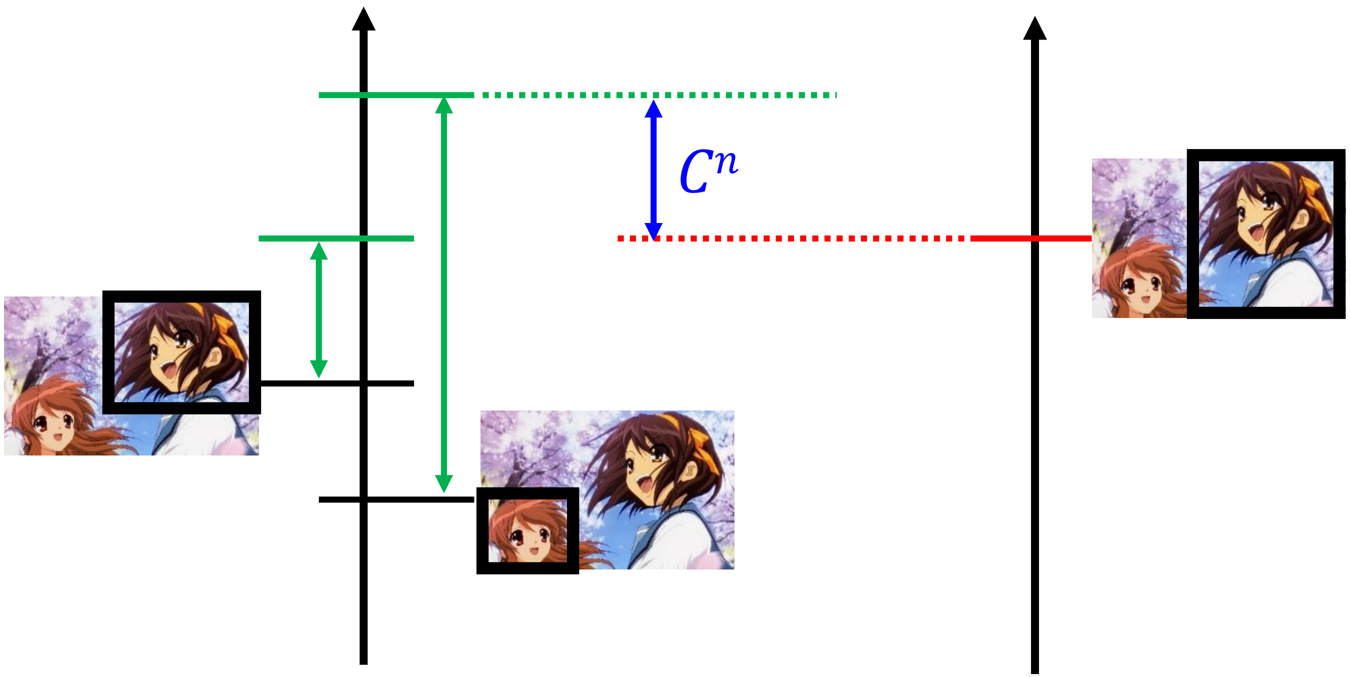
$$\tilde{y} = arg\max_{y}\max_{h} w \cdot \Psi(x, y, h)$$

$$C^n = \max_{y}\max_{h}[w \cdot \Psi(x^n, y, h)] - w \cdot \Psi\left(x^n, \hat{y}^n, \hat{h}^n\right)$$

$$C^n = \max_{y}\max_{h}[\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - w \cdot \Psi\left(x^n, \hat{y}^n, \hat{h}^n\right)$$

Case 1: Training

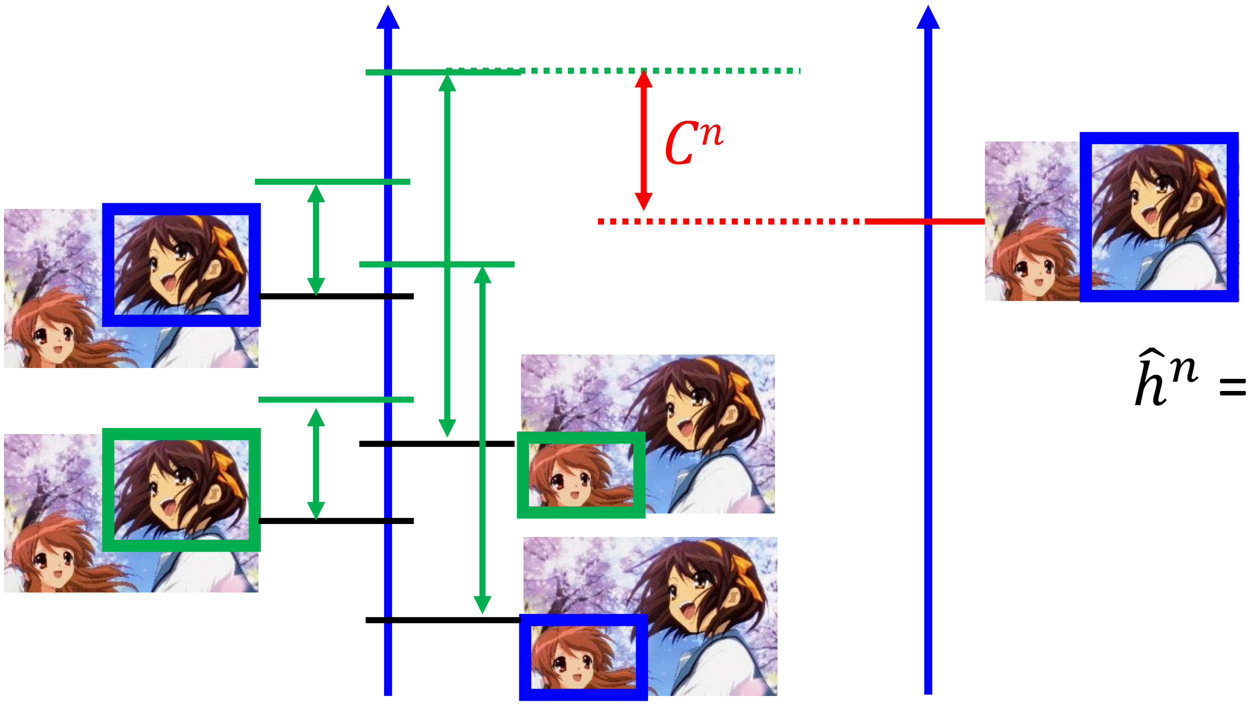
$$C^n = \max_y [\underline{\Delta(\hat{y}^n, y)} + \underline{w \cdot \phi(x^n, y)}] \underline{-w \cdot \phi(x^n, \hat{y}^n)}$$



Case 1: Training

$$C^n = \max_y \max_h [\underline{\Delta(\hat{y}^n, y)} + \underline{w \cdot \Psi(x^n, y, h)}]$$

$$\underline{-w \cdot \Psi(x^n, \hat{y}^n, \hat{h}^n)}$$



$\hat{h}^n$ = "type 1"

Case 1: Training

Given training data: $\left\{\left(x^1, \hat{y}^1, \hat{h}^1\right), \cdots, \left(x^n, \hat{y}^n, \hat{h}^n\right), \cdots, \left(x^N, \hat{y}^N, \hat{h}^N\right)\right\}$

$$C^n = \max_y \max_h \left[\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)\right] - w \cdot \Psi\left(x^n, \hat{y}^n, \hat{h}^n\right)$$

$\Updownarrow$

Find $w, \varepsilon^1, \ldots, \varepsilon^n, \ldots, \varepsilon^N$ minimize: $\frac{1}{2}\|w\|^2 + \lambda \sum_{n=1}^{N} \varepsilon^n$

$\forall n, \forall y \in \mathbb{Y}, \forall h \in \mathbb{H}$

$$w \cdot \Psi\left(x^n, \hat{y}^n, \hat{h}^n\right) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - \varepsilon^n$$

Case 2:
Training with Hidden Information

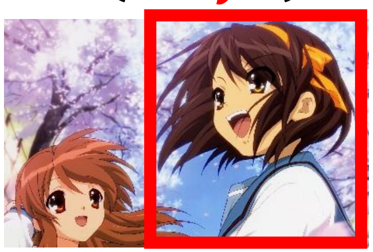
- The useful information are usually ***hidden***

How to deal with hidden information with Structured SVM?

Case 2: Training with Hidden Information

- No types? **Try to generate ourselves**

$(x^1, \hat{y}^1)$ $(x^2, \hat{y}^2)$ $(x^3, \hat{y}^3)$ $(x^4, \hat{y}^4)$



$$F(x, y, h) = \boxed{w} \cdot \Psi(x, y, h)$$

Random initialized

$$w = w^0$$

Evaluate the compatibility of x, y and h

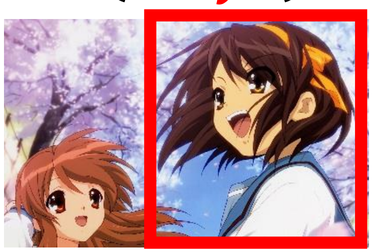
$$\tilde{h} = arg \max_h w \cdot \Psi(x, \hat{y}, h)$$

Given x and $\hat{y}$, find the most compatible h

Case 2: Training with Hidden Information

- No types? **Try to generate ourselves**

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



$\tilde{h}^1$ = type 1

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

$\tilde{h}^2$ = type 2

$(x^3, \hat{y}^3)$

$\tilde{h}^3$ = type 1

$(x^4, \hat{y}^4)$

$\tilde{h}^4$ = type 2

For n = 1, …, 4: $\tilde{h}^n = arg \max_h w^0 \cdot \Psi(x^n, \hat{y}^n, h)$

Good guess? Of course not.

Because $w0$ is random

Case 2:
Training with Hidden Information

- With the types we generate, we can find a w

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

$\tilde{h}^1$ = type 1

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

$\tilde{h}^2$ = type 2

$(x^3, \hat{y}^3)$

$\tilde{h}^3$ = type 1

$(x^4, \hat{y}^4)$

$\tilde{h}^4$ = type 2

Find $w, \varepsilon^1, \varepsilon^2, \varepsilon^3, \varepsilon^4$ minimize: $\frac{1}{2}\|w\|^2 + \lambda \sum_{n=1}^{4} \varepsilon^n$

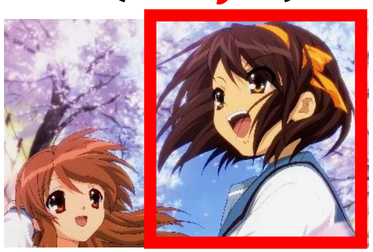
$\mathrm{n} = 1, \dots, 4, \forall y \in \mathbb{Y}, \forall h \in \mathbb{H}$

$$w \cdot \Psi\left(x^n, \hat{y}^n, \tilde{h}^n\right) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - \varepsilon^n$$

Case 2: Training with Hidden Information

For n = 1, …, 4: $\tilde{h}^n = arg \max_h w^1 \cdot \Psi(x^n, \hat{y}^n, h)$

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



$\tilde{h}^1$ = type 1

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

$\tilde{h}^2$ = type 2

$(x^3, \hat{y}^3)$

$\tilde{h}^3$ = type 2

$(x^4, \hat{y}^4)$

$\tilde{h}^4$ = type 2

w^1

Solving a QP

Is w^1 a good weight vector? Probably not

Train from random $\tilde{h}$

Case 2: Training with Hidden Information

For n = 1, …, 4: $\tilde{h}^n = arg \max_h w^2 \cdot \Psi(x^n, \hat{y}^n, h)$

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

$\tilde{h}^1$= type 1

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

$\tilde{h}^2$= type 1

$(x^3, \hat{y}^3)$

$\tilde{h}^3$= type 2

$(x^4, \hat{y}^4)$

$\tilde{h}^4$= type 2

Use the new $\tilde{h}$ to solve the same QP again

Solving a QP

w^2

Is w^2 better than w^1? Yes (?)

Iteratively

Case 2: Training with Hidden Information

Summary Iteration in Iteration

Training data: $\{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots (x^n, \hat{y}^n) \cdots (x^N, \hat{y}^N)\}$

Initialize w

$$\tilde{h}^n = \arg\max_h w \cdot \Psi(x^n, \hat{y}^n, h)$$

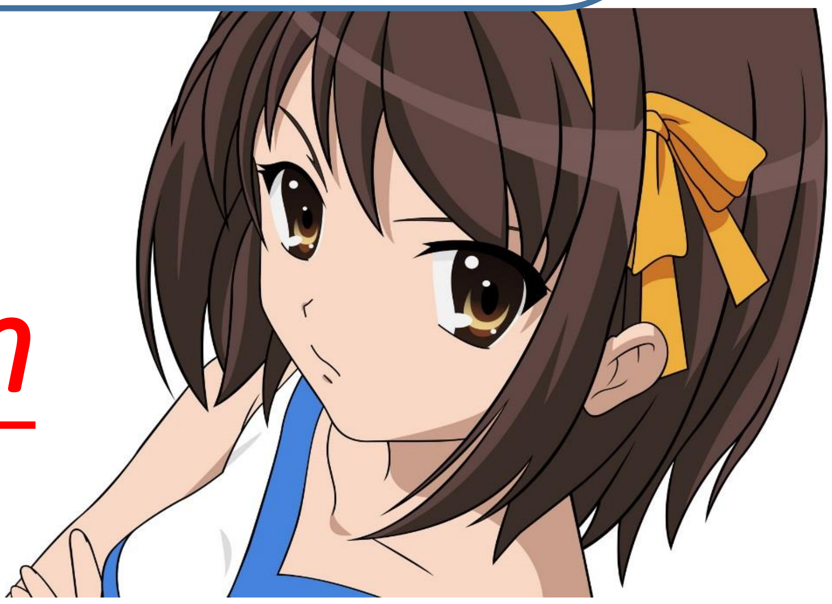
Cutting Plane Algorithm

Find $w, \varepsilon^1, \cdots, \varepsilon^N$ minimize $\frac{1}{2}\|w\|^2 + \lambda \sum_{n=1}^{N} \varepsilon^n$

$\forall r, \forall y \in \mathbb{Y}, \forall h \in \mathbb{H}$

$$w \cdot \Psi\left(x^n, \hat{y}^n, \tilde{h}^n\right) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - \varepsilon^n$$

Why we can get better weight vector after each iteration?



Warning of Math

Structured SVM

Training data: $\{(x^1, \hat{y}^1), \cdots (x^n, \hat{y}^n) \cdots (x^N, \hat{y}^N)\}$

Minimizing cost

$$\tilde{y} = arg \max_y w \cdot \phi(x, y)$$

$$C = \frac{1}{2}\|w\|^2 + \sum_{n=1}^{N} C^n \quad \geq \quad \sum_{n=1}^{N} \Delta(\hat{y}^n, \tilde{y}^n)$$

$$C^n \geq \Delta(\hat{y}^n, \tilde{y}^n)$$

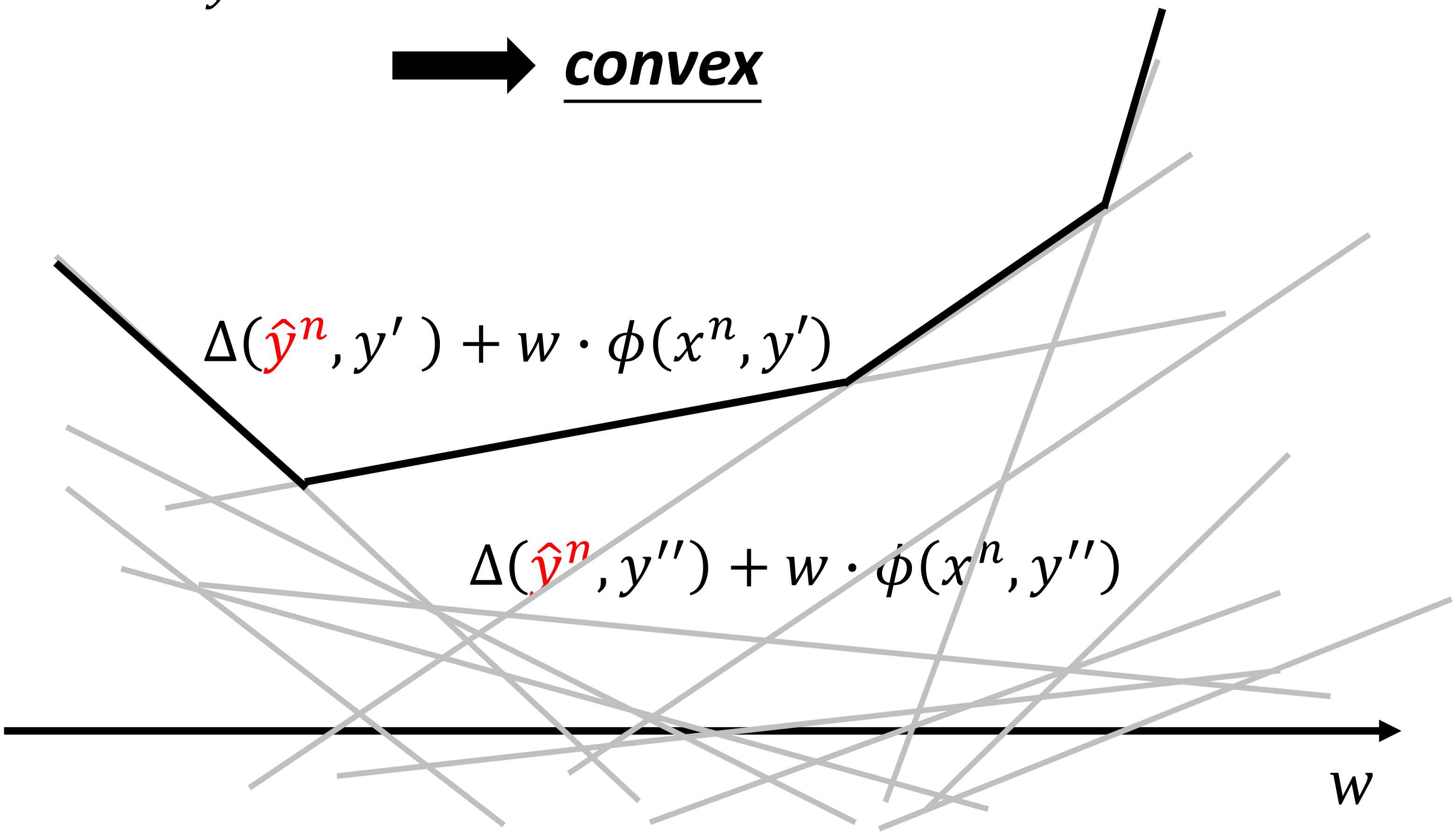
$$C^n = \max_y [\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)] - w \cdot \phi(x^n, \hat{y}^n)$$

What does the function C^n look like?

Structured SVM

$$C^n = \underbrace{\max_y[\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)]}\ - w \cdot \phi(x^n, \hat{y}^n)$$

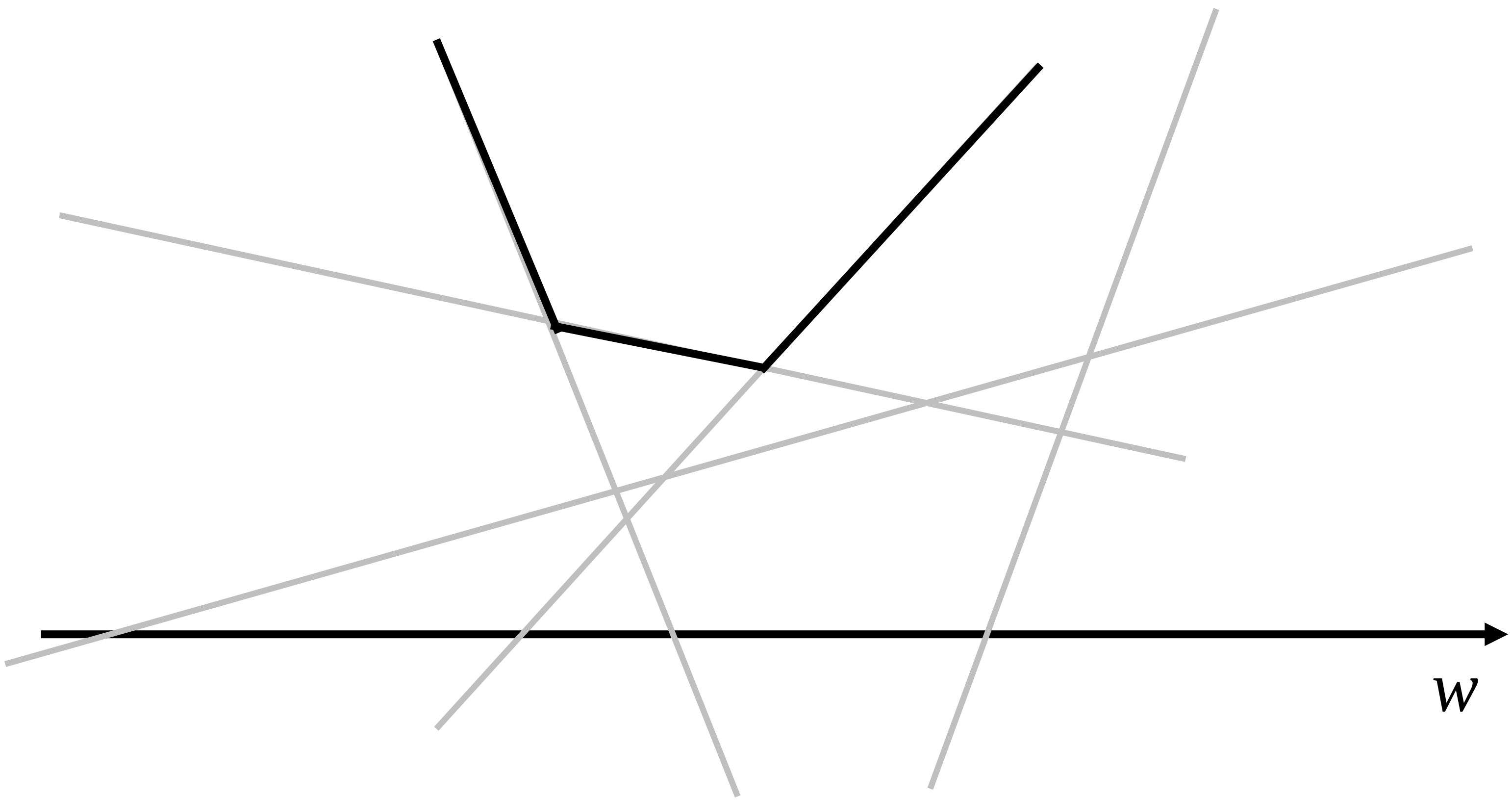
➡ ***convex***



Structured SVM

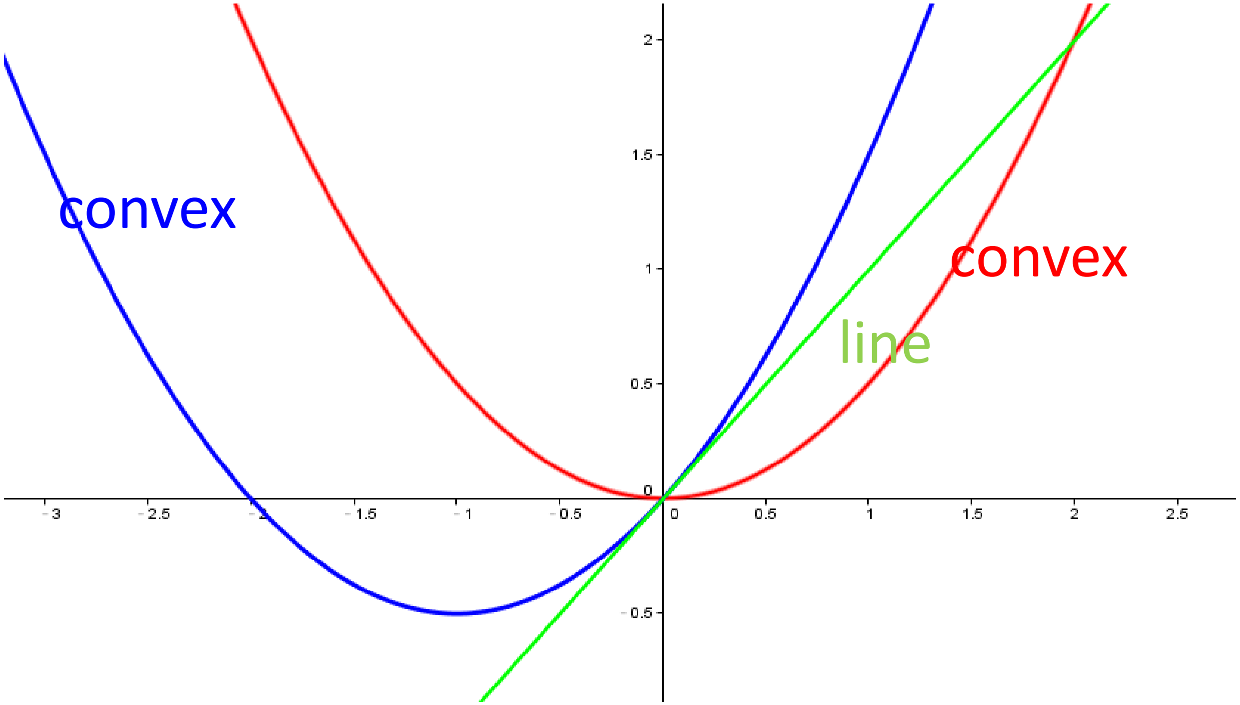
$$C^n = \underline{\max_y[\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)]} - w \cdot \phi(x^n, \hat{y}^n)$$

→ ***convex***



Structured SVM

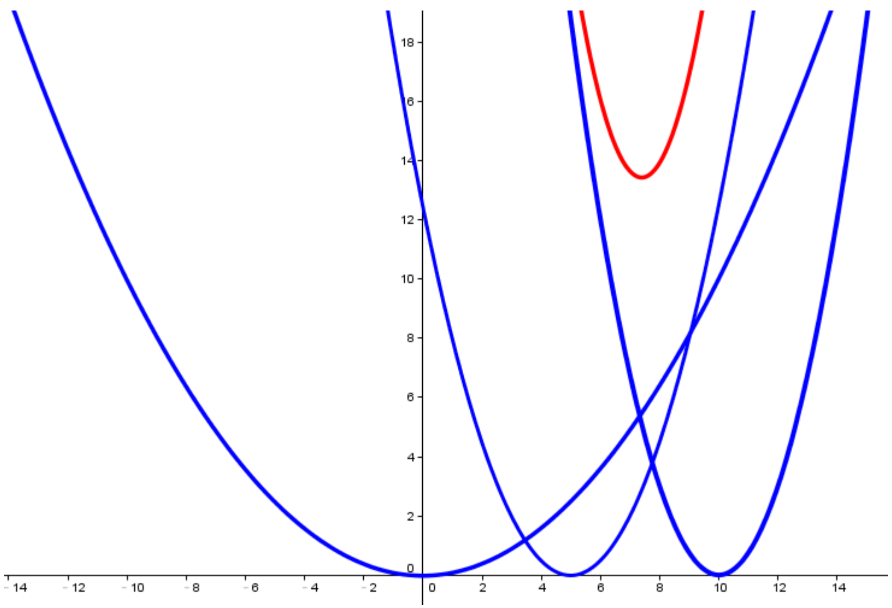
$$C^n = \overbrace{\underbrace{\max_{y}[\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)]}_{\text{convex}} \underbrace{- w \cdot \phi(x^n, \hat{y}^n)}_{\text{line}}}^{\text{convex}}$$

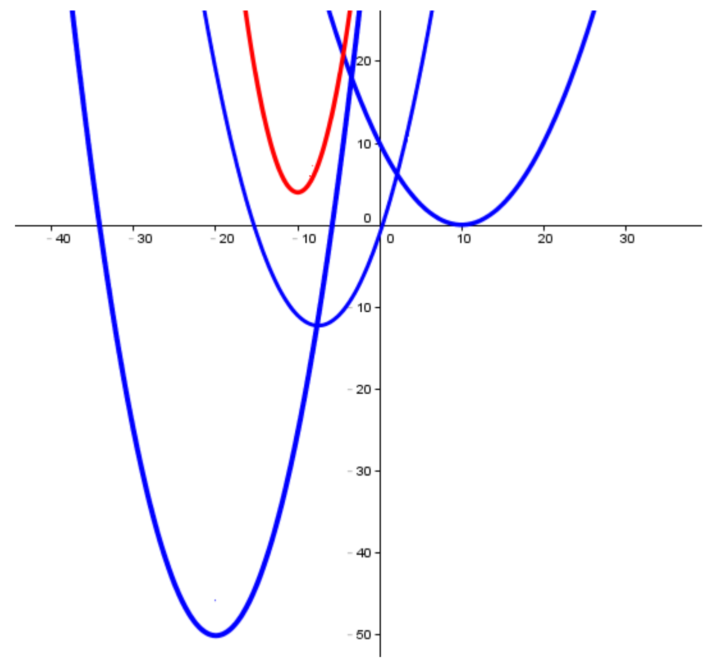


Structured SVM

There is no local minima for structured SVM.

$$C = \overbrace{\underbrace{\frac{1}{2}\|w\|^2}_{\text{convex}} + \underbrace{\sum_{n=1}^{N} C^n}_{\text{convex}}}^{\text{convex}}$$





Structured SVM with Hidden Information

Training data: $\{(x^1, \hat{y}^1), \cdots (x^n, \hat{y}^n) \cdots (x^N, \hat{y}^N)\}$

In each iteration, the following cost is smaller

$$\tilde{y} = arg \max_y \max_h w \cdot \Psi(x, y, h)$$

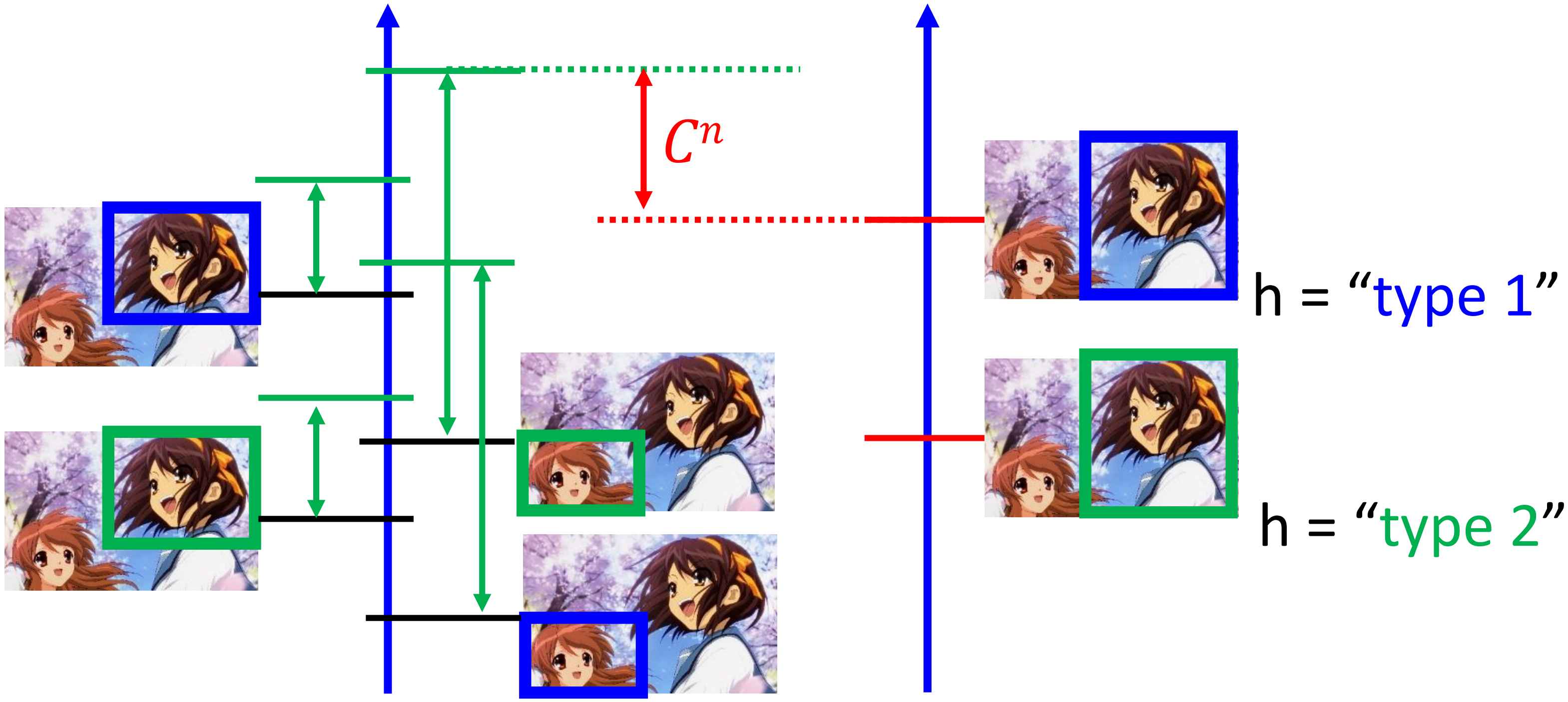
$$C = \frac{1}{2}\|w\|^2 + \sum_{n=1}^{N} C^n \geq \sum_{n=1}^{N} \Delta(\hat{y}^n, \tilde{y}^n)$$

$$C^n \geq \Delta(\hat{y}^n, \tilde{y}^n)$$

$$C^n = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - \max_h w \cdot \Psi(x^n, \hat{y}^n, h)$$

Structured SVM with Hidden Information

$$C^n = \max_y \max_h [\underline{\Delta(\hat{y}^n, y)} + \underline{w \cdot \Psi(x^n, y, h)}] \\ \underline{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}$$



Structured SVM with Hidden Information

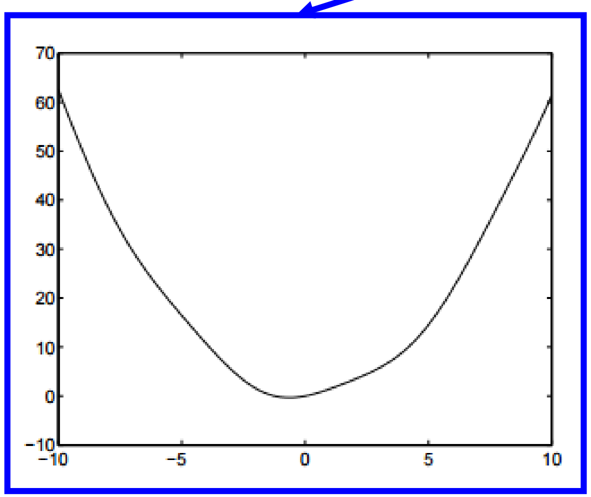
- Cost function to be minimized

$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} - \underbrace{\max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{convex}}$$

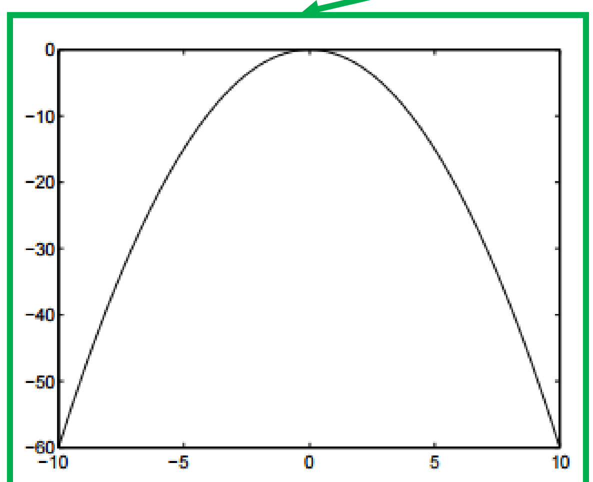
Structured SVM with Hidden Information

- Cost function to be minimized

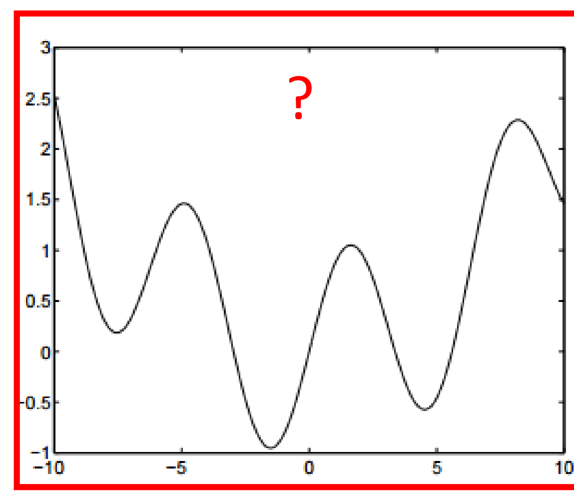
$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} \underbrace{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{concave}}$$



+



=



$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} \underbrace{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{concave}}$$

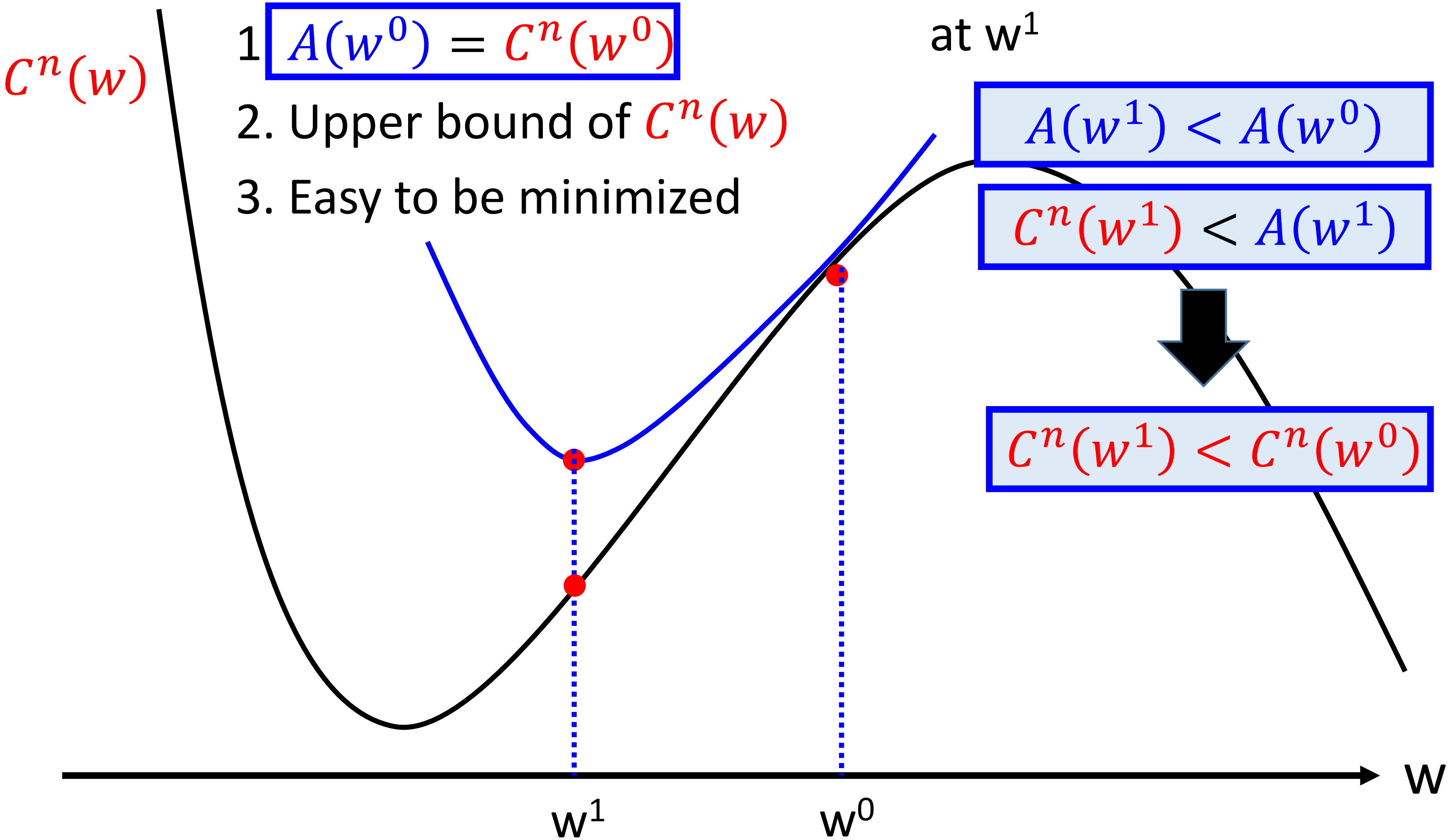
Auxiliary function $A(w)$ at w^0:

1 $\boxed{A(w^0) = C^n(w^0)}$

2. Upper bound of $C^n(w)$

3. Easy to be minimized

Minimum value of $A(w)$ is at w^1



$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} \underbrace{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{concave}}$$

Another auxiliary function $A(w)$ at w^1:

1. $\boxed{A(w^1) = C^n(w^1)}$
2. Upper bound of $C^n(w)$
3. Easy to be minimized

Minimum value of $A(w)$ is at w^2

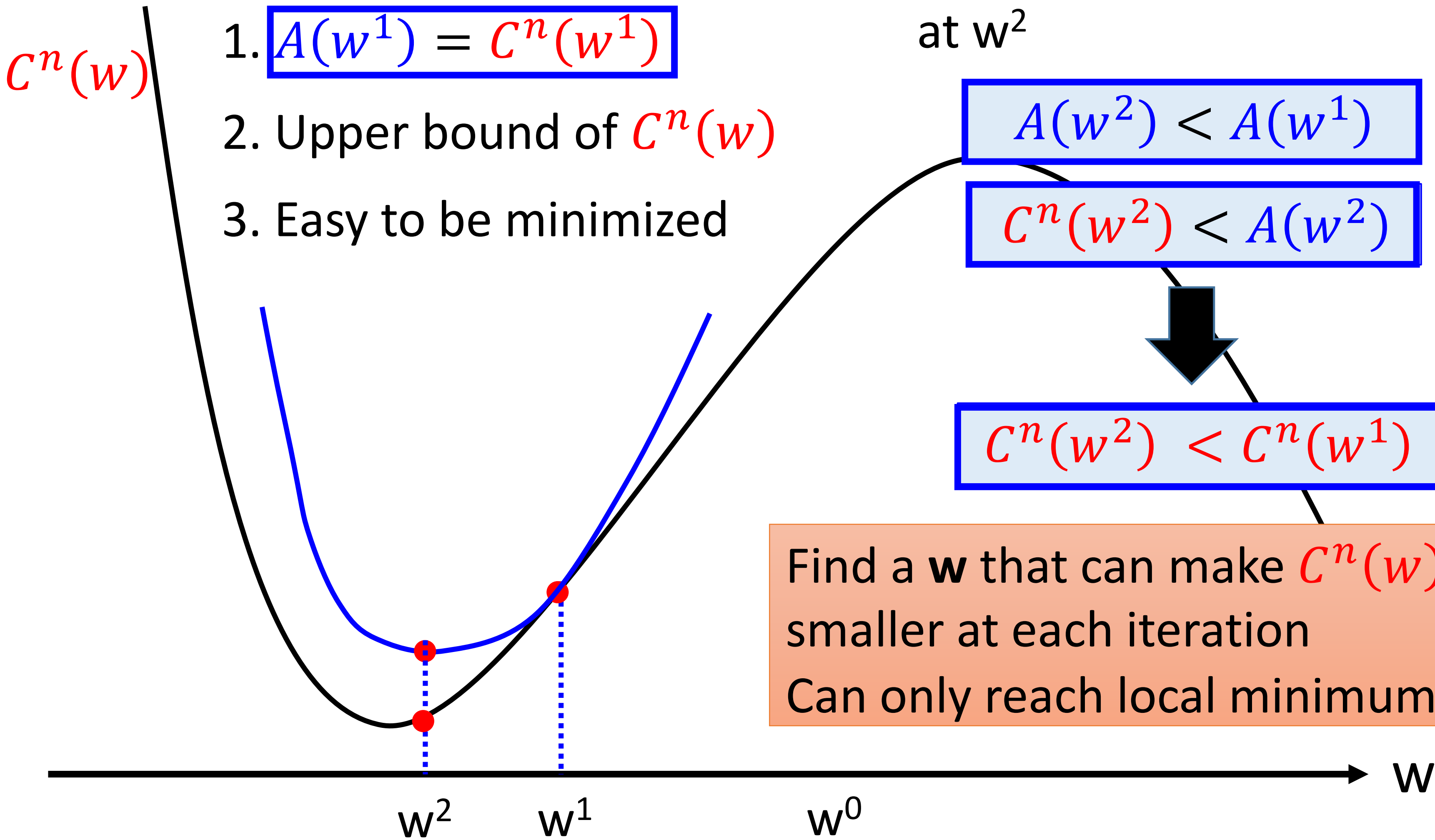
$\boxed{A(w^2) < A(w^1)}$

$\boxed{C^n(w^2) < A(w^2)}$

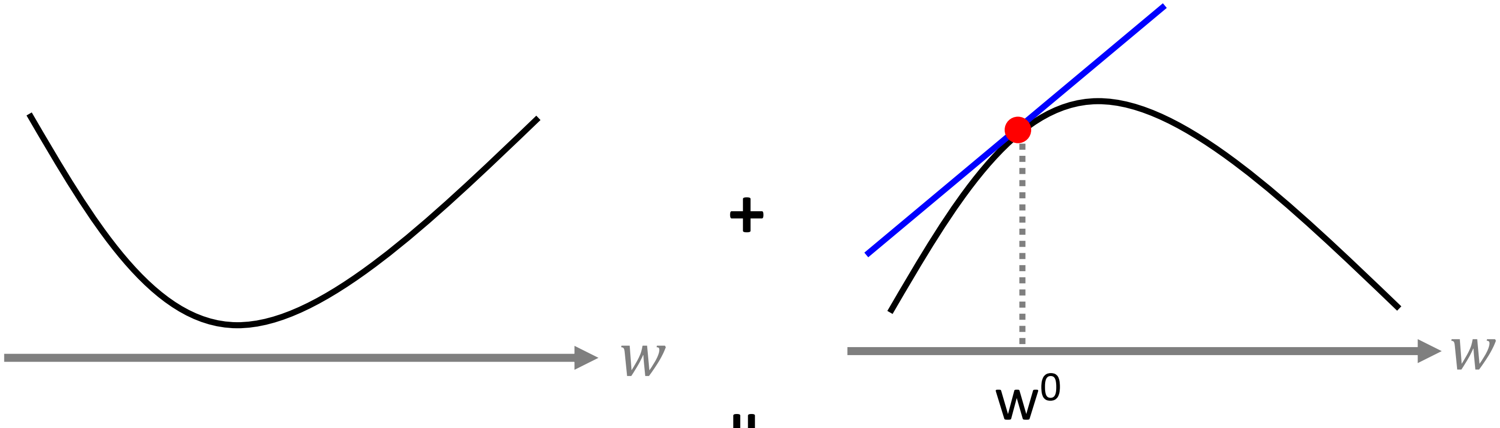
$\boxed{C^n(w^2) < C^n(w^1)}$

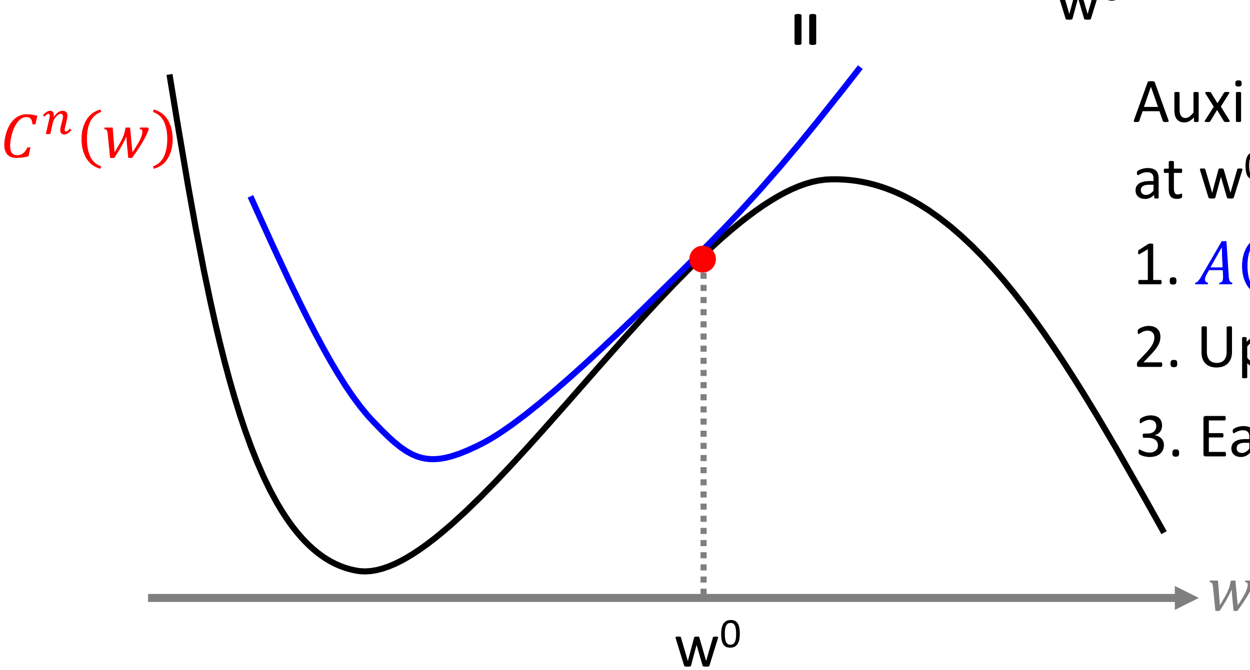
Find a **w** that can make $C^n(w)$ smaller at each iteration

Can only reach local minimum



$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} \underbrace{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{concave}}$$





Auxiliary function $A(w)$ at w^0:

1. $A(w^0) = C^n(w^0)$
2. Upper bound of $C^n(w)$
3. Easy to be minimized

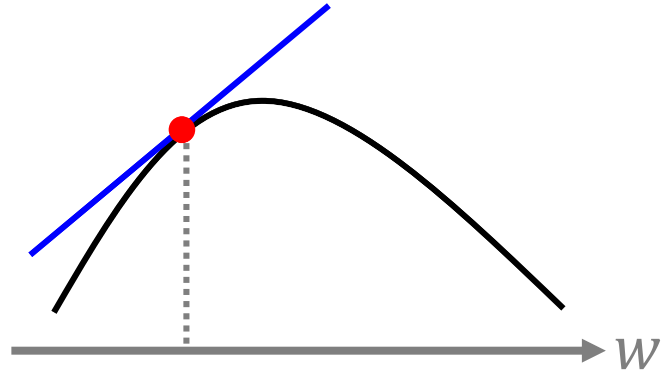
➡ convex

What is the relation to the EM-like process?

$$\tilde{h}^n = arg \max_h w \cdot \Psi(x^n, \hat{y}^n, h)$$

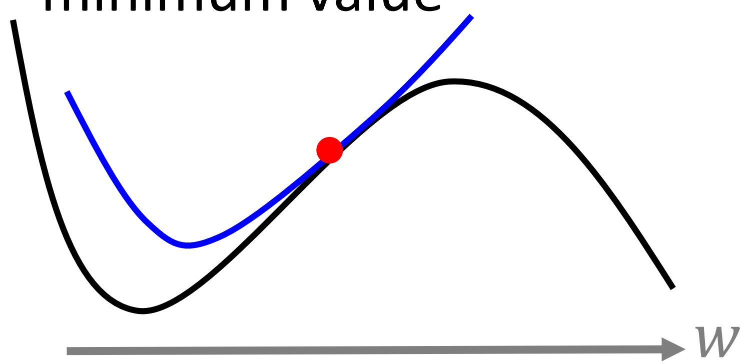
Solving a QP

After each iteration, the w obtained decrease the cost function

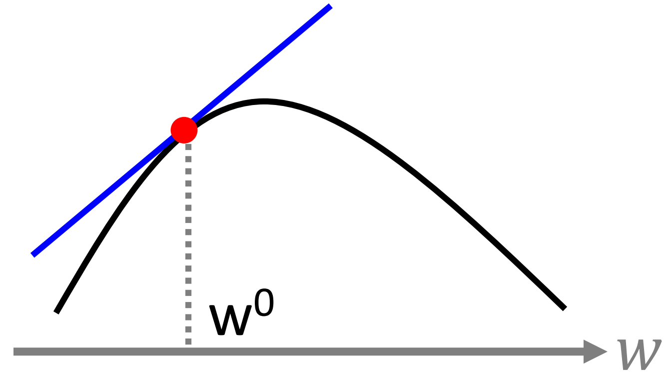


to form the auxiliary function

find the minimum value

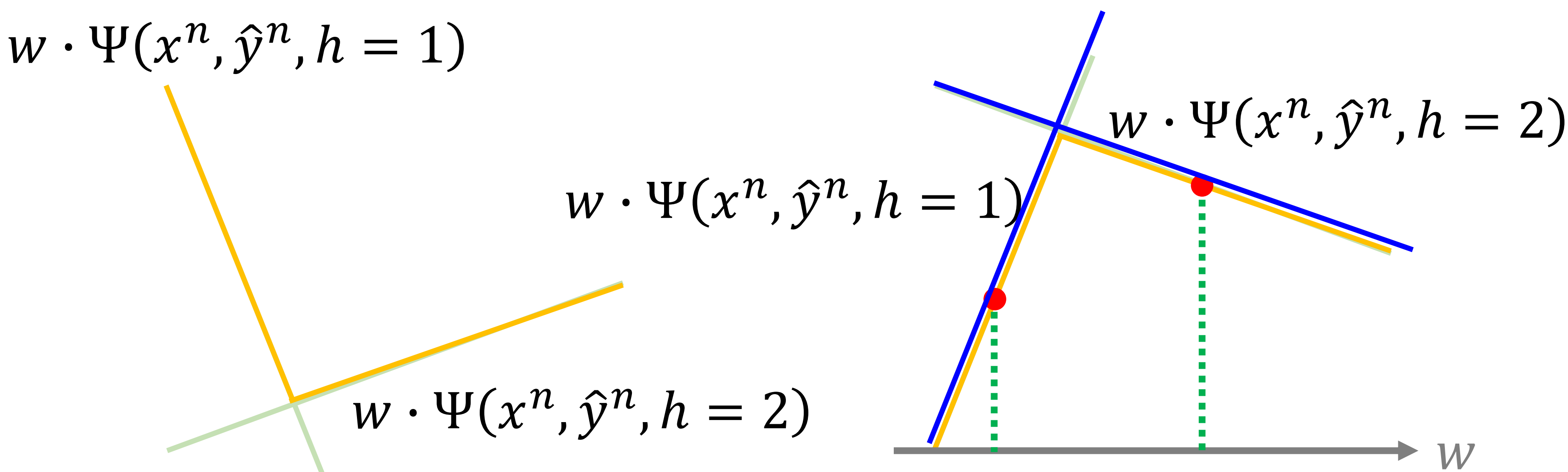


$$\tilde{h}^n = arg \max_h w \cdot \Psi(x^n, \hat{y}^n, h)$$

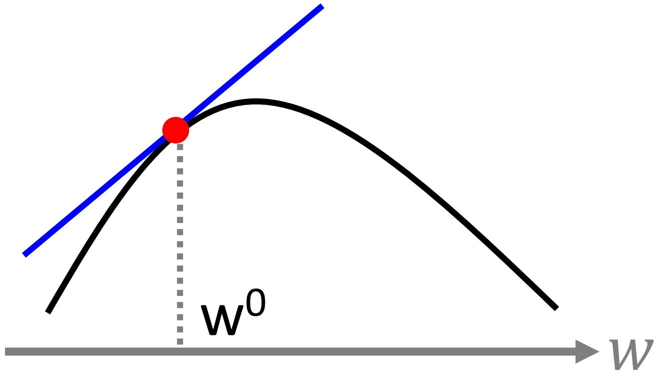


to form the auxiliary function

$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} \underbrace{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{concave}}$$



$$\tilde{h}^n = arg\max_h w \cdot \Psi(x^n, \hat{y}^n, h)$$



to form the auxiliary function

$$C^n = \underbrace{\max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)]}_{\text{convex}} \underbrace{- \max_h w \cdot \Psi(x^n, \hat{y}^n, h)}_{\text{concave}}$$

$$\tilde{h}^n = arg\max_h w^0 \cdot \Psi(x^n, \hat{y}^n, h)$$

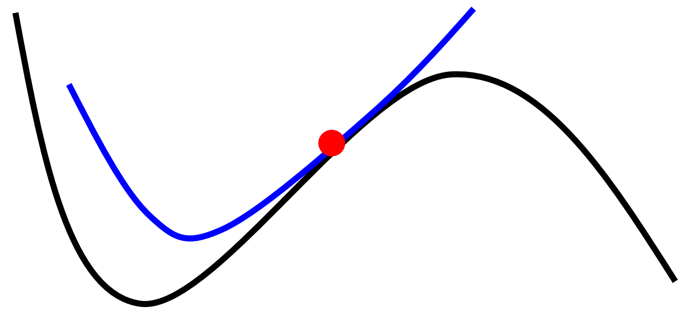
$$A(w) = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - w \cdot \Psi\left(x^n, \hat{y}^n, \tilde{h}^n\right)$$

Minimizing A(w)

$$A(w) = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - w \cdot \Psi(x^n, \hat{y}^n, \tilde{h}^n)$$

find the minimum value

Solving a QP



$$A(w) = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - w \cdot \Psi(x^n, \hat{y}^n, \tilde{h}^n)$$

$$w \cdot \Psi(x^n, \hat{y}^n, \tilde{h}^n) - \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] = -A(w)$$

$\forall y \in \mathbb{Y}, \forall h \in \mathbb{H}$

$$w \cdot \Psi(x^n, \hat{y}^n, \tilde{h}^n) - [w \cdot \Psi(x^n, y, h) + \Delta(\hat{y}^n, y)] \geq -A(w)$$

$\forall y \in \mathbb{Y}, \forall h \in \mathbb{H}$

$$w \cdot \Psi(x^n, \hat{y}^n, \tilde{h}^n) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - A(w)$$

End of Warning

Structured SVM with Hidden Information

Problem 1: Evaluation

$$F(x, y, h) = w \cdot \Psi(x, y, h)$$

Problem 2: Inference

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

Problem 3: Training

EM-like algorithm

Find hidden information ⇄ Find model parameters

To Learn More …

- Framework
  - Chun-Nam John Yu and Thorsten Joachims, ”Learning Structural SVMs with Latent Variables,” ICML 2009
- Video
  - Wang, Yang, and Greg Mori. "Max-margin hidden conditional random fields for human action recognition," *CVPR 2009*
  - Wang, Yang, and Greg Mori. "Hidden part models for human action recognition: Probabilistic versus max margin," *Pattern Analysis and Machine Intelligence, IEEE Transactions,* 2011
- Image
  - Zhu, Long, et al. "Latent hierarchical structural learning for object detection."*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.
  - Felzenszwalb, Pedro F., et al. "Object detection with discriminatively trained part-based models." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.9 (2010): 1627-1645.
- Language processing
  - Sun, Xu, et al. "Latent Variable Perceptron Algorithm for Structured Classification," *IJCAI*. Vol. 9. 2009
  - *http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015/Structured%20Lecture/Summarization%20Hidden_2.ecm.mp4/index.html*

Appendix:
EM in one slide

EM in one slide

Maximizing

$$\prod_{n=1}^{N} \sum_{h} P(x^n, y^n, h)$$

Problem 1: Evaluation

$$F(x, y, h) = \mathrm{P}(x, y, h)$$

Problem 2: Inference

$$\tilde{y} = arg \max_{y} \sum_{h} P(x, y, h)$$

Problem 3: Training

Find hidden information

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

Find model parameters

$$P(x, y, h) = P(h|x, y) P(x, y)$$