Learning with Hidden Information

Hung-yi Lee
Different Kinds of Learning

- **Supervised Learning**
  - Data: \{ (x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots \}

- **Semi-supervised Learning**
  - Data: \{ (x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots, (x^{N+1}, ?), (x^{N+2}, ?) \ldots \}

- **Unsupervised Learning**
  - Data: \{ (x^1, ?), (x^2, ?), \ldots \}

- **Hidden variable learning**
  - Data: \{ (x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots \}

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

\[ x^1: \quad \text{看了這部電影覺得很高興} \ldots \quad \Rightarrow \quad \hat{y}^1: \text{Positive (正雷)} \]

\[ x^2: \quad \text{這部電影太糟了} \ldots \quad \Rightarrow \quad \hat{y}^2: \text{Negative (負雷)} \]

\[ x^3: \quad \text{這部電影很棒} \ldots \quad \Rightarrow \quad \hat{y}^3: \text{Positive (正雷)} \]

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.

Filter out the irrelevant part

Positive (正義)

Negative (負面)
Example Applications

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

Text document $\rightarrow$ 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 ...

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

**Machine Translation**

What is the anticipated cost of collecting fees under the new proposal?  
En vertu de les nouvelles propositions, quel est le coût prévu de perception de les droits?

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

**Step 2: Inference (Testing)**

- Given object $x$
  - $\tilde{y} = \arg\max_y \max_h F(x, y, h)$
  - $\hat{y} = \arg\max_y \sum_h F(x, y, h)$

Which one is more reasonable?
Three Problems

- **Problem 1: Evaluation**
  - What does $F(x, y, h)$ look like?
  - E.g. $F(x, y, h) = w \cdot \Psi(x, y, h)$

- **Problem 2: Inference**
  - $\hat{y} = \arg \max_y \max_h F(x, y, h)$
  - $\hat{y} = \arg \max_y \sum_h F(x, y, h)$

- **Problem 3: Training**
  - Given $\{(x^1, \hat{y}^1), \ldots (x^n, \hat{y}^n), \ldots (x^N, \hat{y}^N)\}$
  - 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) \}

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

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) \}
Structured SVM with Hidden Information

Taking object detection as Example
Motivation

• An object can have more than one types
Motivation

Type 1. Short hair

\[ x^1 \quad \hat{y}^1 \]

Type 2. Long hair

\[ x^2 \quad \hat{y}^2 \]

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

Original Training:

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

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

• Involving object types into object detection

• **Case 1**
  • 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

• **Case 2**
  • 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**
Case 1: Two kinds of Objects?

Evaluation:

\[ F_1(x, y) = w_1 \cdot \phi(x, y) \]

Training Target:

\( x^n \) is Haruhi_1

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

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:

\[
\hat{y}_1 = \arg \max_{y \in \mathbb{Y}} w_1 \cdot \phi(x,y)
\]

If the Harihu in image is Haruhi_1:

\[
\hat{y}_2 = \arg \max_{y \in \mathbb{Y}} w_2 \cdot \phi(x,y)
\]

If the Harihu in image is Haruhi_2:

Critical Problem: Given an input image, we do not know the Harihu 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**

\[ w_1 \cdot \phi(\text{Type 1}) \]

\[ \vdots \]

\[ w_2 \cdot \phi(\text{Type 1}) \]

\[ \vdots \]
Case 1: Problematic Inference

- \( w_1 \) and \( w_2 \) are learned separately

Training Target:
- If \( x^n \) 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:
- If \( x^n \) is Haruhi_2
  \[ w_2 \cdot \phi(x^n, \hat{y}^n) > w_2 \cdot \phi(x^n, y) \]
  \[ 1000000 > 999999 \]

\( w_1 \) 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

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

\[ w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \]

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

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

For “type 2”, \( F(x, y, h) = w_1 \cdot 0 + w_2 \cdot \phi(x, y) \)
Case 1: Inference

\[
\tilde{y} = \arg\max_y \max_h \mathbf{w} \cdot \Psi(x, y, h)
\]
Case 1: Training

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

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

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

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

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

\[ C^n = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - w \cdot \Psi(x^n, \hat{y}^n, \hat{h}^n) \]
Case 1: Training

\[ C^n = \max_{y} [\Delta(\hat{y}^n, y) + w \cdot \phi(x^n, y)] - w \cdot \phi(x^n, \hat{y}^n) \]
Case 1: Training

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

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

\[ \hat{h}^n = \text{“type 1”} \]
Case 1: Training

Given training data: \( \{(x^1, \hat{y}^1, \hat{h}^1), \ldots, (x^n, \hat{y}^n, \hat{h}^n), \ldots, (x^N, \hat{y}^N, \hat{h}^N)\} \)

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

Find \( w, \varepsilon^1, \ldots, \varepsilon^n, \ldots, \varepsilon^N \) minimize:

\[
\frac{1}{2} \|w\|^2 + \lambda \sum_{n=1}^{N} \varepsilon^n
\]

\( \forall n, \forall y \in Y, \forall h \in H \)

\[
w \cdot \Psi(x^n, \hat{y}^n, \hat{h}^n) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - \varepsilon^n
\]
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**

\[ F(x, y, h) = \mathbf{w} \cdot \Psi(x, y, h) \]

Random initialized
\[ \mathbf{w} = \mathbf{w}^0 \]

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

*Given* \( x \) and \( \hat{y} \), find the most compatible \( h \)
Case 2: Training with Hidden Information

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

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

\[
\tilde{h}^1 = \text{type 1} \\
\tilde{h}^2 = \text{type 2} \\
\tilde{h}^3 = \text{type 1} \\
\tilde{h}^4 = \text{type 2}
\]

For \( n = 1, \ldots, 4 \):

\[
\tilde{h}^n = \arg \max_h w_0 \cdot \Psi(x^n, \hat{y}^n, h)
\]

**Good guess?**  **Of course not.**

Because \( w_0 \) is random
Case 2: Training with Hidden Information

• With the types we generate, we can find a $w$

\[
\begin{align*}
(x^1, \hat{y}^1) & \quad \tilde{h}^1 = \text{type 1} \\
(x^2, \hat{y}^2) & \quad \tilde{h}^2 = \text{type 2} \\
(x^3, \hat{y}^3) & \quad \tilde{h}^3 = \text{type 1} \\
(x^4, \hat{y}^4) & \quad \tilde{h}^4 = \text{type 2}
\end{align*}
\]

Find $w, \varepsilon^1, \varepsilon^2, \varepsilon^3, \varepsilon^4$ minimize:

\[
\frac{1}{2} \|w\|^2 + \lambda \sum_{n=1}^{4} \varepsilon^n
\]

$n = 1, \ldots, 4, \forall y \in \mathcal{Y}, \forall h \in \mathcal{H}$

\[
w \cdot \Psi(x^n, \hat{y}^n, \hat{h}^n) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - \varepsilon^n
\]
Case 2: Training with Hidden Information

For \( n = 1, \ldots, 4 \):

\[
\tilde{h}^n = \arg \max_{\tilde{h}} w^1 \cdot \Psi(x^n, \hat{y}^n, \tilde{h})
\]

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

\( \tilde{h}^1 = \text{type 1} \)

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

\( \tilde{h}^2 = \text{type 2} \)

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

\( \tilde{h}^3 = \text{type 2} \)

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

\( \tilde{h}^4 = \text{type 2} \)

\( w^1 \)

Solving a QP

Is \( w^1 \) a good weight vector? Probably not

Train from random \( \tilde{h} \)
Case 2: Training with Hidden Information

For \( n = 1, \ldots, 4 \):

\[
\tilde{h}^n = \arg \max_h w^2 \cdot \Psi(x^n, \hat{y}^n, h)
\]

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

\( \tilde{h}^1 = \text{type 1} \)  \( \tilde{h}^2 = \text{type 1} \)  \( \tilde{h}^3 = \text{type 2} \)  \( \tilde{h}^4 = \text{type 2} \)

Is \( w^2 \) better than \( w^1 \)? Yes (?)
Case 2: Training with Hidden Information

Summary
Iteration in Iteration

Training data: \{ (x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots (x^n, \hat{y}^n) \ldots (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 \( w, \varepsilon^1, \ldots, \varepsilon^N \)

\[
\text{minimize} \quad \frac{1}{2} \|w\|^2 + \lambda \sum_{n=1}^{N} \varepsilon^n
\]

\[\forall r, \forall y \in \mathbb{Y}, \forall h \in \mathbb{H} \quad w \cdot \Psi(x^n, \hat{y}^n, \tilde{h}^n) - w \cdot \Psi(x^n, y, h) \geq \Delta(\hat{y}^n, y) - \varepsilon^n\]
Why we can get better weight vector after each iteration?

Warning of Math
Structured SVM

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

Minimizing cost

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

\[ C^n \geq \Delta(\hat{y}^n, \hat{y}^n) \]

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

What does the function \( C^n \) look like?
Structured SVM

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

\[ \Delta(\hat{y}^n, y') + w \cdot \phi(x^n, y') \]

\[ \Delta(\hat{y}^n, y'') + w \cdot \phi(x^n, y'') \]
Structured SVM

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

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

There is no local minima for structured SVM.

\[
C = \frac{1}{2} \|w\|^2 + \sum_{n=1}^{N} C^n
\]
Structured SVM with Hidden Information

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

In each iteration, the following cost is smaller

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

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

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

\[
C^n = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] \\
- \max_w \cdot \Psi(x^n, \hat{y}^n, h)
\]
Structured SVM with Hidden Information

• Cost function to be minimized

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

convex

convex
Structured SVM with Hidden Information

- Cost function to be minimized

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

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

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

- \( A(w^1) < A(w^0) \)
- \( C^n(w^1) < A(w^1) \)
- \( C^n(w^1) < C^n(w^0) \)
\[ C^n = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - \max_h w \cdot \Psi(x^n, \hat{y}^n, h) \]

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

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

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

\[ A(w^2) < A(w^1) \]

\[ C^n(w^2) < A(w^2) \]

\[ C^n(w^2) < C^n(w^1) \]

Find a \( w \) that can make \( C^n(w) \) smaller at each iteration
Can only reach local minimum
\[ C^n = \max_y \max_h [\Delta(\hat{y}^n, y) + w \cdot \Psi(x^n, y, h)] - \max_h w \cdot \Psi(x^n, \hat{y}^n, h) \]

Convex

Auxiliary function \( A(w) \)

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?

Solving a QP

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

After each iteration, the \( w \) obtained decrease the cost function
\[ \tilde{h}^n = \arg \max_h w \cdot \Psi(x^n, \hat{y}^n, h) \]

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

\[ w \cdot \Psi(x^n, \hat{y}^n, h = 1) \]

\[ w \cdot \Psi(x^n, \hat{y}^n, h = 2) \]
\[ \tilde{h}^n = \text{arg max}_{h} w \cdot \Psi(x^n, \hat{y}^n, h) \]

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

convex

\[ \tilde{h}^n = \text{arg max}_{h} w^0 \cdot \Psi(x^n, \hat{y}^n, h) \]

concave

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

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

Problem 2: Inference

Problem 3: Training

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

\[ \tilde{y} = \arg \max_y \max_h F(x, y, h) \]

**EM-like algorithm**

Find hidden information \( \leftrightarrow \) Find model parameters
To Learn More ...

• Framework
  • Chun-Nam John Yu and Thorsten Joachims, "Learning Structural SVMs with Latent Variables," ICML 2009

• Video

• Image

• Language processing
Appendix:
EM in one slide
**EM in one slide**

Problem 1: Evaluation

\[ F(x, y, h) = P(x, y, h) \]

Problem 2: Inference

\[ \tilde{y} = \arg \max_y \sum_h P(x, y, h) \]

Problem 3: Training

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

Maximizing

\[ \prod_{n=1}^{N} \sum_h P(x^n, y^n, h) \]