Part II: Structured Prediction

Alex Schwing & Raquel Urtasun

University of Toronto

December 12, 2015

Roadmap Towards Learning Deep Structured Models

1. Part I: Deep Models

2. Part II: Structured Models

3. Part III: Deep Structured Models

Part II: Structured Prediction

- How to predict in structured spaces?
- How to learn in structured spaces?

How to predict in structured spaces?

Binary Classification

Prediction answers a Yes-No question

$$x \rightarrow \quad y \in \{\text{Yes}, \text{No}\}$$

Binary Classification

Prediction answers a Yes-No question

- Spam filtering (Is this eMail spam?)

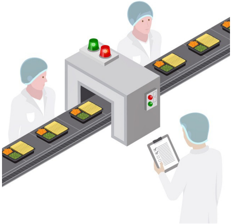
Subject: ICCV Tutorial
Text: Hey Raquel, ...

$x \rightarrow \quad y \in \{\text{Yes}, \text{No}\}$

Binary Classification

Prediction answers a Yes-No question

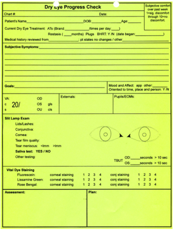
- Spam filtering (Is this eMail spam?)
- Quality control (Can I sell this product?)



$x \rightarrow \quad y \in \{\text{Yes}, \text{No}\}$

Prediction answers a Yes-No question

- Spam filtering (Is this eMail spam?)
- Quality control (Can I sell this product?)
- Medical testing (Does this lab analysis look normal?)



Dry Eye Progress Check

Subjective comfort over past week. 1=sig. discomfort through 10=no discomfort.

Chart #: Date:

Patient's Name________ DOB:________ Age:____

Current Dry Eye Treatment: ATs (Brand:________/times per day:____)

Restasis (____months) Plugs BHRT: Y /N (date began:________)

Medical history reviewed from________; pt states no changes / other________

Subjective Symptoms:

Goals:

Mood and Affect: app other:____

Oriented to time, place and person: Y /N

VA: c s 20/ OD OS OU gls cls

Externals:

Pupils/EOMs:

Slit Lamp Exam

Lids/Lashes:

Conjunctiva:

Cornea:

Tear film quality:

Tear meniscus: <1mm >1mm

Saliva test: YES / NO

Other testing:

TBUT OD:____seconds > 10 sec OS:____seconds > 10 sec

Vital Dye Staining

Fluorescein: corneal staining 1 2 3 4 conj staining 1 2 3 4

Lissamine Green: corneal staining 1 2 3 4 conj staining 1 2 3 4

Rose Bengal: corneal staining 1 2 3 4 conj staining 1 2 3 4

Assessment:

Plan:

$$x \rightarrow \quad y \in \{\text{Yes}, \text{No}\}$$

Prediction answers a Yes-No question

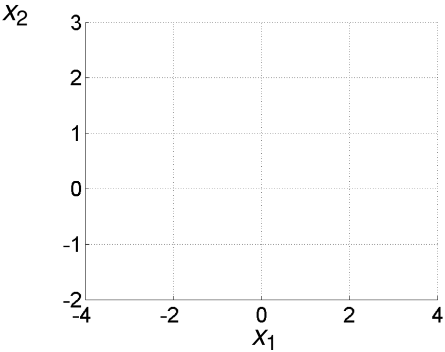
- Spam filtering (Is this eMail spam?)
- Quality control (Can I sell this product?)
- Medical testing (Does this lab analysis look normal?)
- Driver assistance systems (Is there an obstacle upfront?)

$$x \rightarrow \quad y \in \{\text{Yes}, \text{No}\}$$

Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

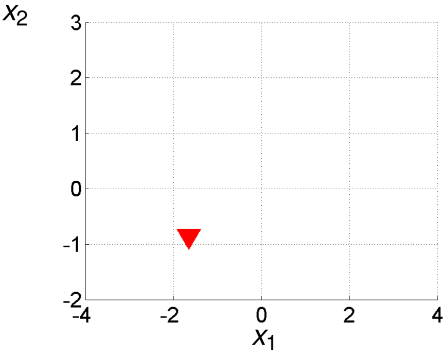
$\xrightarrow{\textit{Feature}}$



Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

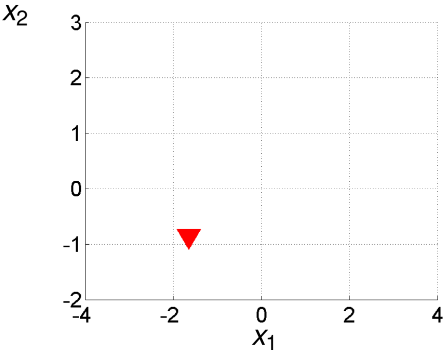
$\xrightarrow{\textit{Feature}}$



Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

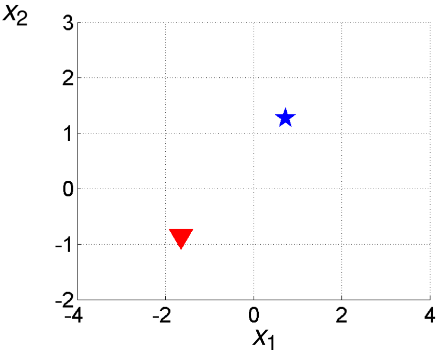
$\xrightarrow{\textit{Feature}}$



Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

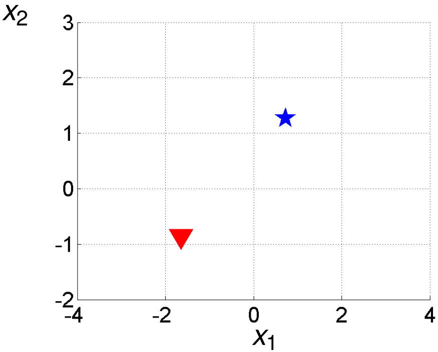
$\xrightarrow{\textit{Feature}}$



Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

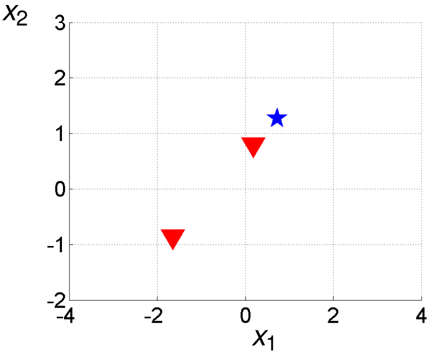
$\xrightarrow{\textit{Feature}}$



Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

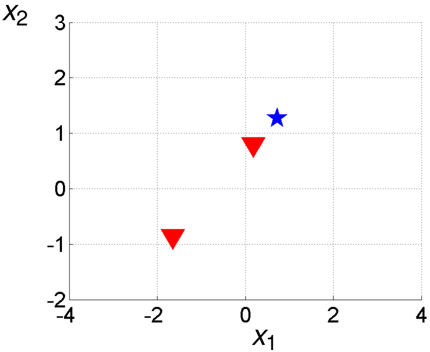
$\xrightarrow{\textit{Feature}}$



Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

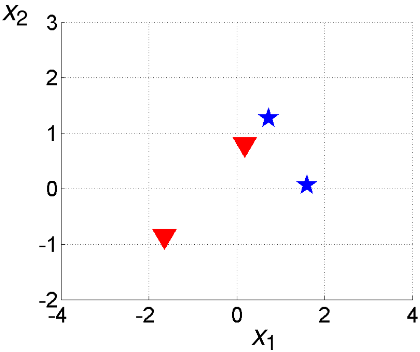
$\xrightarrow{\textit{Feature}}$



Binary Classification

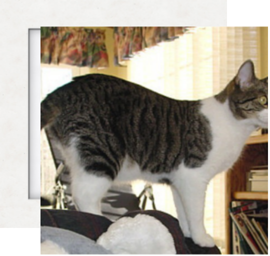
How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

$\xrightarrow{\textit{Feature}}$

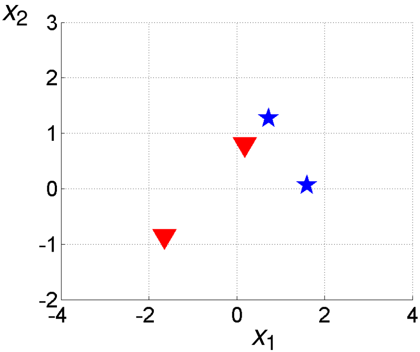


Binary Classification

How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?



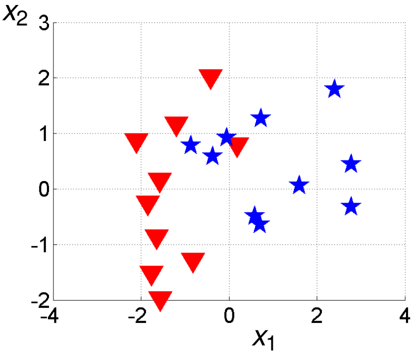
$\xrightarrow{\textit{Feature}}$



Binary Classification

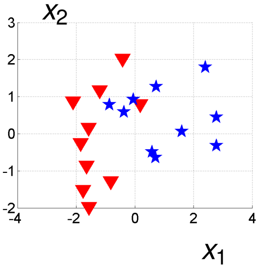
How to find $y \in \{\text{Yes}, \text{No}\}$ given input data x?

Feature $\longrightarrow$



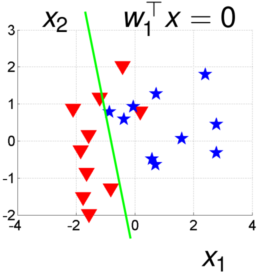
Linear Discriminant

$$y \in \{-1, 1\}$$

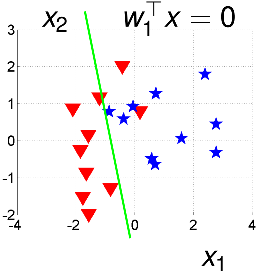


Linear Discriminant

$$y \in \{-1, 1\}$$



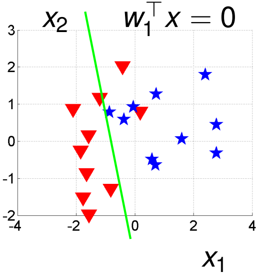
Linear Discriminant



$$y \in \{-1, 1\}$$

$$y^* = \text{sign}(w_1^\top x)$$

Linear Discriminant



$$y \in \{-1, 1\}$$

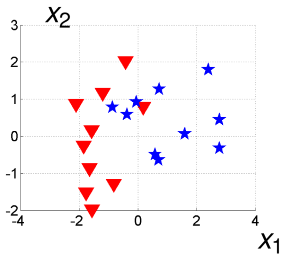
$$y^* = \text{sign}(w_1^\top x)$$

Equivalent:

$$y \in \{1, 2\}$$

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ -w_1 \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ x\delta(\hat{y} = 2) \end{bmatrix}$$

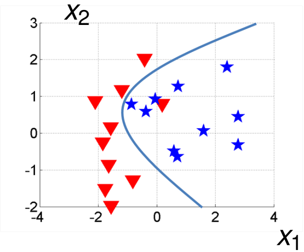
Non-linear Discriminant



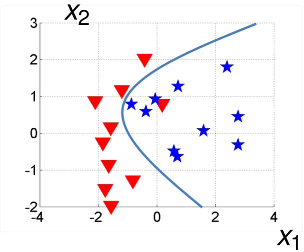
$$y \in \{1, 2\}$$

Non-linear Discriminant

$$y \in \{1, 2\}$$



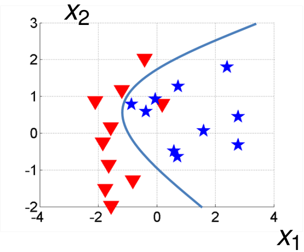
Non-linear Discriminant



$$y \in \{1, 2\}$$

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

Non-linear Discriminant



$$y \in \{1, 2\}$$

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

More generally:

$$y^* = \arg\max_{\hat{y}} F(\hat{y}, x, w)$$

Multiclass Classification

Prediction answers a categorical question

$$x \rightarrow \quad y \in \{1, \ldots, K\}$$

Multiclass Classification

Prediction answers a categorical question

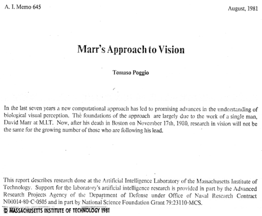
- Object classification (Which object is in the image?)

$$x \rightarrow \quad y \in \{1, \ldots, K\}$$

Multiclass Classification

Prediction answers a categorical question

- Object classification (Which object is in the image?)
- Document retrieval (What topic is this document about?)


A. I. Memo 645

August, 1981

Marr's Approach to Vision

Tomaso Poggio

In the last seven years a new computational approach has led to promising advances in the understanding of biological visual perception. The foundations of the approach are largely due to the work of a single man, David Marr at M.I.T. Now, after his death in Boston on November 17th, 1980, research in vision will not be the same for the growing number of those who are following his lead.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-80-C-0505 and in part by National Science Foundation Grant 79:23110-MCS.

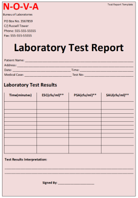
© MASSACHUSETTS INSTITUTE OF TECHNOLOGY 1981

$$x \rightarrow \quad y \in \{1, \ldots, K\}$$

Multiclass Classification

Prediction answers a categorical question

- Object classification (Which object is in the image?)
- Document retrieval (What topic is this document about?)
- Medical testing (Which disease fits the symptoms?)


N-O-V-A

Test Report Template

Bureau of Laboratories

PO Box No. 3567859
C/) Russell Tower
Phone: 555-555-55555
Fax: 555-555-55555

Laboratory Test Report

Patient Name: __________
Address: __________
Date: __________ Time: __________
Medical Case: __________ Test No: __________

Laboratory Test Results

Time(minutes)	ESC(cfu/ml)**	PSA(cfu/ml)**	SAU(cfu/ml)**

Test Results Interpretation:

Signed By: __________

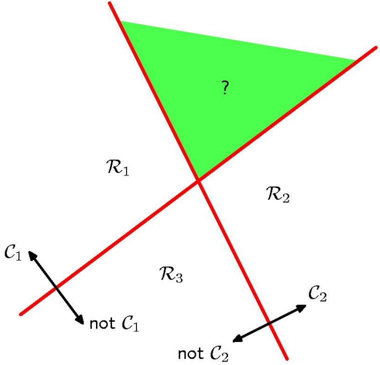
$$x \rightarrow \quad y \in \{1, \ldots, K\}$$

1 vs all

How to deal with more than two classes?

How to deal with more than two classes?

- Use $K-1$ classifiers, each solving a two class problem of separating a point in class C_k from points not in the class.
- Known as **1 vs all** or **1 vs the rest** classifier



- Issue: more than one good answer

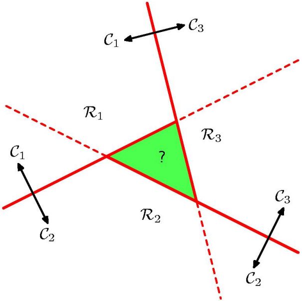
1 vs 1 classifier

How to deal with more than two classes?

1 vs 1 classifier

How to deal with more than two classes?

- Introduce $K(K-1)/2$ two-way classifiers, one for each possible pair of classes
- Each point is classified according to majority vote amongst the discriminant function
- Known as the **1 vs 1 classifier**



- Issue: two-way preferences need not be transitive

Score Maximization

$$y \in \{1, 2, \ldots, K\}$$

Prediction/Inference:

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More Generally:

Score Maximization

$$y \in \{1, 2, \ldots, K\}$$

Prediction/Inference:

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More Generally:

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

Even more generally:

Score Maximization

$$y \in \{1, 2, \ldots, K\}$$

Prediction/Inference:

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More Generally:

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

Even more generally:

$$y^* = \arg\max_{\hat{y}} F(\hat{y}, x, w)$$

Score Maximization

$$y \in \{1, 2, \ldots, K\}$$

Prediction/Inference:

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More Generally:

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

Even more generally:

$$y^* = \arg\max_{\hat{y}} F(\hat{y}, x, w)$$

Try all possible classes and return the highest scoring one.

Structured Prediction

Structured Prediction

Structured Prediction

V

Structured Prediction

Structured Prediction

I

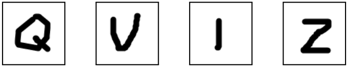
Structured Prediction

Structured Prediction

Z

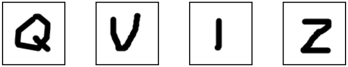
Structured Prediction

Structured Prediction

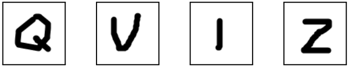


Q V I Z

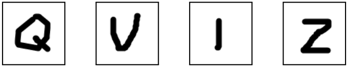
Structured Prediction



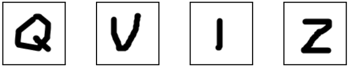
Q V I Z



Structured Prediction

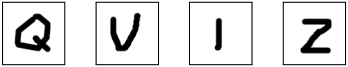


Q V I Z

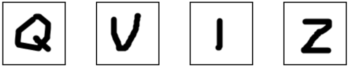


Q U I Z

Structured Prediction



Q V I Z



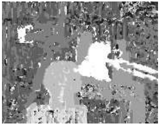
Q U I Z

Relationship not explicitly taken into account.

Structured Prediction

Example: Disparity map estimation

Why not to predict every variable separately:



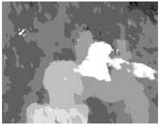


Image | Independent Prediction | Structured Prediction

Structured Prediction

Prediction estimates a complex object

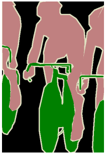
$$x \qquad \rightarrow \qquad \mathbf{y} = (y_1, \ldots, y_D)$$

Structured Prediction

Prediction estimates a complex object

- Image segmentation (estimate a labeling)

$x \quad \rightarrow \quad \mathbf{y} = (y_1, \ldots, y_D)$



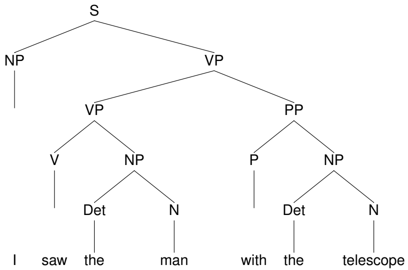
Structured Prediction

Prediction estimates a complex object

- Image segmentation (estimate a labeling)
- Sentence parsing (estimate a parse tree)

$x \quad \rightarrow \quad \mathbf{y} = (y_1, \ldots, y_D)$

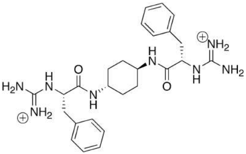
I saw the man with the telescope.

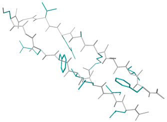


Prediction estimates a complex object

- Image segmentation (estimate a labeling)
- Sentence parsing (estimate a parse tree)
- Protein folding (estimate a protein structure)

$x \quad \rightarrow \quad \mathbf{y} = (y_1, \ldots, y_D)$



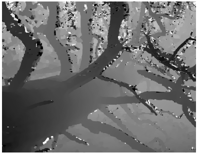


Structured Prediction

Prediction estimates a complex object

- Image segmentation (estimate a labeling)
- Sentence parsing (estimate a parse tree)
- Protein folding (estimate a protein structure)
- Stereo vision (estimate a disparity map)

$$x \quad \rightarrow \quad \mathbf{y} = (y_1, \ldots, y_D)$$



Structured Prediction

- "Standard" Prediction: output $y \in \mathcal{Y}$ is a scalar number

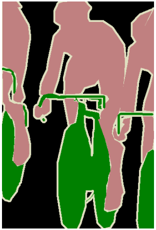
$$\mathcal{Y} = \{1, \ldots, K\} \quad \text{or} \quad \mathcal{Y} = \mathbb{R}$$

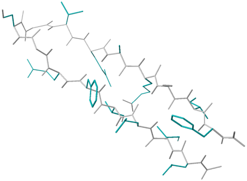
Structured Prediction

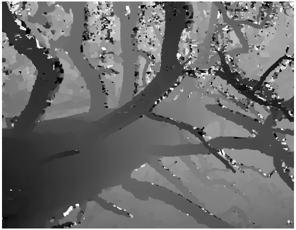
- "Standard" Prediction: output $y \in \mathcal{Y}$ is a scalar number

$$\mathcal{Y} = \{1, \ldots, K\} \quad \text{or} \quad \mathcal{Y} = \mathbb{R}$$

- "Structured" Prediction: output $\mathbf{y}$ is a structured output:







Structured Prediction

Formally:

$$\mathbf{y} = (y_1, \ldots, y_D) \qquad y_i = \{1, \ldots, K\}$$

Structured Prediction

Formally:

$$\mathbf{y} = (y_1, \ldots, y_D) \qquad y_i = \{1, \ldots, K\}$$

Inference/Prediction:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{y}_1, \ldots, \hat{y}_D, x, w)$$

How many possibilities do we have to store and explore?

Structured Prediction

Formally:

$$\mathbf{y} = (y_1, \ldots, y_D) \qquad y_i = \{1, \ldots, K\}$$

Inference/Prediction:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{y}_1, \ldots, \hat{y}_D, x, w)$$

How many possibilities do we have to store and explore?

$$K^D$$

Structured Prediction

Formally:

$$\mathbf{y} = (y_1, \ldots, y_D) \qquad y_i = \{1, \ldots, K\}$$

Inference/Prediction:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{y}_1, \ldots, \hat{y}_D, x, w)$$

How many possibilities do we have to store and explore?

$$K^D$$

That's a problem. What can we do?

Structured Prediction

Separate prediction:

$$\max_{\mathbf{y}} F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} \max_{y_i} f_i(y_i, x, w)$$

Why not predict every variable y_i from $\mathbf{y} = (y_1, \ldots, y_D)$ separately?

Structured Prediction

Separate prediction:

$$\max_{\mathbf{y}} F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} \max_{y_i} f_i(y_i, x, w)$$

Why not predict every variable y_i from $\mathbf{y} = (y_1, \ldots, y_D)$ separately?

Relationship not explicitly taken into account.

Structured Prediction

Discriminant function decomposes:

$$F(y_1, \ldots, y_D, x, w) = \sum_r f_r(\mathbf{y}_r, x, w)$$

Restriction: every $r \subseteq \{1, \ldots, D\}$

Structured Prediction

Discriminant function decomposes:

$$F(y_1, \ldots, y_D, x, w) = \sum_r f_r(\mathbf{y}_r, x, w)$$

Restriction: every $r \subseteq \{1, \ldots, D\}$
Discrete domain:

$$f_{\{1,2\}}(\mathbf{y}_{\{1,2\}}) = f_{\{1,2\}}(y_1, y_2) = \left[f_{\{1,2\}}(1, 1), f_{\{1,2\}}(1, 2), \ldots\right]$$

Structured Prediction

Discriminant function decomposes:

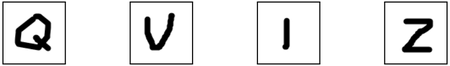
$$F(y_1, \dots, y_D, x, w) = \sum_r f_r(\mathbf{y}_r, x, w)$$

Restriction: every $r \subseteq \{1, \dots, D\}$
Discrete domain:

$$f_{\{1,2\}}(\mathbf{y}_{\{1,2\}}) = f_{\{1,2\}}(y_1, y_2) = \left[f_{\{1,2\}}(1,1), f_{\{1,2\}}(1,2), \dots\right]$$

	Q	U	I	Z	V
Q	0	0.8	0.2	0.1	0.1
U		$\ddots$	$\dots$		
I	$\vdots$				
Z					
V					

Structured Prediction

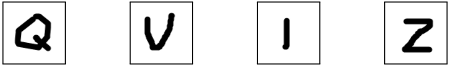


Q U I Z

Example:

$$F(y_1, \ldots, y_4, x, w) = f_1(y_1, x, w) + f_2(y_2, x, w) + f_3(y_3, x, w) + f_4(y_4, x, w) \\ + f_{1,2}(y_1, y_2, x, w) + f_{2,3}(y_2, y_3, x, w) + f_{3,4}(y_3, y_4, x, w)$$

Structured Prediction



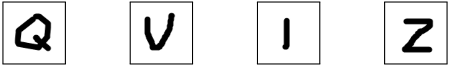
Q U I Z

Example:

$$F(y_1, \dots, y_4, x, w) = f_1(y_1, x, w) + f_2(y_2, x, w) + f_3(y_3, x, w) + f_4(y_4, x, w) \\ + f_{1,2}(y_1, y_2, x, w) + f_{2,3}(y_2, y_3, x, w) + f_{3,4}(y_3, y_4, x, w)$$

How many function values need to be stored if $y_i \in \{1, \dots, 26\}\ \forall i$?

Structured Prediction



Q U I Z

Example:

$$F(y_1,\dots,y_4,x,w) = f_1(y_1,x,w) + f_2(y_2,x,w) + f_3(y_3,x,w) + f_4(y_4,x,w)$$
$$+ f_{1,2}(y_1,y_2,x,w) + f_{2,3}(y_2,y_3,x,w) + f_{3,4}(y_3,y_4,x,w)$$

How many function values need to be stored if $y_i \in \{1,\dots,26\}\ \forall i$?

$$26^4 \qquad \text{v.s.} \qquad 3 \cdot 26^2 (+4 \cdot 26)$$

Structured Prediction

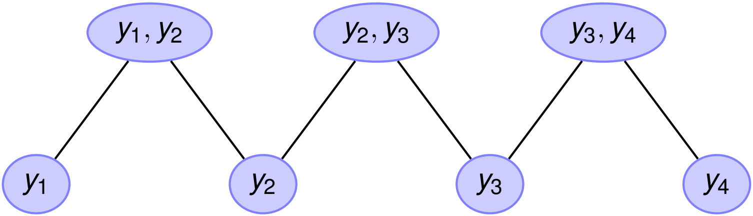
Visualization of the decomposition:

$$F(y_1,\ldots,y_4,x,w) = f_1(y_1,x,w) + f_2(y_2,x,w) + f_3(y_3,x,w) + f_4(y_4,x,w) + f_{1,2}(y_1,y_2,x,w) + f_{2,3}(y_2,y_3,x,w) + f_{3,4}(y_3,y_4,x,w)$$

Structured Prediction

Visualization of the decomposition:

$$F(y_1,\ldots,y_4,x,w) = f_1(y_1,x,w) + f_2(y_2,x,w) + f_3(y_3,x,w) + f_4(y_4,x,w) \\ + f_{1,2}(y_1,y_2,x,w) + f_{2,3}(y_2,y_3,x,w) + f_{3,4}(y_3,y_4,x,w)$$



Edges denote subset relationship

Structured Prediction

Special cases

Predicting every variable separately:

$$F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w)$$

Special cases

Predicting every variable separately:

$$F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w)$$

Structured Prediction

Special cases

Predicting every variable separately:

$$F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w)$$

Markov random field with only unary variables

Structured Prediction

Special cases

Predicting every variable separately:

$$F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w)$$

Markov random field with only unary variables

Multi-variate prediction:

$$F(y_1, \ldots, y_D, x, w) = f_{1,\ldots,D}(\mathbf{y}_{1,\ldots,D}, x, w)$$

Structured Prediction

Special cases

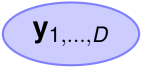
Predicting every variable separately:

$$F(y_1, \ldots, y_D, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w)$$

Markov random field with only unary variables

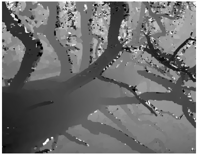
Multi-variate prediction:

$$F(y_1, \ldots, y_D, x, w) = f_{1,\ldots,D}(\mathbf{y}_{1,\ldots,D}, x, w)$$

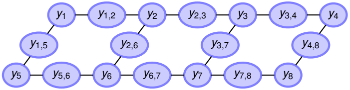


Structured Prediction

Example: stereo vision

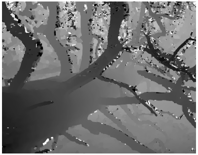


Markov/Conditional random field:

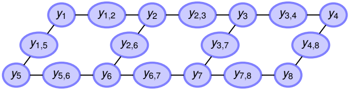


Structured Prediction

Example: stereo vision



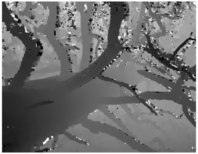
Markov/Conditional random field:



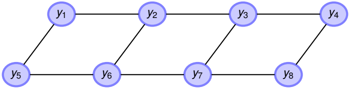
$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

Structured Prediction

Example: stereo vision



Markov/Conditional random field:

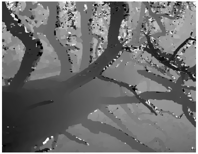


$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

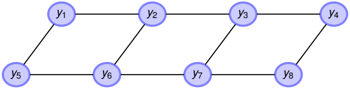
- Unary term:
- Pairwise term:

Structured Prediction

Example: stereo vision



Markov/Conditional random field:

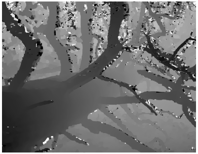


$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

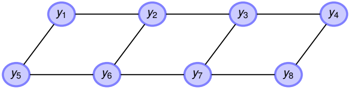
- Unary term: image evidence
- Pairwise term:

Structured Prediction

Example: stereo vision



Markov/Conditional random field:

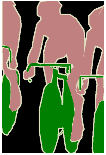


$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

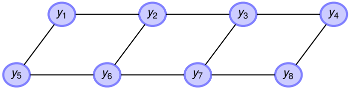
- Unary term: image evidence
- Pairwise term: smoothness prior

Structured Prediction

Example: semantic segmentation



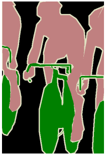
Markov/Conditional random field:



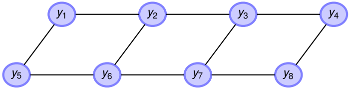
$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

Structured Prediction

Example: semantic segmentation

Markov/Conditional random field:

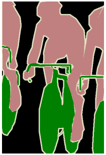


$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

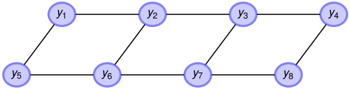
- Unary term:
- Pairwise term:

Structured Prediction

Example: semantic segmentation

Markov/Conditional random field:

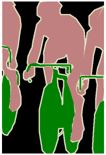


$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

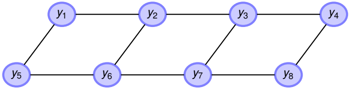
- Unary term: image evidence
- Pairwise term:

Structured Prediction

Example: semantic segmentation



Markov/Conditional random field:



$$F(\mathbf{y}, x, w) = \sum_{i=1}^{D} f_i(y_i, x, w) + \sum_{i,j} f_{i,j}(y_i, y_j, x, w)$$

- Unary term: image evidence
- Pairwise term: smoothness prior

Structured Prediction

Inference:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w)$$

Structured Prediction

Inference:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w)$$

Probability of a configuration $\mathbf{y}$:

$$p(\mathbf{y} \mid x, w) = \frac{1}{Z(x, w)} \exp F(\mathbf{y}, x, w)$$

Structured Prediction

Inference:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w)$$

Probability of a configuration $\mathbf{y}$:

$$p(\mathbf{y} \mid x, w) = \frac{1}{Z(x, w)} \exp F(\mathbf{y}, x, w)$$

Normalization constant/partition function:

$$Z(x, w) = \sum_{\mathbf{y}} \exp F(\mathbf{y}, x, w)$$

Structured Prediction

Inference:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w)$$

Probability of a configuration **y**:

$$p(\mathbf{y} \mid x, w) = \frac{1}{Z(x, w)} \exp F(\mathbf{y}, x, w)$$

Normalization constant/partition function:

$$Z(x, w) = \sum_{\mathbf{y}} \exp F(\mathbf{y}, x, w)$$

Inference as probability maximization:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}} \mid x, w)$$

Structured Prediction

Inference:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r, x, w)$$

Some inference algorithms:

Structured Prediction

Inference:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r, x, w)$$

Some inference algorithms:

- Exhaustive search
- Dynamic programming
- Integer linear program
- Linear programming relaxation
- Message passing
- Graph-cut

Structured Prediction - Exhaustive Search

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r, x, w)$$

Structured Prediction - Exhaustive Search

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r, x, w)$$

Algorithm:

- try all possible configurations $\hat{\mathbf{y}} \in \mathcal{Y}$
- keep highest scoring element

Structured Prediction - Exhaustive Search

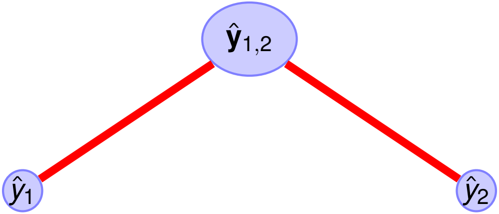
$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r, x, w)$$

Algorithm:

- try all possible configurations $\hat{\mathbf{y}} \in \mathcal{Y}$
- keep highest scoring element

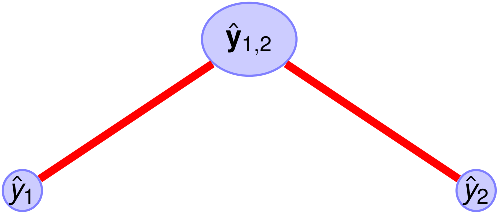
- **Advantage:** very simple to implement
- **Disadvantage:** very slow for reasonably sized problems

Structured Prediction - Dynamic Programming



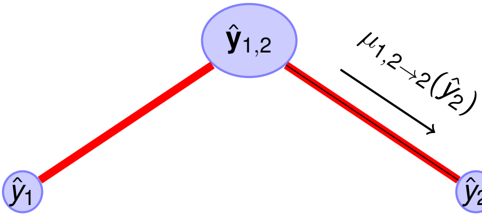
$$\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w) = \max_{\hat{y}_1, \hat{y}_2} f_2(\hat{y}_2) + f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2)$$

Structured Prediction - Dynamic Programming



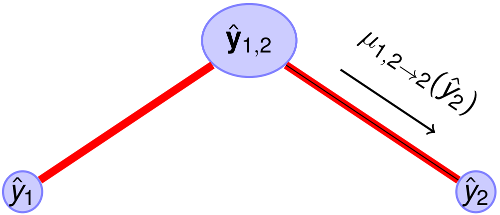
$$
\begin{aligned}
\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w) &= \max_{\hat{y}_1, \hat{y}_2} f_2(\hat{y}_2) + f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \\
&= \max_{\hat{y}_2} f_2(\hat{y}_2) + \max_{\hat{y}_1} \left\{ f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \right\}
\end{aligned}
$$

Structured Prediction - Dynamic Programming



$$
\begin{aligned}
\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w) &= \max_{\hat{y}_1, \hat{y}_2} f_2(\hat{y}_2) + f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \\
&= \max_{\hat{y}_2} f_2(\hat{y}_2) + \underbrace{\max_{\hat{y}_1} \left\{ f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \right\}}_{\mu_{1,2\to 2}(\hat{y}_2)}
\end{aligned}
$$

Structured Prediction - Dynamic Programming



$$
\begin{aligned}
\max_{\hat{\mathbf{y}}} F(\hat{\mathbf{y}}, x, w) &= \max_{\hat{y}_1,\hat{y}_2} f_2(\hat{y}_2) + f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \\
&= \max_{\hat{y}_2} f_2(\hat{y}_2) + \underbrace{\max_{\hat{y}_1} \left\{ f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \right\}}_{\mu_{1,2\to 2}(\hat{y}_2)} \\
&= \max_{\hat{y}_2} f_2(\hat{y}_2) + \mu_{1,2\to 2}(\hat{y}_2)
\end{aligned}
$$

Structured Prediction - Dynamic Programming

We can reorganize terms whenever the graph is a **tree.**

What to do for general loopy graphs?

Structured Prediction - Dynamic Programming

We can reorganize terms whenever the graph is a **tree.**

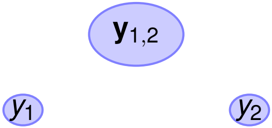
What to do for general loopy graphs?

- Dynamic programming extensions (message passing)
- Graph cut algorithms

Structured Prediction - Integer Linear Program

Example:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$



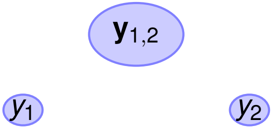
Integer Linear Program (LP) equivalence: variables $b_r(\mathbf{y}_r)$

$$\max_{b_1,b_2,b_{12}} \begin{bmatrix} b_1(1) \\ b_1(2) \\ b_2(1) \\ b_2(2) \\ b_{12}(1,1) \\ b_{12}(2,1) \\ b_{12}(1,2) \\ b_{12}(2,2) \end{bmatrix}^\top \begin{bmatrix} f_1(1) \\ f_1(2) \\ f_2(1) \\ f_2(2) \\ f_{12}(1,1) \\ f_{12}(2,1) \\ f_{12}(1,2) \\ f_{12}(2,2) \end{bmatrix}$$

Structured Prediction - Integer Linear Program

Example:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$



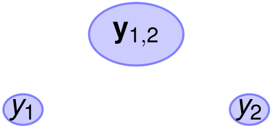
Integer Linear Program (LP) equivalence: variables $b_r(\mathbf{y}_r)$

$$\max_{b_1,b_2,b_{12}} \begin{bmatrix} b_1(1) \\ b_1(2) \\ b_2(1) \\ b_2(2) \\ b_{12}(1,1) \\ b_{12}(2,1) \\ b_{12}(1,2) \\ b_{12}(2,2) \end{bmatrix}^\top \begin{bmatrix} f_1(1) \\ f_1(2) \\ f_2(1) \\ f_2(2) \\ f_{12}(1,1) \\ f_{12}(2,1) \\ f_{12}(1,2) \\ f_{12}(2,2) \end{bmatrix} \quad \text{s.t.} \quad b_r(\mathbf{y}_r) \in \{0,1\}$$

Structured Prediction - Integer Linear Program

Example:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$



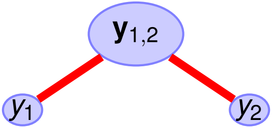
Integer Linear Program (LP) equivalence: variables $b_r(\mathbf{y}_r)$

$$\max_{b_1, b_2, b_{12}} \begin{bmatrix} b_1(1) \\ b_1(2) \\ b_2(1) \\ b_2(2) \\ b_{12}(1,1) \\ b_{12}(2,1) \\ b_{12}(1,2) \\ b_{12}(2,2) \end{bmatrix}^\top \begin{bmatrix} f_1(1) \\ f_1(2) \\ f_2(1) \\ f_2(2) \\ f_{12}(1,1) \\ f_{12}(2,1) \\ f_{12}(1,2) \\ f_{12}(2,2) \end{bmatrix} \quad \text{s.t.} \quad \begin{array}{c} b_r(\mathbf{y}_r) \in \{0, 1\} \\ \sum_{\mathbf{y}_r} b_r(\mathbf{y}_r) = 1 \end{array}$$

Structured Prediction - Integer Linear Program

Example:

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$



Integer Linear Program (LP) equivalence: variables $b_r(\mathbf{y}_r)$

$$\max_{b_1,b_2,b_{12}} \begin{bmatrix} b_1(1) \\ b_1(2) \\ b_2(1) \\ b_2(2) \\ b_{12}(1,1) \\ b_{12}(2,1) \\ b_{12}(1,2) \\ b_{12}(2,2) \end{bmatrix}^\top \begin{bmatrix} f_1(1) \\ f_1(2) \\ f_2(1) \\ f_2(2) \\ f_{12}(1,1) \\ f_{12}(2,1) \\ f_{12}(1,2) \\ f_{12}(2,2) \end{bmatrix} \quad \text{s.t.} \quad \begin{array}{l} b_r(\mathbf{y}_r) \in \{0,1\} \\ \sum_{\mathbf{y}_r} b_r(\mathbf{y}_r) = 1 \\ \sum_{\mathbf{y}_p \setminus \mathbf{y}_r} b_p(\mathbf{y}_p) = b_r(\mathbf{y}_r) \end{array}$$

Structured Prediction - Integer Linear Program

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

Integer linear program:

$$\max_{b_1,b_2,b_{12}} \begin{bmatrix} b_1(1) \\ b_1(2) \\ b_2(1) \\ b_2(2) \\ b_{12}(1,1) \\ b_{12}(2,1) \\ b_{12}(1,2) \\ b_{12}(2,2) \end{bmatrix}^\top \begin{bmatrix} f_1(1) \\ f_1(2) \\ f_2(1) \\ f_2(2) \\ f_{12}(1,1) \\ f_{12}(2,1) \\ f_{12}(1,2) \\ f_{12}(2,2) \end{bmatrix} \quad \text{s.t.} \quad \begin{array}{l} b_r(\mathbf{y}_r) \in \{0,1\} \\ b_r(\mathbf{y}_r) \geq 0 \\ \sum_{\mathbf{y}_r} b_r(\mathbf{y}_r) = 1 \\ \sum_{\mathbf{y}_p \setminus \mathbf{y}_r} b_p(\mathbf{y}_p) = b_r(\mathbf{y}_r) \end{array}$$

Structured Prediction - Integer Linear Program

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

Integer linear program:

$$\max_{b_r} \quad \sum_{r,\mathbf{y}_r} b_r(\mathbf{y}_r) f_r(\mathbf{y}_r) \quad \text{s.t.} \quad \begin{array}{l} b_r(\mathbf{y}_r) \in \{0, 1\} \\ b_r(\mathbf{y}_r) \geq 0 \\ \sum_{\mathbf{y}_r} b_r(\mathbf{y}_r) = 1 \\ \sum_{\mathbf{y}_p \backslash \mathbf{y}_r} b_p(\mathbf{y}_p) = b_r(\mathbf{y}_r) \end{array}$$

Structured Prediction - Integer Linear Program

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

Integer linear program:

$$\max_{b_r} \quad \sum_{r,\mathbf{y}_r} b_r(\mathbf{y}_r) f_r(\mathbf{y}_r) \quad \text{s.t.} \quad \begin{array}{l} b_r(\mathbf{y}_r) \in \{0,1\} \\ b_r(\mathbf{y}_r) \geq 0 \\ \sum_{\mathbf{y}_r} b_r(\mathbf{y}_r) = 1 \\ \text{Marginalization} \end{array}$$

Structured Prediction - Integer Linear Program

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

Integer linear program:

$$b_r(\mathbf{y}_r) \in \{0, 1\}$$

$$\max_{b_r} \quad \sum_{r,\mathbf{y}_r} b_r(\mathbf{y}_r) f_r(\mathbf{y}_r) \quad \text{s.t.}$$

Local probability b_r

Marginalization

Structured Prediction - Integer Linear Program

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

Integer linear program:

$$\max_{b_r} \quad \sum_{r,\mathbf{y}_r} b_r(\mathbf{y}_r) f_r(\mathbf{y}_r) \quad \text{s.t.} \quad \begin{array}{l} b_r(\mathbf{y}_r) \in \{0,1\} \\ \text{Local probability } b_r \\ \text{Marginalization} \end{array}$$

- **Advantage:** very good solvers available
- **Disadvantage:** very slow for larger problems

Structured Prediction - Linear Programming Relaxation

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

LP relaxation:

$$\cancel{b_r(\mathbf{y}_r) \in \{0,1\}}$$

$$\max_{b_r} \quad \sum_{r,\mathbf{y}_r} b_r(\mathbf{y}_r) f_r(\mathbf{y}_r) \quad \text{s.t.} \quad \underbrace{\begin{array}{l}\text{Local probability } b_r \\ \text{Marginalization}\end{array}}_{\text{s.t.} \quad b \in \mathcal{C}}$$

Structured Prediction - Linear Programming Relaxation

$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r)$$

LP relaxation:

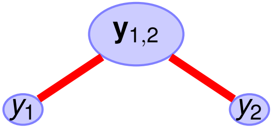
$$\cancel{b_r(\mathbf{y}_r) \in \{0,1\}}$$

$$\max_{b_r} \quad \sum_{r,\mathbf{y}_r} b_r(\mathbf{y}_r) f_r(\mathbf{y}_r) \quad \underbrace{\text{s.t.} \quad \begin{array}{l} \text{Local probability } b_r \\ \text{Marginalization} \end{array}}_{\text{s.t.} \quad b \in \mathcal{C}}$$

- **Advantage:** very good solvers available
- **Disadvantage:** slow for larger problems

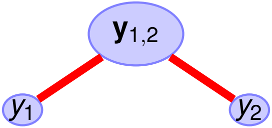
Structured Prediction - Message Passing

Graph structure defined via marginalization constraints



Structured Prediction - Message Passing

Graph structure defined via marginalization constraints



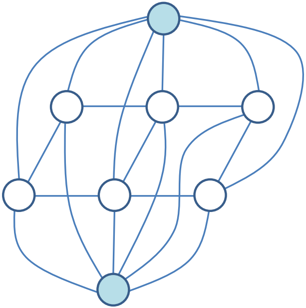
Message passing solvers:

Advantage:	Efficient due to analytically computable sub-problems
Problem:	Special care required to find global optimum

Structured Prediction - Graph-cut Solvers

For $y_i \in \{1, 2\}$:

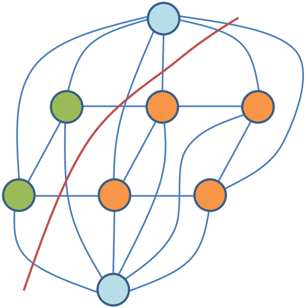
- Convert scoring function F into auxiliary graph



Structured Prediction - Graph-cut Solvers

For $y_i \in \{1, 2\}$:

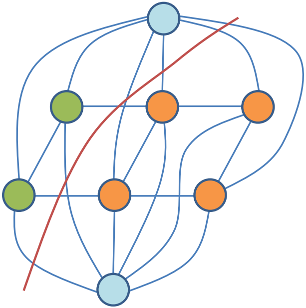
- Convert scoring function F into auxiliary graph
- Compute a weighted cut cost corresponding to the labeling score



Structured Prediction - Graph-cut Solvers

For $y_i \in \{1, 2\}$:

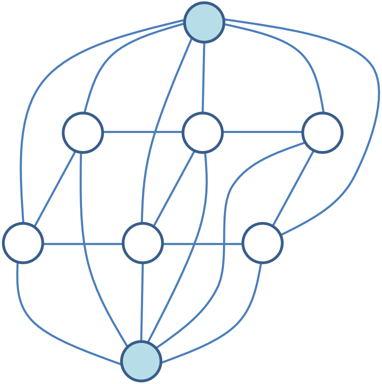
- Convert scoring function F into auxiliary graph
- Compute a weighted cut cost corresponding to the labeling score



What are the nodes and what are the weights on the edges?

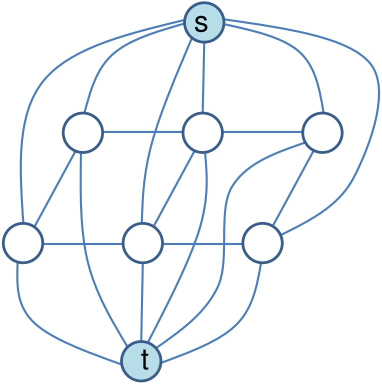
Structured Prediction - Graph-cut Solvers

What are the nodes?



Structured Prediction - Graph-cut Solvers

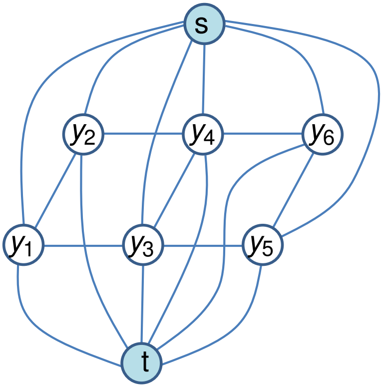
What are the nodes?



- Two special nodes called ‘source’ and ‘terminal’

Structured Prediction - Graph-cut Solvers

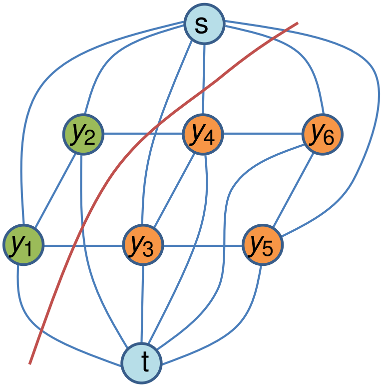
What are the nodes?



- Two special nodes called ‘source’ and ‘terminal’
- Variables y_i as nodes

Structured Prediction - Graph-cut Solvers

What are the nodes?



- Two special nodes called ‘source’ and ‘terminal’
- Variables y_i as nodes

Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?

Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

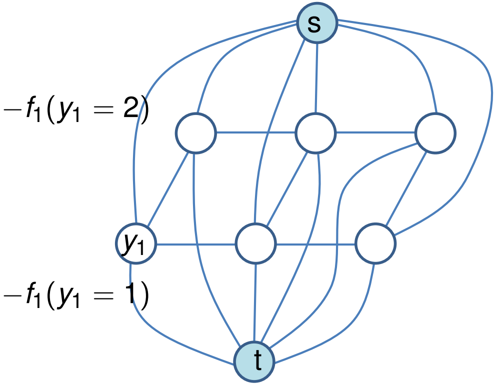
$$\begin{bmatrix} f_1(y_1 = 1) & f_1(y_1 = 2) \end{bmatrix}$$

Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

$$\left[\begin{array}{cc} f_1(y_1 = 1) & f_1(y_1 = 2) \end{array}\right]$$

Graph-cut solvers compute a min-cut:



Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

$$\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix} =$$

Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

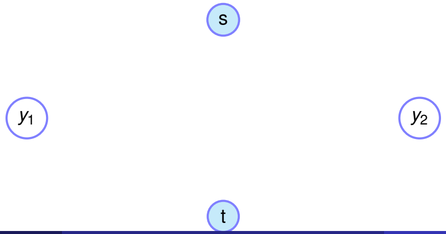
$$\begin{aligned}
\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix} &= f(1,1) - f(2,1) + f(2,2) \\
&+ \begin{bmatrix} 0 & 0 \\ f(2,1) - f(1,1) & f(2,1) - f(1,1) \end{bmatrix} \\
&+ \begin{bmatrix} f(2,1) - f(2,2) & 0 \\ f(2,1) - f(2,2) & 0 \end{bmatrix} \\
&+ \begin{bmatrix} 0 & f(1,2) + f(2,1) - f(1,1) - f(2,2) \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

$$\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix} = f(1,1) - f(2,1) + f(2,2)$$
$$+ \begin{bmatrix} 0 & 0 \\ f(2,1) - f(1,1) & f(2,1) - f(1,1) \end{bmatrix}$$
$$+ \begin{bmatrix} f(2,1) - f(2,2) & 0 \\ f(2,1) - f(2,2) & 0 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & f(1,2) + f(2,1) - f(1,1) - f(2,2) \\ 0 & 0 \end{bmatrix}$$

Graph-cut solvers compute a min-cut:

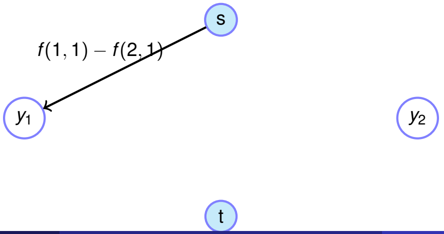


Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

$$\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix} = f(1,1) - f(2,1) + f(2,2)$$
$$+ \begin{bmatrix} 0 & 0 \\ f(2,1)-f(1,1) & f(2,1)-f(1,1) \end{bmatrix}$$
$$+ \begin{bmatrix} f(2,1)-f(2,2) & 0 \\ f(2,1)-f(2,2) & 0 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & f(1,2)+f(2,1)-f(1,1)-f(2,2) \\ 0 & 0 \end{bmatrix}$$

Graph-cut solvers compute a min-cut:

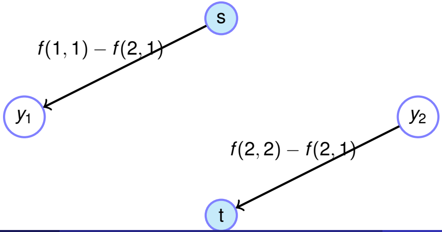


Structured Prediction - Graph-cut Solvers

What weights do we assign to edges?
Recall that local scoring functions are arrays:

$$
\begin{aligned}
\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix} &= f(1,1) - f(2,1) + f(2,2) \\
&+ \begin{bmatrix} 0 & 0 \\ f(2,1) - f(1,1) & f(2,1) - f(1,1) \end{bmatrix} \\
&+ \begin{bmatrix} f(2,1) - f(2,2) & 0 \\ f(2,1) - f(2,2) & 0 \end{bmatrix} \\
&+ \begin{bmatrix} 0 & f(1,2) + f(2,1) - f(1,1) - f(2,2) \\ 0 & 0 \end{bmatrix}
\end{aligned}
$$

Graph-cut solvers compute a min-cut:

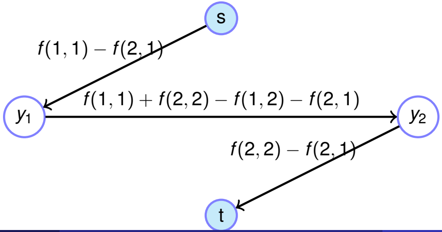


Structured Prediction - Graph-cut Solvers

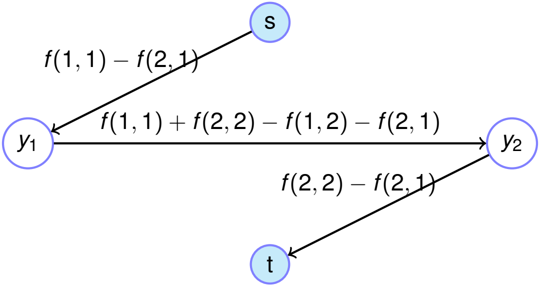
What weights do we assign to edges?
Recall that local scoring functions are arrays:

$$\begin{bmatrix} f_{12}(1,1) & f_{12}(1,2) \\ f_{12}(2,1) & f_{12}(2,2) \end{bmatrix} = f(1,1) - f(2,1) + f(2,2)$$
$$+ \begin{bmatrix} 0 & 0 \\ f(2,1) - f(1,1) & f(2,1) - f(1,1) \end{bmatrix}$$
$$+ \begin{bmatrix} f(2,1) - f(2,2) & 0 \\ f(2,1) - f(2,2) & 0 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & f(1,2) + f(2,1) - f(1,1) - f(2,2) \\ 0 & 0 \end{bmatrix}$$

Graph-cut solvers compute a min-cut:

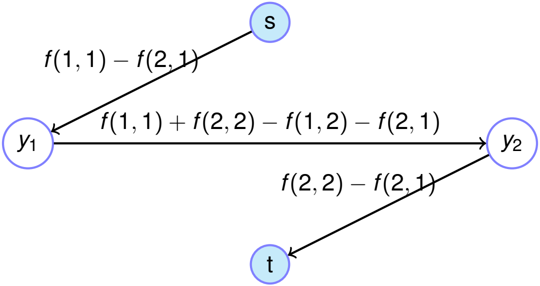


Structured Prediction - Graph-cut Solvers



Requirement for optimality:

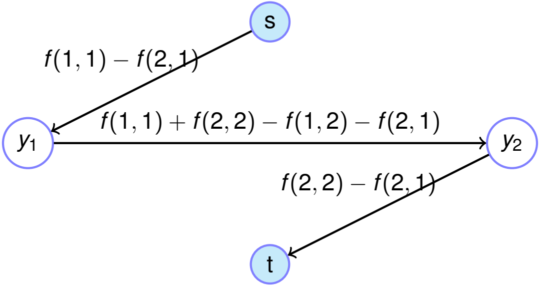
Structured Prediction - Graph-cut Solvers



Requirement for optimality: Pairwise edge weights are positive

$$f(1,1) + f(2,2) - f(1,2) - f(2,1) \geq 0 \quad \text{sub-modularity}$$

Structured Prediction - Graph-cut Solvers

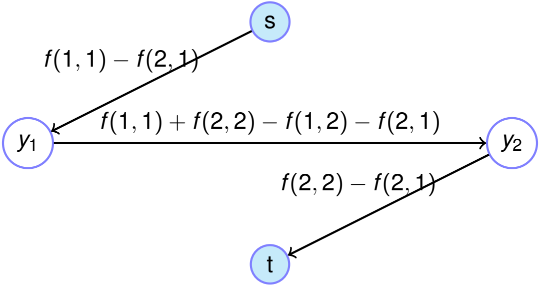


Requirement for optimality: Pairwise edge weights are positive

$$f(1,1) + f(2,2) - f(1,2) - f(2,1) \geq 0 \quad \text{sub-modularity}$$

For higher order functions?

Structured Prediction - Graph-cut Solvers

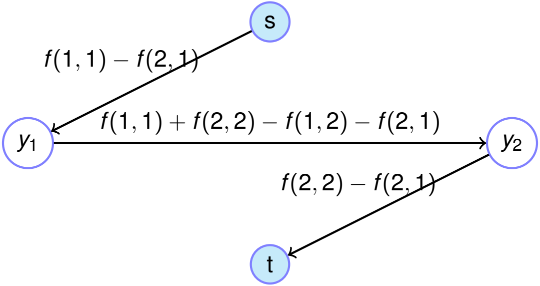


Requirement for optimality: Pairwise edge weights are positive

$$f(1,1) + f(2,2) - f(1,2) - f(2,1) \geq 0 \quad \text{sub-modularity}$$

For higher order functions? More complicated graph constructions

Structured Prediction - Graph-cut Solvers

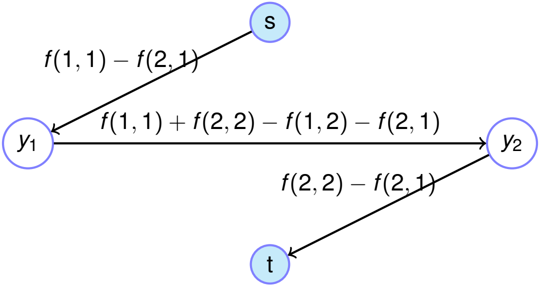


Requirement for optimality: Pairwise edge weights are positive

$$f(1,1) + f(2,2) - f(1,2) - f(2,1) \geq 0 \quad \text{sub-modularity}$$

For higher order functions? More complicated graph constructions
For more than two labels?

Structured Prediction - Graph-cut Solvers



Requirement for optimality: Pairwise edge weights are positive

$$f(1,1) + f(2,2) - f(1,2) - f(2,1) \geq 0 \quad \text{sub-modularity}$$

For higher order functions? More complicated graph constructions
For more than two labels? Move making algorithms

Structured Prediction

Inference:

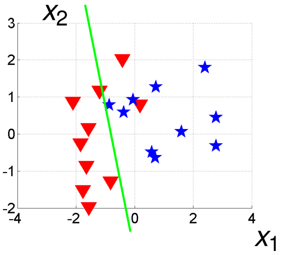
$$\mathbf{y}^* = \arg\max_{\hat{\mathbf{y}}} \sum_r f_r(\hat{\mathbf{y}}_r, x, w)$$

Efficiency and accuracy of inference algorithms is problem dependent:

- Exhaustive search
- Dynamic programming
- Integer linear program
- Linear programming relaxation
- Message passing
- Graph-cut

How to learn in structured spaces?

Binary SVM



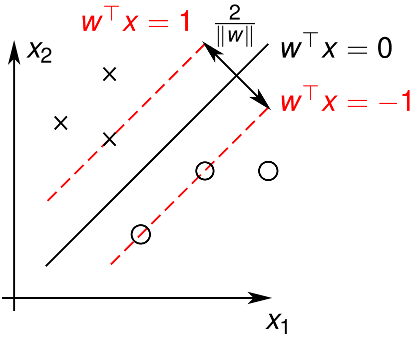
Given: dataset

$$\mathcal{D} = \{(x, y)\}, \qquad N = |\mathcal{D}|$$

containing independent and identically drawn pairs

Binary SVM

Intuitively:

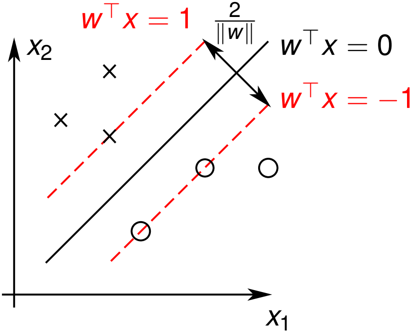


Maximize margin $\frac{2}{\|w\|}$:

$$\min_w \frac{1}{2}\|w\|_2^2 \qquad \text{s.t.} \qquad yw^\top x \geq 1 \qquad \forall (x, y) \in \mathcal{D}$$

Binary SVM

Intuitively:



Maximize margin $\frac{2}{\|w\|}$:

$$\min_w \frac{1}{2}\|w\|_2^2 \qquad \text{s.t.} \qquad yw^\top x \geq 1 \qquad \forall (x, y) \in \mathcal{D}$$

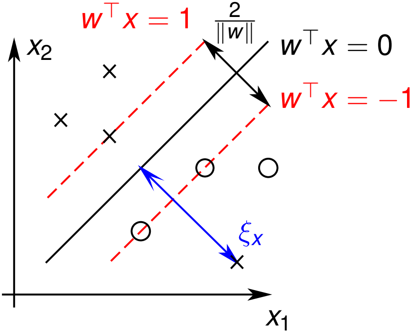
Issue: what if data not linearly separable?

Binary SVM

Introduce slack variables ξ:

$$\min_{w,\xi_x \geq 0} \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,y)\in\mathcal{D}} \xi_x \qquad \text{s.t.} \qquad y w^\top x \geq 1 - \xi_x \quad \forall (x,y) \in \mathcal{D}$$

Intuitively:



Binary SVM

$$\min_{w,\xi_x \geq 0} \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,y)\in\mathcal{D}} \xi_x \qquad \text{s.t.} \qquad y w^\top x \geq 1 - \xi_x \quad \forall (x,y) \in \mathcal{D}$$

Binary SVM

$$\min_{w,\xi_x\geq 0} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \xi_x \qquad \text{s.t.} \qquad yw^\top x \geq 1-\xi_x \quad \forall (x,y)\in\mathcal{D}$$

Equivalent:

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \max\{0, 1-yw^\top x\}$$

Binary SVM

$$\min_{w,\xi_x\geq 0} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \xi_x \qquad \text{s.t.} \qquad yw^\top x \geq 1-\xi_x \quad \forall (x,y)\in\mathcal{D}$$

Equivalent:

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \max\{0, 1-yw^\top x\}$$

Generally:

$$\min_{w} R(w) + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \bar{\ell}(x,y,w)$$

Binary SVM

$$\min_{w,\xi_x\geq 0} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \xi_x \qquad \text{s.t.} \qquad yw^\top x \geq 1-\xi_x \quad \forall (x,y)\in\mathcal{D}$$

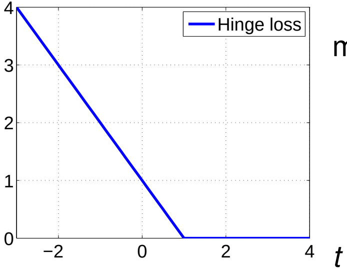
Equivalent:

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \max\{0, 1-yw^\top x\}$$

Generally:

$$\min_{w} R(w) + \frac{C}{N}\sum_{(x,y)\in\mathcal{D}} \bar{\ell}(x,y,w)$$

Hinge-Loss $\bar{\ell}$:



$$\max\{0, 1-t\}$$

Multiclass SVM

Multiclass SVM

Recap of Inference:

$$y \in \{1, 2, \ldots, K\}$$

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More generally:

Multiclass SVM

Recap of Inference:

$$y \in \{1, 2, \ldots, K\}$$

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More generally:

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

Even more generally:

Multiclass SVM

Recap of Inference:

$$y \in \{1, 2, \ldots, K\}$$

$$y^* = \arg\max_{\hat{y}} \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^\top \begin{bmatrix} x\delta(\hat{y} = 1) \\ \vdots \\ x\delta(\hat{y} = K) \end{bmatrix}$$

More generally:

$$y^* = \arg\max_{\hat{y}} w^\top \phi(x, \hat{y})$$

Even more generally:

$$y^* = \arg\max_{\hat{y}} F(\hat{y}, x, w)$$

Multiclass SVM

What do we want?

Multiclass SVM

What do we want? Groundtruth y scores higher than any other $\hat{y}$

Multiclass SVM

What do we want? Groundtruth y scores higher than any other $\hat{y}$

$$w^\top \phi(x, y) \geq w^\top \phi(x, \hat{y}) \qquad \forall (x, y) \in \mathcal{D}, \hat{y} \in \{1, \dots, K\}$$

Multiclass SVM

What do we want? Groundtruth y scores higher than any other $\hat{y}$

$$w^\top \phi(x, y) \geq w^\top \phi(x, \hat{y}) \qquad \forall (x, y) \in \mathcal{D}, \hat{y} \in \{1, \dots, K\}$$

Let's employ margin and slack:

$$\begin{aligned} \min_{w, \xi_x \geq 0} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x \\ \text{s.t.} \quad & w^\top (\phi(x, y) - \phi(x, \hat{y})) \geq 1 - \xi_x \qquad \forall (x, y), \hat{y} \end{aligned}$$

Multiclass SVM

$$\min_{w,\xi_x \geq 0} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x$$

$$\text{s.t.} \quad w^\top \left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y), \hat{y}$$

Properties:

Multiclass SVM

$$\min_{w,\xi_x \geq 0} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x$$

$$\text{s.t.} \quad w^\top \left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y),\hat{y}$$

Properties:

- Quadratic cost function
- Linear constraints
- Convex program

Algorithms:

Multiclass SVM

$$\min_{w,\xi_x \geq 0} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x$$

$$\text{s.t.} \quad w^\top \left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y), \hat{y}$$

Properties:

- Quadratic cost function
- Linear constraints
- Convex program

Algorithms:

- Primal algorithms (Sub-gradient, cutting plane)
- Dual algorithms

Multiclass SVM

$$\begin{aligned} \min_{w,\xi_x \geq 0} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x \\ \text{s.t.} \quad & w^\top (\phi(x,y) - \phi(x,\hat{y})) \geq 1 - \xi_x \qquad \forall (x,y), \hat{y} \end{aligned}$$

Properties:

- Quadratic cost function
- Linear constraints
- Convex program

Algorithms:

- Primal algorithms (Sub-gradient, cutting plane)
- Dual algorithms

How does this fit into regularized surrogate loss minimization?

$$\min_w R(w) + \frac{C}{N}\sum_{(x,y)} \bar{\ell}(x,y,w)$$

Multiclass SVM

$$\begin{aligned} \min_{w,\xi_x\geq 0} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x \\ \text{s.t.} \quad & w^\top\left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y),\hat{y} \end{aligned}$$

Let's get rid of the slack variable:

Multiclass SVM

$$\begin{aligned}
\min_{w,\xi_x\geq 0} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x \\
\text{s.t.} \quad & w^\top\left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y), \hat{y}
\end{aligned}$$

Let's get rid of the slack variable:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max\left\{0, \max_{\hat{y}}\left(1 - w^\top(\phi(x,y) - \phi(x,\hat{y}))\right)\right\}$$

Multiclass SVM

$$\begin{aligned} \min_{w,\xi_x\geq 0} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x \\ \text{s.t.} \quad & w^\top\left(\phi(x,y)-\phi(x,\hat{y})\right) \geq 1-\xi_x \qquad \forall (x,y),\hat{y} \end{aligned}$$

Let's get rid of the slack variable:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max\left\{0, \underbrace{\max_{\hat{y}}\left(1 - w^\top\left(\phi(x,y)-\phi(x,\hat{y})\right)\right)}_{\geq 0}\right\}$$

Multiclass SVM

$$\begin{aligned} \min_{w,\xi_x \geq 0} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x \\ \text{s.t.} \quad & w^\top \left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y), \hat{y} \end{aligned}$$

Let's get rid of the slack variable:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max\left\{0, \underbrace{\max_{\hat{y}}\left(1 - w^\top\left(\phi(x,y) - \phi(x,\hat{y})\right)\right)}_{\geq 0}\right\}$$

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max_{\hat{y}}\left(1 + w^\top\phi(x,\hat{y})\right) - w^\top\phi(x,y)$$

Multiclass SVM

$$\min_{w,\xi_x \geq 0} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \xi_x$$

$$\text{s.t.} \quad w^\top \left(\phi(x,y) - \phi(x,\hat{y})\right) \geq 1 - \xi_x \qquad \forall (x,y), \hat{y}$$

Let's get rid of the slack variable:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max\left\{0, \underbrace{\max_{\hat{y}} \left(1 - w^\top \left(\phi(x,y) - \phi(x,\hat{y})\right)\right)}_{\geq 0}\right\}$$

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x,y)$$

Multiclass SVM

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x,y)$$

Fits into the general formulation:

Multiclass SVM

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x,y)$$

Fits into the general formulation:

$$\min_w R(w) + \frac{C}{N}\sum_{(x,y)} \bar{\ell}(x,y,w)$$

Multiclass SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x,y)$$

E.g., with gradient descent:

Iterate:

Multiclass SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x,y)$$

E.g., with gradient descent:

Iterate:

1. Loss-augmented inference:

$$\arg\max_{\hat{y}_i} \left(1 + w^\top \phi(x,\hat{y}_i)\right)$$

Multiclass SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x, \hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x, y)$$

E.g., with gradient descent:

Iterate:

1. Loss-augmented inference:

$$\arg\max_{\hat{y}_i} \left(1 + w^\top \phi(x, \hat{y}_i)\right)$$

2. Perform gradient step:

$$w \leftarrow w - \alpha \nabla_w L$$

Multiclass SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \underbrace{\max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right)}_{\text{Loss-augmented inference}} - w^\top \phi(x,y)$$

E.g., with gradient descent:

Iterate:

1. Loss-augmented inference:

$$\arg\max_{\hat{y}_i} \left(1 + w^\top \phi(x,\hat{y}_i)\right)$$

2. Perform gradient step:

$$w \leftarrow w - \alpha \nabla_w L$$

How complicated is this?

Solve one multi-class inference task per sample per iteration.

Structured SVM

Structured SVM

From Multiclass SVM:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right) - w^\top \phi(x,y)$$

Structured SVM

From Multiclass SVM:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right) - w^\top \phi(x,y)$$

via general losses:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max_{\hat{y}} \left(\Delta(y,\hat{y}) + w^\top \phi(x,\hat{y})\right) - w^\top \phi(x,y)$$

Structured SVM

From Multiclass SVM:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max_{\hat{y}} \left(1 + w^\top \phi(x,\hat{y})\right) - w^\top \phi(x,y)$$

via general losses:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)} \max_{\hat{y}} \left(\Delta(y,\hat{y}) + w^\top \phi(x,\hat{y})\right) - w^\top \phi(x,y)$$

to general functions and structures:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})} \underbrace{\max_{\hat{\mathbf{y}}} \left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)}_{\text{Loss-augmented inference}} - F(\mathbf{y},x,w)$$

Structured SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \underbrace{\max_{\hat{\mathbf{y}}}\left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)}_{\text{Loss-augmented inference}} - F(\mathbf{y},x,w)$$

E.g., with gradient descent:

Iterate:

Structured SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \underbrace{\max_{\hat{\mathbf{y}}}\left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)}_{\text{Loss-augmented inference}} - F(\mathbf{y},x,w)$$

E.g., with gradient descent:

Iterate:

1. Loss-augmented inference:

$$\arg\max_{\hat{\mathbf{y}}}\left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)$$

Structured SVM

How to optimize this?

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \underbrace{\max_{\hat{\mathbf{y}}}\left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)}_{\text{Loss-augmented inference}} - F(\mathbf{y},x,w)$$

E.g., with gradient descent:

Iterate:

1. Loss-augmented inference:

$$\arg\max_{\hat{\mathbf{y}}}\left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)$$

2. Perform gradient step:

$$w \leftarrow w - \alpha\nabla_w L$$

Structured SVM

How to optimize this?

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,\mathbf{y})\in\mathcal{D}} \underbrace{\max_{\hat{\mathbf{y}}} \left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)}_{\text{Loss-augmented inference}} - F(\mathbf{y},x,w)$$

E.g., with gradient descent:

Iterate:

1. Loss-augmented inference:
$$\arg\max_{\hat{\mathbf{y}}} \left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right)$$
2. Perform gradient step:
$$w \leftarrow w - \alpha \nabla_w L$$

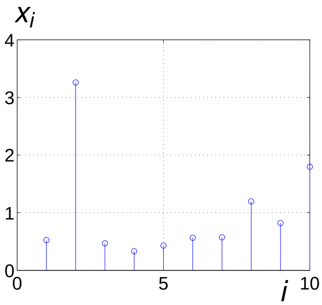
How complicated is this?

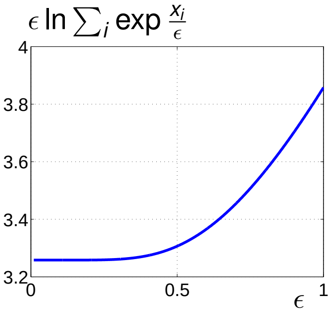
Solve one structured prediction task per sample per iteration.

Structured Learning

Soft-max extension:

$$\epsilon \ln \sum_i \exp \frac{x_i}{\epsilon} \quad \xrightarrow{\epsilon \to 0} \quad \max_i x_i$$





Soft-max extension:

$$\epsilon \ln \sum_i \exp \frac{x_i}{\epsilon} \qquad \xrightarrow{\epsilon \to 0} \qquad \max_i x_i$$

Let's generalize

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,\mathbf{y})\in\mathcal{D}} \max_{\hat{\mathbf{y}}} \left(\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)\right) - F(\mathbf{y},x,w)$$

even further

Structured Learning

Soft-max extension:

$$\epsilon \ln \sum_i \exp \frac{x_i}{\epsilon} \quad \xrightarrow{\epsilon \to 0} \quad \max_i x_i$$

Let's generalize

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,\mathbf{y})\in\mathcal{D}} \max_{\hat{\mathbf{y}}} \left(\Delta(\mathbf{y}, \hat{\mathbf{y}}) + F(\hat{\mathbf{y}}, x, w)\right) - F(\mathbf{y}, x, w)$$

even further

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y}, \hat{\mathbf{y}}) + F(\hat{\mathbf{y}}, x, w)}{\epsilon} - F(\mathbf{y}, x, w)$$

Structured Learning

A pretty general formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Structured Learning

A pretty general formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Generalizes:

- any form of SVM
- conditional random fields
- maximum-likelihood
- logistic regression
- convolutional neural networks

Structured Learning

A pretty general formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Generalizes:

- any form of SVM
- conditional random fields
- maximum-likelihood
- logistic regression
- convolutional neural networks

How does this correspond to a binary SVM?

Structured Learning

A pretty general formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Generalizes:

- any form of SVM
- conditional random fields
- maximum-likelihood
- logistic regression
- convolutional neural networks

How does this correspond to a binary SVM? Reverse slides.

Structured Learning

A pretty general formulation:

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Generalizes:

- any form of SVM
- conditional random fields
- maximum-likelihood
- logistic regression
- convolutional neural networks

How does this correspond to a binary SVM? Reverse slides.
How does this correspond to logistic regression?

Structured Learning

A pretty general formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Generalizes:

- any form of SVM
- conditional random fields
- maximum-likelihood
- logistic regression
- convolutional neural networks

How does this correspond to a binary SVM? Reverse slides.
How does this correspond to logistic regression? Up next...

Structured Learning

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{N} \sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Structured Learning

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Equivalent formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Structured Learning

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Equivalent formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Annealed loss-augmented probability:

$$p_\epsilon^\Delta(\mathbf{y} \mid x,w) = \left(\frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}\right)^\epsilon$$

Structured Learning

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}},x,w)}{\epsilon} - F(\mathbf{y},x,w)$$

Equivalent formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Annealed loss-augmented probability:

$$p_\epsilon^\Delta(\mathbf{y} \mid x, w) = \left(\frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}\right)^\epsilon$$

Equivalent maximum-likelihood formulation:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\ln \prod_{(x,\mathbf{y})\in\mathcal{D}} p_\epsilon^\Delta(\mathbf{y} \mid x, w)$$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Simplifications:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp\frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}}\exp\frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Simplifications:

- Binary classification: $y \in \{-1, 1\}$
- Simple feature-function: $F(\mathbf{y}, x, w) = yw^\top x$
- Epsilon: $\epsilon = 1$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Simplifications:

- Binary classification: $y \in \{-1, 1\}$
- Simple feature-function: $F(\mathbf{y}, x, w) = yw^\top x$
- Epsilon: $\epsilon = 1$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N} \ln \prod_{(x,y)} \frac{\exp yw^\top x}{\exp(w^\top x) + \exp(-w^\top x)}$$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Simplifications:

- Binary classification: $y \in \{-1, 1\}$
- Simple feature-function: $F(\mathbf{y}, x, w) = yw^\top x$
- Epsilon: $\epsilon = 1$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N} \ln \prod_{(x,y)} \frac{\exp yw^\top x}{\exp(w^\top x) + \exp(-w^\top x)}$$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N} \ln \prod_{(x,y)} \frac{1}{1 + \exp\left(-2yw^\top x\right)}$$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \frac{\exp \frac{\Delta(\mathbf{y},\mathbf{y})+F(\mathbf{y},x,w)}{\epsilon}}{\sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}})+F(\hat{\mathbf{y}},x,w)}{\epsilon}}$$

Simplifications:

- Binary classification: $y \in \{-1, 1\}$
- Simple feature-function: $F(\mathbf{y}, x, w) = yw^\top x$
- Epsilon: $\epsilon = 1$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\ln \prod_{(x,y)} \frac{\exp yw^\top x}{\exp(w^\top x) + \exp(-w^\top x)}$$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\ln \prod_{(x,y)} \frac{1}{1 + \exp\left(-2yw^\top x\right)}$$

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,y)} \ln \sigma\left(2yw^\top x\right) \qquad \text{with} \qquad \sigma(z) = \frac{1}{1+\exp(-z)}$$

We started with hinge loss:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)}^{n} \max\{0, 1 - yw^\top x\}$$

We started with hinge loss:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)}^{n} \max\{0, 1 - yw^\top x\}$$

Derived the general structured prediction framework:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}}, x, w)}{\epsilon} - F(\mathbf{y}, x, w)$$

We started with hinge loss:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)}^{n} \max\{0, 1 - yw^\top x\}$$

Derived the general structured prediction framework:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}}, x, w)}{\epsilon} - F(\mathbf{y}, x, w)$$

And showed how to simplify it to logistic regression:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,y)} \ln \sigma \left(2yw^\top x\right)$$

We started with hinge loss:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,y)}^{n} \max\{0, 1 - yw^\top x\}$$

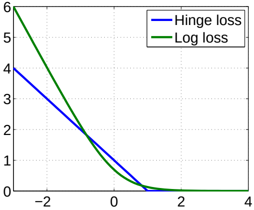
Derived the general structured prediction framework:

$$\min_w \frac{1}{2}\|w\|_2^2 + \frac{C}{N}\sum_{(x,\mathbf{y})\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{y}}} \exp \frac{\Delta(\mathbf{y},\hat{\mathbf{y}}) + F(\hat{\mathbf{y}}, x, w)}{\epsilon} - F(\mathbf{y}, x, w)$$

And showed how to simplify it to logistic regression:

$$\min_w \frac{1}{2}\|w\|_2^2 - \frac{C}{N}\sum_{(x,y)} \ln \sigma\left(2yw^\top x\right)$$

$$\min_w R(w) + \frac{C}{N}\sum_{(x,y)} \bar{\ell}(x, y, w)$$



$\max\{0, 1 - t\}$

$-\ln \sigma(2t)$

t

Structured Learning

How does this framework simplify to neural networks?

- Multiclass setting: $y \in \{1, \ldots, K\}$
- Composite feature function:

$$F(y, x, w) = f_1\Big(y, w_1, f_2\big(w_2, f_3(\ldots)\big)\Big)$$

- Epsilon: $\epsilon = 1$

Stay tuned for part III