Structured Prediction

CMSC 678
UMBC
Outline

Recap

Structured Prediction
  Example Applications
  Formalization and the problem of inference
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique
Recap from last time...
Two Problems for Sampling Methods to Solve

Generate samples from $p$

$$p(x) = \frac{u(x)}{Z}, x \in \mathbb{R}^D$$

$x_1, x_2, ..., x_R$ samples

Estimate expectation of a function $\phi$

$$\Phi = \langle \phi(x) \rangle_p = \mathbb{E}_{x \sim p}[\phi(x)] = \int p(x)\phi(x)dx$$

$$\hat{\Phi} = \frac{1}{R}\sum_r \phi(x_r)$$

If we could sample from $p$...

$$\mathbb{E}[\hat{\Phi}] = \Phi$$

consistent estimator

Q: Why is sampling from $p(x)$ hard?

A1: Can we evaluate $Z$?

A2: Can we sample without enumerating? (Correct samples should be where $p$ is big)
Three Sampling Methods

**Uniform Sampling**

Sample uniformly: 
\[ x_1, x_2, \ldots, x_R \]

\[ \Phi = \sum_r \phi(x_r) \frac{u(x)}{Z} \]

\[ Z^* = \sum_r u(x_r) \]

**Importance Sampling**

Sample from approximating \( Q(x) \propto u_q(x) \): 
\[ x_1, x_2, \ldots, x_R \]

\[ \Phi = \frac{\sum_r \phi(x_r)w(x_r)}{\sum_r w(x_r)} \]

\[ w(x_r) = \frac{u_p(x)}{u_q(x)} \]

**Rejection Sampling**

Sample \( x_1, x_2, \ldots, x_{R^*} \) from approximating distribution: 
\( Q(x) \propto u_q(x), c \ast u_q > u_p \)

Sample uniformly: 
\[ z_k \sim \text{Unif}(0, c \ast u_q(x_k)) \]

If \( z_k \leq u_p(x_k) \): add \( x_k \) to sampled \( R \) points

Otherwise: reject it

\[ \hat{\Phi} = \frac{1}{R} \sum_r \phi(x_r) \]
Goal:
\[ \Phi = \langle \phi(x) \rangle_p = \mathbb{E}_{x \sim p}[\phi(x)] \]

transition kernel/distribution:
\[ Q(x|x^{(t)}) \propto u_q(x|x^{(t)}) \]

sample from \( Q(x|x^{(t)}) \):
\[ x_1, x_2, \ldots, x_{R^*} \]

sample uniformly:
\[ \alpha_k = \frac{u_p(x_k) u_q(x_k)}{u_p(x^{(t)}) u_q(x^{(t)})} \]

if \( \alpha_k \geq 1 \): add \( x_k \) to sampled \( R \) points
otherwise: accept with probability \( \alpha_k \)

if accepted: \( x^{(t+1)} = x_k \)
otherwise: \( x^{(t+1)} = x^{(t)} \)

samples are not independent

Metropolis-Hastings can be used effectively in high-dimensional spaces 😊
Gibbs Sampling

transition kernel/distribution:
\[ Q(x|x^{(t)}) = p(x \mid \text{MB}(x^{(t)})) \]

sample (always accept) from
\[ Q(x|x^{(t)}): x_1, x_2, \ldots, x_{R^*} \]

Goal:
\[ \Phi = \langle \phi(x) \rangle_p = \mathbb{E}_{x \sim p}[\phi(x)] \]

\[ x^{(t+1)} = x_k \]

\[ \hat{\Phi} = \frac{1}{R} \sum_r \phi(x_r) \]

samples are not independent

Gibbs Sampling can be used effectively in high-dimensional spaces 😊
Collapsed Gibbs Sampler for LDirA

randomly assign $z_{*,*}$
maintain count tables:
  - $c(d,k)$: document-topic counts
  - $c(k,v)$: topic-word counts
for each document $d$:
  for each token $i$ in $d$:
    unassign topic: $z_{d,i}$
    resample $z_{d,i} \mid w_{d,i}, \{\psi_k\}, \{z_{*,-i}\}$
    reassign topic: $z_{d,i}$

\[
p(z_{di} \mid z_{*,-i}) = \alpha (c(d, k) - 1 + \alpha_k) \text{ *topic-word counts}
\]
Outline

Recap

Structured Prediction

Example Applications
Formalization and the problem of inference
Example: HMM Reformulation

Learning for Structured Prediction
Structured Perceptron
Structured SVM

Integer Linear Programming: A General Decoding Technique
Classification vs. Structured Prediction

“Flat” prediction
y is a single label

\[ p_\alpha (y \mid \text{structure}) \]

Structured prediction
y has some internal structure to predict

\[ p_\alpha (y = (y_1, y_2, \ldots, y_M) \mid \text{structure}) \]
Classification vs. Structured Prediction

“Flat” prediction

\( y \) is a single label

\[
p_\alpha \left( y \mid \right)
\]

ideally, this decomposes nicely

Structured prediction

\( y \) has some internal structure to predict

\[
p_\alpha \left( y = (y_1, y_2, \ldots, y_M) \mid \right)
\]
Example: Part-of-Speech Sequence Tagging

\[ p(British \ Left \ Waffles \ on \ Falkland \ Islands) \]

\[ p(z_1, w_1, z_2, w_2, ..., z_N, w_N) = p(z_1 | z_0)p(w_1 | z_1) \cdots p(z_N | z_{N-1})p(w_N | z_N) \]

\[ = \prod_i p(w_i | z_i) p(z_i | z_{i-1}) \]
Example: Other Sequence Models

Directed (e.g.,
hidden Markov model [HMM]; generative)

Directed (e.g.,
maximum entropy Markov model [MEMM]; conditional)

Undirected as factor graph
(e.g., conditional random field [CRF])
Example: Handwriting Recognition

Data: \[ D = \{ x^{(n)}, y^{(n)} \}_{n=1}^{N} \]

Sample 1:
\[ y^{(1)} \]
\[ x^{(1)} \]

Sample 2:
\[ y^{(2)} \]
\[ x^{(2)} \]

Sample 3:
\[ y^{(3)} \]
\[ x^{(3)} \]

Figures from (Chatzis & Demiris, 2013)

Slide courtesy Matt Gormley
Example: Machine Translation/Word Alignment

Le chat est sur la chaise.
The cat is on the chair.

\[ p(\text{English}|\text{French}) \propto p(\text{French}|\text{English}) \times p(\text{English}) \]

alignment/translation model
Example: Word Alignment, Phrase Extraction

Maria no daba una cla la bruja verde
Mary
did
not
slap
the
green
witch

... 5 farmers were thrown into jail in Ireland...
... fünf Landwirte festgenommen, weil ...
... oder wurden festgenommen, gefoltert ...
... or have been imprisoned, tortured ...

(Burkett & Klein, 2012)

http://www.cis.upenn.edu/~ccb/figures/research-statement/pivoting.jpg
Example: Object Recognition

Data consists of images $x$ and labels $y$. 

leopard $y$
Example: Object Recognition

Data consists of images $x$ and labels $y$.

Preprocess data into “patches”

leopard $y$
Example: Object Recognition

Data consists of images $x$ and labels $y$.

Preprocess data into “patches”

Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)

Slide courtesy Matt Gormley
Example: Object Recognition

Data consists of images $x$ and labels $y$.

Preprocess data into “patches”

Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)

Define graphical model with these latent variables in mind

$z$ is not observed at train or test time
Example: Probabilistic Parsing

\[ p(S) = p(NP) \times p(VP) \times p(Noun) \times p(Verb) \times p(NP) \times p(Noun) \times p(VP) \times p(Noun) \times p(NP) \times p(Verb) \times p(NP) \times p(A great city) \]
Example: Probabilistic Parsing

\[
p(S) = p(NP) \ast p(VP) \ast p(NP) \ast p(N) \ast p(\text{Baltimore}) \ast p(VP) \ast p(NP) \ast p(V) \ast p(\text{is}) \ast p(NP) \ast p(\text{a great city})
\]

Take CMSC 673 to learn more.
Outline

Recap

Structured Prediction
  Example Applications
  **Formalization and the problem of inference**
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique
Classification vs. Structured Prediction

“Flat” prediction
\( y \) is a single label

\[ p_\alpha ( y | ) \]

Structured prediction
\( y \) has some internal structure to predict

\[ p_\alpha ( y = (y_1, y_2, \ldots, y_M) | ) \]
Classification vs. Structured Prediction

“Flat” prediction

$y$ is a single label

$$\text{argmax} \quad p_\alpha (y | \cdot)$$

$y \in \{0,1,\ldots,L\}$

Structured prediction

$y$ has some internal structure to predict

$$\text{argmax} \quad p_\alpha (\mathbf{y} = (y_1, y_2, \ldots, y_M) | \cdot)$$

$\mathbf{y} \in \mathcal{Y}$

very large set
Formalizing Structured Prediction

Input $x \rightarrow$ Output $y$

Feature Extractor: $\Phi(x, y)$

Weights: $w$

Inference (decoding): $\arg\max_y w^T \Phi(x, y)$

Loss: $\ell(y_i, y)$, or $\Delta(y_i, y)$
Outline

Recap

Structured Prediction
  Example Applications
  Formalization and the problem of inference
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique
Reformulating a HMM

\[ p(y_1, x_1, y_2, x_2, \ldots, y_N, x_N) = \prod_i p(x_i | y_i) p(y_i | y_{i-1}) \]

How do we write this in the structured prediction formalism?

\[ \arg\max_y p(y | x) = \arg\max_y w^T \Phi(x, y) \]
Reformulating a HMM

\[ p(y_1, x_1, y_2, x_2, \ldots, y_N, x_N) = \prod_i p(x_i | y_i) p(y_i | y_{i-1}) \]

How do we write this in the structured prediction formalism?

\[ \arg\max_y p(y | x) = \arg\max_y p(y, x) = \arg\max_y w^T \Phi(x, y) \]

scoring must respect the original model
Reformulating a HMM

\[ p(y_1, x_1, y_2, x_2, ..., y_N, x_N) = \prod_{i} p(x_i | y_i) p(y_i | y_{i-1}) \]

How do we write this in the structured prediction formalism?

\[ \text{argmax}_y p(y | x) = \text{argmax}_y p(y, x) = \text{argmax}_y w^T \Phi(x, y) \]

Idea: \( w^T \Phi(x, y) = \log p(y, x) \)

scoring must respect the original model
Reformulating a HMM

\[ p(y_1, x_1, y_2, x_2, ..., y_N, x_N) = \prod_i p(x_i | y_i) p(y_i | y_{i-1}) \]

How do we write this in the structured prediction formalism?

\[ \arg \max_y p(y | x) = \arg \max_y p(y, x) = \arg \max_y w^T \Phi(x, y) \]

Idea: \( w^T \Phi(x, y) = \log p(y, x) \)

\[ w = (\log p(v_1 | k_1), ..., \log p(v_V | k_H), \log p(k_1 | k_1), ..., \log p(k_H | k_H)) \]

scoring must respect the original model

H*V weights

H*H weights
Reformulating a HMM

\[ p(y_1, x_1, y_2, x_2, ..., y_N, x_N) = \prod_i p(x_i | y_i) p(y_i | y_{i-1}) \]

How do we write this in the structured prediction formalism?

\[ \arg\max_y p(y | x) = \arg\max_y p(y, x) = \arg\max_y w^T \Phi(x, y) \]

Idea: \( w^T \Phi(x, y) = \log p(y, x) \)

\[ w = (\log p(v_1 | k_1), ..., \log p(v_V | k_H), \log p(k_1 | k_1), \log p(k_H | k_H)) \]

\[ \Phi = (c(k_1 \to v_1), ..., c(k_H \to v_V), c(k_1 \to k_1), ..., c(k_H \to k_H)) \]

scoring must respect the original model
Inference in HMMs

Viterbi: incrementally compute the best sequence

Greedy: Choose current latent state to maximize score

Sampling
Outline

Recap

Structured Prediction
  Example Applications
  Formalization and the problem of inference
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique
Remember: Learning a (Binary) Perceptron

Input: training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Perceptron training algorithm (Rosenblatt, 57)

Initialize $w \leftarrow [0, \ldots, 0]$

for iteration = 1,...,T

for example $i = 1,.., N$

predict according to the current model

$\hat{y}_i = \begin{cases} 
+1 & \text{if } w^T x_i > 0 \\
-1 & \text{if } w^T x_i \leq 0
\end{cases}$

if $y_i = \hat{y}_i$, no change
else, $w \leftarrow w + y_i x_i$
Remember: Learning a (Multiclass) Perceptron

Input: training data \( (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \)

Initialize for each class \( k \): \( w^{(k)} = (0, \ldots, 0) \)
for iteration = 1,\ldots,T
  for example \( i = 1,\ldots, N \)
    predict according to the current model
    \( z = (\ldots, \text{dot}(w^{(k)}, x_i), \ldots) \)
    \( \hat{y}_i = \arg\max_k z \)

if \( y_i \neq \hat{y}_i \)
  \( w^{(y_i)} += x_i \)
  \( w^{(\hat{y}_i)} -= x_i \)
Initialize outerproduct weights: \( w = (0, \ldots, 0) \in \mathbb{R}^{K \times \text{len}(x_i)} \)

for iteration = 1,\ldots,T

for example i = 1,\ldots, N

predict according to the current model

\[
z = (\ldots, \text{dot}(w^{(k)}, x_i), \ldots)
\]

\[
\hat{y}_i = \arg\max_k z
\]

if \( y_i \neq \hat{y}_i \)

\[
w = w + \text{join}(x_i, y_i) - \text{join}(x_i, \hat{y}_i)
\]

Remember: Learning a (Multiclass) Perceptron (Alternative Formulation)

Input: training data \((x_1 , y_1), (x_2 , y_2), \ldots , (x_n , y_n)\)
Remember: Learning a (Multiclass) Perceptron (Alternative Formulation)

Input: training data \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)

Initialize outerproduct weights: \(w = (0, \ldots, 0) \in \mathbb{R}^{K \times \text{len}(x_i)}\)
for iteration = 1,...,T
for example i = 1,..., N
    predict according to the current model
    \[ z = (\ldots, \text{dot}(w^{(k)}, x_i), \ldots) \]
    \[ \hat{y}_i = \arg\max_k z \]

if \(y_i \neq \hat{y}_i\)
    \[ w = w + \Phi(x_i, y_i) - \Phi(x_i, \hat{y}_i) \]

slight notation change
Learning a Structured Perceptron

Input: training data \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)

Initialize: \(w = (0, \ldots, 0)\)

for iteration = 1,\ldots,T
  for example i = 1,\ldots, N
    predict according to the current model:
    \[
    \hat{y}_i = \arg\max_{y \in \mathcal{Y}} w^T \Phi(x_i, y)
    \]
    if \(y_i \neq \hat{y}_i\)
      \[
      w = w + \Phi(x_i, y_i) - \Phi(x_i, \hat{y}_i)
      \]
Remember: Averaged perceptron

\begin{equation}
\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(K)} \text{ sequence of weights}
\end{equation}

\begin{equation}
c^{(1)}, c^{(2)}, \ldots, c^{(K)} \text{ "survival" times for each of these (# iterations since last update)}
\end{equation}

A weight that gets updated immediately gets \( c = 1 \)

A weight that survives another round gets \( c = 2 \), etc.

\begin{equation}
\hat{y} = \text{sign} \left( \sum_{k=1}^{K} c^{(k)} \text{sign} \left( \mathbf{w}^{(k)T} \mathbf{x} \right) \right)
\end{equation}

\begin{equation}
\text{voted}
\end{equation}

\begin{equation}
\hat{y} = \text{sign} \left( \sum_{k=1}^{K} c^{(k)} \left( \mathbf{w}^{(k)T} \mathbf{x} \right) \right) = \text{sign} \left( \bar{\mathbf{w}}^{T} \mathbf{x} \right)
\end{equation}

\begin{equation}
\text{averaged}
\end{equation}

Slide courtesy Hamed Pirsiavash
Learning an **Averaged Structured Perceptron**

**Input:** training data $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$

Initialize:

$w_a = (0, \ldots, 0)$

$w_c = (0, \ldots, 0)$

$c = 1$

for iteration = 1,...,T

for example $i = 1,\ldots, N$

predict according to the current model:

$$\hat{y}_i = \arg\max_{y \in y} w^T \Phi(x_i, y)$$

if $y_i \neq \hat{y}_i$

$$w_c = w_c + \Phi(x_i, y_i) - \Phi(x_i, \hat{y}_i)$$

$$w_a = w_a + c(\Phi(x_i, y_i) - \Phi(x_i, \hat{y}_i))$$

$$c = 0$$

$$c += 1$$
Outline

Recap

Structured Prediction
  Example Applications
  Formalization and the problem of inference
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique
Hamming Loss

The number of mis-predictions internally

The number of ("atomic") substructures wrong
Hamming Loss

The number of mis-predictions internally

The number of ("atomic") substructures wrong

$$\ell_{\text{Hamming}}(y, \hat{y}) = \sum_k 1[y_k \neq \hat{y}_k]$$
Remember: Soft-Margin Support Vector Machines

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{n} \xi_n$$

maximize margin  
minimize slack

subject to:  
$$y_n \mathbf{w}^T \mathbf{x}_n \geq 1 - \xi_n, \forall n$$ 
allow some slack

$$\xi_n \geq 0$$
Structured SVM: Formulate the Constraints

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n
\]

\[
w^T \Phi(x, y) - w^T \Phi(x, \hat{y}) \geq \ell^H(y, \hat{y}) - \xi_{n, \hat{y}}
\]

\[\hat{y} \in y\]
Structured SVM: Formulate the Constraints

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n
\]

\[
w^T \Phi(x, y) - w^T \Phi(x, \hat{y}) \geq \ell^H (y, \hat{y}) - \xi_{n, \hat{y}}
\]

\[\hat{y} \in y\]

Q: What’s wrong with this?
Structured SVM: Formulate the Constraints

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n
\]

\[
w^T \Phi(x, y) - w^T \Phi(x, \hat{y}) \geq \ell^H (y, \hat{y}) - \xi_{n, \hat{y}}
\]

\[\hat{y} \in y\]

**Q:** What’s wrong with this?

**A:** Too many slack variables (one per instance and per possible output)
Structured SVM: Formulate the Constraints

\[ \min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n \]

\[ w^T \Phi(x, y) - w^T \Phi(x, \hat{y}) \geq \ell^H(y, \hat{y}) - \xi_{n, \hat{y}} \]

attempt 1

idea: bound this

\[ \hat{y} \in y \]
Structured SVM: Formulate the Constraints

$$\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n$$

attempt 1

$$w^T \Phi(x, y) - w^T \Phi(x, \hat{y}) \geq \ell^H(y, \hat{y}) - \xi_{n, \hat{y}}$$

idea: bound this

$$\hat{y} \in y$$

attempt 2

$$w^T \Phi(x, y) \geq \max_{\hat{y}}[w^T \Phi(x, \hat{y}) + \ell^H(y, \hat{y})] - \xi_n$$
Structured SVM: Formulate the Constraints

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n
\]

attempt 1

\[
w^T \Phi(x, y) - w^T \Phi(x, \hat{y}) \geq \ell^H(y, \hat{y}) - \xi_{n, \hat{y}}
\]

idea: bound this

\[
\hat{y} \in y
\]

attempt 2

\[
w^T \Phi(x, y) \geq \max_{\hat{y}} [w^T \Phi(x, \hat{y}) + \ell^H(y, \hat{y})] - \xi_n
\]

“loss-augmented score”
Structured SVM: Formulate the Constraints

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n
\]

**attempt 1**

\[
w^T \Phi(x, \hat{y}) - w^T \Phi(x, \hat{y}) \geq \ell^H(y, \hat{y}) - \xi_{n, \hat{y}}
\]

**idea:** bound this

\[\hat{y} \in y\]

**attempt 2**

\[
w^T \Phi(x, y) \geq \max_{\hat{y}} [w^T \Phi(x, \hat{y}) + \ell^H(y, \hat{y})] - \xi_n
\]

“loss-augmented score”
Remember: SVMs as loss-based optimization

\[
\min_w \frac{1}{2} \|w\|^2 + C \sum \xi_n
\]

subject to: \( y_n w^T x_n \geq 1 - \xi_n, \forall n \)

\( \xi_n \geq 0 \)

\[
\min_w \frac{1}{2} \|w\|^2 + C \sum \max(0, 1 - y_n w^T x_n)
\]

same as hinge loss with squared norm regularization!
Structured SVM: Remove Slack

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n
\]

\[
w^T \Phi(x, y) \geq \max_{\hat{y}} [w^T \Phi(x, \hat{y}) + \ell^H (y, \hat{y})] - \xi_n
\]

"loss-augmented score"

at the optimum

\[
\xi_n = \max \left\{ 0, \max_w [w^T \Phi(x_n, \hat{y}_n) + \ell^H (y_n, \hat{y}_n)] - w^T \Phi(x_n, y_n) \right\}
\]
Structured SVM: Remove Slack

$$\min_w \frac{1}{2} |w|^2 + C \sum_n \xi_n$$

$$w^T \Phi(x, y) \geq \max_{\tilde{y}} [w^T \Phi(x, \tilde{y}) + \ell_H (y, \tilde{y})] - \xi_n$$

"loss-augmented score"

at the optimum

$$\xi_n = \max \left\{ 0, \max_w [w^T \Phi(x_n, \tilde{y}_n) + \ell_H (y_n, \tilde{y}_n)] - w^T \Phi(x_n, y_n) \right\}$$

= structured hinge loss
Structured SVM as Unconstrained Optimization

\[
\min_w \frac{1}{2} |w|^2 + C \sum_n \ell(\text{struct. hinge})(x_n, y_n)
\]

\[
\ell(\text{struct. hinge})(x_n, y_n) = \max \left\{ 0, \max_w [w^T \Phi(x_n, \hat{y}_n) + \ell^H(y_n, \hat{y}_n)] - w^T \Phi(x_n, y_n) \right\}
\]

Q: What’s the benefit of this?
Structured SVM as Unconstrained Optimization

$$\min_w \frac{1}{2} |w|^2 + C \sum_n \ell(\text{struct. hinge}) (x_n, y_n)$$

$$\ell(\text{struct. hinge}) (x_n, y_n) = \max \left\{ 0, \max_w [w^T \Phi (x_n, \hat{y}_n) + \ell^H (y_n, \hat{y}_n) ] - w^T \Phi (x_n, y_n) \right\}$$

Q: What’s the benefit of this?  
A: Derive subgradient and optimize with gradient-based solvers
Structured Hinge Loss (Sub)Gradient

\[ \nabla_w \ell(\text{struct. hinge})(x_n, y_n) = \nabla_w \max \left\{ 0, \max_w [w^T \Phi(x_n, \hat{y}_n) + \ell^H(y_n, \hat{y}_n)] - w^T \Phi(x_n, y_n) \right\} \]

\[ = \nabla_w [w^T \Phi(x_n, y^*_n) + \ell^H(y_n, y^*_n) - w^T \Phi(x_n, y_n)] \]

\[ = \Phi(x_n, y^*_n) - \Phi(x_n, y_n) \quad \text{(look familiar?)} \]
Outline

Recap

Structured Prediction
  Example Applications
  Formalization and the problem of inference
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique
Linear Programming

$$\max \ W \ x \ \text{subject to} \ \begin{cases} \ A W \leq B \\ a_1 w_1 + a_2 w_2 + \cdots a_M w_M \leq b \end{cases}$$

- input data
- matrix (or vector) of learned parameters
- linear constraints on $W$
Example: The diet problem

A student wants to spend as little money on food while getting sufficient amount of vitamin Z and nutrient X. Her options are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost/100g</th>
<th>Vitamin Z</th>
<th>Nutrient X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrots</td>
<td>2</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>Sunflower seeds</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Double cheeseburger</td>
<td>0.3</td>
<td>0.01</td>
<td>2</td>
</tr>
</tbody>
</table>

How should she spend her money to get at least 5 units of vitamin Z and 3 units of nutrient X?

Let c, s and d denote how much of each item is purchased.

Minimize total cost such that

- At least 5 units of vitamin Z,
- At least 3 units of nutrient X,
- The number of units purchased is not negative.
Example: The diet problem

A student wants to spend as little money on food while getting sufficient amount of vitamin Z and nutrient X. Her options are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost/100g</th>
<th>Vitamin Z</th>
<th>Nutrient X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrots</td>
<td>2</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>Sunflower seeds</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Double cheeseburger</td>
<td>0.3</td>
<td>0.01</td>
<td>2</td>
</tr>
</tbody>
</table>

How should she spend her money to get at least 5 units of vitamin Z and 3 units of nutrient X?

Let $c$, $s$ and $d$ denote how much of each item is purchased

$$\begin{align*}
\text{min} & \quad 2c + 6s + 0.3d \\
\text{such that} & \\
\text{At least 5 units of vitamin Z}, \\
\text{At least 3 units of nutrient X}, \\
\text{The number of units purchased is not negative}
\end{align*}$$

Minimize total cost
Example: The diet problem

A student wants to spend as little money on food while getting sufficient amount of vitamin Z and nutrient X. Her options are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost/100g</th>
<th>Vitamin Z</th>
<th>Nutrient X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrots</td>
<td>2</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>Sunflower seeds</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Double cheeseburger</td>
<td>0.3</td>
<td>0.01</td>
<td>2</td>
</tr>
</tbody>
</table>

How should she spend her money to get at least 5 units of vitamin Z and 3 units of nutrient X?

Let c, s and d denote how much of each item is purchased

\[
\begin{align*}
\text{min} & \quad 2c + 6s + 0.3d \\
\text{such that} & \quad 4c + 10s + 0.01d \geq 5 \\
& \quad \text{At least 5 units of vitamin Z,} \\
& \quad \text{At least 3 units of nutrient X,} \\
& \quad \text{The number of units purchased is not negative}
\end{align*}
\]

Minimize total cost
Example: The diet problem

A student wants to spend as little money on food while getting sufficient amount of vitamin Z and nutrient X. Her options are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost/100g</th>
<th>Vitamin Z</th>
<th>Nutrient X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrots</td>
<td>2</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>Sunflower seeds</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Double cheeseburger</td>
<td>0.3</td>
<td>0.01</td>
<td>2</td>
</tr>
</tbody>
</table>

How should she spend her money to get at least 5 units of vitamin Z and 3 units of nutrient X?

Let c, s and d denote how much of each item is purchased

\[
\text{min } 2c + 6s + 0.3d \\
\text{such that } 4c + 10s + 0.01d \geq 5 \\
0.4c + 4s + 2d \geq 3 \\
\text{Minimize total cost} \\
\text{At least 5 units of vitamin Z,} \\
\text{At least 3 units of nutrient X,} \\
\text{The number of units purchased is not negative} \]
Example: The diet problem

A student wants to spend as little money on food while getting sufficient amount of vitamin Z and nutrient X. Her options are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost/100g</th>
<th>Vitamin Z</th>
<th>Nutrient X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrots</td>
<td>2</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>Sunflower seeds</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Double cheeseburger</td>
<td>0.3</td>
<td>0.01</td>
<td>2</td>
</tr>
</tbody>
</table>

How should she spend her money to get at least 5 units of vitamin Z and 3 units of nutrient X?

Let $c$, $s$ and $d$ denote how much of each item is purchased

$$
\text{min \ } 2c + 6s + 0.3d \\
\text{such that} \\
4c + 10s + 0.01d \geq 5 \\
0.4c + 4s + 2d \geq 3 \\
c \geq 0, \ s \geq 0, \ d \geq 0.
$$

Minimize total cost

At least 5 units of vitamin Z,
At least 3 units of nutrient X,
The number of units purchased is not negative
The constraint matrix defines a polytope that contains allowed solutions (possibly not closed).

The objective defines cost for every point in the space.

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0.
\end{align*}
\]

Every constraint forbids a half-plane.

The points that are allowed form the feasible region.

Even though all points in the region are allowed, points on the faces maximize/minimize the cost.
Integer Linear Programming

\[
\max_{\mathbf{W}} \mathbf{W} \mathbf{x} \quad \text{subject to}
\]

1. linear constraints on \( \mathbf{W} \)
2. Each \( w_k \) is an integer
Integer Linear Programming

\[
\max_{\mathbf{W}} \mathbf{W} \mathbf{x} \quad \text{subject to} \\
\mathbf{A}\mathbf{W} \leq \mathbf{B} \\
a_1w_1 + a_2w_2 + \cdots a_Mw_M \leq b
\]

1. linear constraints on \( \mathbf{W} \)
2. Each \( w_k \) is an integer
Integer Linear Programming

\[
\text{max } \mathbf{W} \mathbf{x} \text{ subject to } \mathbf{A}\mathbf{W} \leq \mathbf{B} \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad a_1 w_1 + a_2 w_2 + \cdots + a_M w_M \leq b
\]

1. linear constraints on \( \mathbf{W} \)
2. Each \( w_k \) is an integer
Integer Linear Programming

\[
\max_W \mathcal{W} x \quad \text{subject to} \quad AW \leq B \\
\begin{align*}
    a_1 w_1 + a_2 w_2 + \cdots + a_M w_M &\leq b \\
1. \text{linear constraints on } W \\
2. \text{Each } w_k \text{ is an integer}
\end{align*}
\]
Integer Linear Programming

\[
\max_W \, W \cdot x \text{ subject to } \begin{align*}
AW & \leq B \\
\sum_{k=1}^{M} a_k w_k & \leq b
\end{align*}
\]

1. linear constraints on \( W \)
2. Each \( w_k \) is an integer

ILP is NP-complete 😞
Integer Linear Programming

\[
\begin{align*}
\max_{\mathbf{W}} & \quad \mathbf{W} \mathbf{x} \\
\text{subject to} & \quad \mathbf{A} \mathbf{W} \leq \mathbf{B} \\
& \quad a_1 w_1 + a_2 w_2 + \cdots a_M w_M \leq b
\end{align*}
\]

1. linear constraints on \( \mathbf{W} \)
2. Each \( w_k \) is an integer

ILP is NP-complete 😞

But there are still well-designed solvers \( \rightarrow \) it’s useful
Example: Sequence Tag Decoding as an ILP

$$\max_Z \sum x \text{ subject to}$$

1. linear constraints on $W$
2. Each $z_\star$ is an integer
Example: Sequence Tag Decoding as an ILP

\[\max_Z \ x \ \text{subject to}\]

1. linear constraints on \(W\)
2. Each \(z_*\) is an integer

(Encode the transitions)

\[z_{i,k',k} = 1[\text{label at step } i \text{ is } k \text{ and label at step } i-1 \text{ is } k']\]
Example: Sequence Tag Decoding as an ILP

\[
\max_Z \ x \ \text{subject to} \quad 1. \ \text{linear constraints on } \ W \\
\quad \quad \quad \quad 2. \text{Each } z_* \text{ is an integer}
\]

(Encode the transitions)

\[z_{i,k',k} = 1[\text{label at step } i \text{ is } k \text{ and label at step } i - 1 \text{ is } k']\]

(One state per time)
Example: Sequence Tag Decoding as an ILP

\[
\max_Z \ x \ \text{subject to} \quad 1. \text{linear constraints on } W \\
2. \text{Each } z_\ast \text{ is an integer}
\]

(Encode the transitions)

\[
z_{i,k',k} = 1 \ [\text{label at step } i \text{ is } k \text{ and label at step } i - 1 \text{ is } k']
\]

(One state per time)

\[
\sum_{k',k} z_{i,k',k} = 1 \quad \text{for each time } i
\]
Example: Sequence Tag Decoding as an ILP

\[
\max_Z \; x \quad \text{subject to} \quad \begin{align*}
1. & \text{ linear constraints on } W \\
2. & \text{ Each } z_* \text{ is an integer}
\end{align*}
\]

(Encode the transitions)

\[
z_{i,k',k} = 1 [\text{label at step } i \text{ is } k \text{ and label at step } i - 1 \text{ is } k']
\]

(One state per time)

\[
\sum_{k',k} z_{i,k',k} = 1 \quad \text{for each time } i
\]

(Internally consistent)

\[
\sum_{k'} z_{i,k',k} = \sum_v z_{i+1,k,v} \quad \text{for each time } i
\]

\[
\sum_k z_{i,k',k} = 1 \quad \text{for each state } k
\]
Outline

Recap

Structured Prediction
  Example Applications
  Formalization and the problem of inference
  Example: HMM Reformulation

Learning for Structured Prediction
  Structured Perceptron
  Structured SVM

Integer Linear Programming: A General Decoding Technique