ECE 6504: Advanced Topics in Machine Learning
Probabilistic Graphical Models and Large-Scale Learning

Topics
- Markov Random Fields
  - (Finish) MLE
  - Structured SVMs

Readings: KF 20.1-3, Barber 9.6

Dhruv Batra
Virginia Tech
Administrativia

• HW3
  – Extra credit

• Project Presentations
  – When: April 22, 24
  – Where: in class
  – 5 min talk
    • Main results
    • Semester completion 2 weeks out from that point so nearly finished results expected
    • Slides due: April 21 11:55pm
Recap of Last Time
Main Issues in PGMs

• **Representation**
  – How do we store $P(X_1, X_2, \ldots, X_n)$
  – What does my model mean/imply/assume? (Semantics)

• **Inference**
  – How do I answer questions/queries with my model? such as
  – Marginal Estimation: $P(X_5 | X_1, X_4)$
  – Most Probable Explanation: $\arg\max P(X_1, X_2, \ldots, X_n)$

• **Learning**
  – How do we learn parameters and structure of $P(X_1, X_2, \ldots, X_n)$ from data?
  – What model is the right for my data?
Recall -- Learning Bayes Nets

Data
\[ x^{(1)} \]
\[ \ldots \]
\[ x^{(m)} \]

True Distribution \( P^* \)
(Maybe corresponds to a BN \( G^* \) maybe not)

Domain Experts

\[ P(X_i \mid Pa_{X_i}) \]

CPTs --
## Learning Bayes Nets

<table>
<thead>
<tr>
<th></th>
<th>Known structure</th>
<th>Unknown structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully observable data</td>
<td>Very easy</td>
<td>Hard</td>
</tr>
<tr>
<td>Missing data</td>
<td>Somewhat easy (EM)</td>
<td>Very very hard</td>
</tr>
</tbody>
</table>

### Data

- $\mathbf{x}^{(1)}$
- $\ldots$
- $\mathbf{x}^{(m)}$

### Structure

### CPTs

$$P(X_i | \text{Pa}_{X_i})$$

(C) Dhruv Batra

Slide Credit: Carlos Guestrin
Learning the CPTs

For each discrete variable $X_i$

\[
\hat{P}_{MLE}(X_i = a \mid Pa_{X_i} = b) = \frac{\text{Count}(X_i = a, Pa_{X_i} = b)}{\text{Count}(Pa_{X_i} = b)}
\]
## Learning Markov Nets

<table>
<thead>
<tr>
<th></th>
<th>Known structure</th>
<th>Unknown structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully observable data</td>
<td>NP-Hard (but doable)</td>
<td>Harder</td>
</tr>
<tr>
<td>Missing data</td>
<td>Harder (EM)</td>
<td>Don’t try this at home</td>
</tr>
</tbody>
</table>

**Equation:**

\[ x^{(1)} \ldots x^{(m)} \]

**Diagram:**

- **Data:** \( x^{(1)} \ldots x^{(m)} \)
- **Structure:**
- **Factors:** \( \Psi_c(x_c) \)
- **Parameters:**

(C) Dhruv Batra

Slide Credit: Carlos Guestrin
Learning Parameters of a BN

- Log likelihood decomposes:

\[
\log P(D \mid \theta) = m \sum_i \sum_{x_i, \text{Pa}_{x_i}} \hat{P}(x_i, \text{Pa}_{x_i}) \log P(x_i \mid \text{Pa}_{x_i})
\]

- Learn each CPT independently

- Use counts

\[
\hat{P}(u) = \frac{\text{Count}(U = u)}{m}
\]
Log Likelihood for MN

• Log likelihood decomposes:

\[
\log P(D | \theta, \mathcal{G}) = m \sum_i \sum_{c_i} \hat{P}(c_i) \log \psi_i(c_i) - m \log Z
\]

• Doesn’t decompose!
  – \( \log Z \) couples all parameters together
Log-linear Markov network (most common representation)

• **Feature (or Sufficient Statistic)** is some function $\phi[D]$ for some subset of variables $D$
  - e.g., indicator function

• **Log-linear model** over a Markov network $H$:
  - a set of features $\phi_1[D_1], \ldots, \phi_k[D_k]$
    • each $D_i$ is a subset of a clique in $H$
    • two $\phi$’s can be over the same variables
  - a set of weights $w_1, \ldots, w_k$
    • usually learned from data

\[
P(X_1, \ldots, X_n) = \frac{1}{Z} \exp \left[ \sum_{i=1}^{k} w_i \phi_i(D_i) \right]
\]
Learning params for log linear models – Gradient Ascent

\[ P(X_1, \ldots, X_n) = \frac{1}{Z} \exp \left[ \sum_{i=1}^{k} w_i \phi_i(D_i) \right] \]

- Log-likelihood of data:

\[ \log P(\mathcal{D} | w, \mathcal{G}) = \sum_{j=1}^{m} \log \frac{1}{Z} \exp \left[ \sum_{i=1}^{k} w_i \phi_i(d_i^{(j)}) \right] \]

- Compute derivative & optimize
  - usually with gradient ascent or L-BFGS

\[
\frac{\partial \ell(\mathcal{D} : w)}{\partial w_i} = m \sum_{d_i} \hat{P}(d_i) \phi_i(d_i) - m \frac{\partial \log Z}{\partial w_i}
\]
Learning log-linear models with gradient ascent

• Gradient:

\[ \frac{\partial \ell(D : w)}{\partial w_i} = m \sum_{d_i} \hat{P}(d_i) \phi_i(d_i) - m \sum_{d_i} P(d_i \mid w) \phi_i(d_i) \]

• Requires one inference computation per

• Theorem: \( w \) is maximum likelihood solution iff

• Usually, must regularize
  – E.g., \( L_2 \) regularization on parameters
Plan for today

• MRF Parameter Learning
  – MLE
    • Conditional Random Fields
    • Feature example
  – Max-Margin
    • Structured SVMs
    • Cutting-Plane Algorithm
    • (Stochastic) Subgradient Descent
Semantic Segmentation

• Setup
  – 20 categories + background
    • Dataset: Pascal Segmentation Challenge (VOC 2012)
    • 1500 train/val/test images
Conditional Random Fields
Conditional Random Fields

- Log-Potentials / Scores

\[ S(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{(i,j) \in E} \theta_{ij}(y_i, y_j) \]

\[ P(y) = \frac{1}{Z} e^{S(y)} \]

- Express as a Log-Linear Model
  - On board

\[ \theta_i(y_i) = w_i \cdot \phi(x, y_i) \]

\[ \theta_{ij}(y_i, y_j) = w_{ij} \cdot \phi(x, y_i, y_j) \]
MLE for CRFs

• Model

\[ P(y|x) = \frac{1}{\mathcal{Z}_x} e^{S(y;x)} \]
\[ = \frac{1}{\mathcal{Z}_x} e^{w^T \phi(x,y)} \]

• Log-Likelihood:
  – On board

• Derivative:
  – On board
New Topic: Structured SVMs
Recall: Generative vs. Discriminative

• Generative Approach *(Naïve Bayes)*
  – Estimate $p(X|Y)$ and $p(Y)$
  – Use Bayes Rule to predict $y$

• Discriminative Approach
  – Estimate $p(Y|X)$ directly *(Logistic Regression)*
  – Learn “discriminant” function $h(x)$ *(Support Vector Machine)*
Recall: Generative vs. Discriminative

• Generative Approach (Markov Random Fields)
  – Estimate $p(X,Y)$
  – At test time, use $P(X=x,Y)$ to predict $y$

• Discriminative Approach
  – Estimate $p(Y|X)$ directly (Conditional Random Fields)
  – Learn “discriminant” function $h(x)$ (Structured SVMs)

\[
h(x) = \arg\max_{y \in \mathcal{Y}} w^T \phi(x, y)
\]
Structured SVM

- Joint features $\phi(x, y)$ describe match between $x$ and $y$
- Learn weights $w$ so that $w^T \phi(x, y)$ is max for correct $y$

$w^T \phi(x^1, y)$

$w^T \phi(x^j, y)$

$w^T \phi(x^m, y)$

$(x^j, y^j)$
Structured SVM

• Hard Margin
  – On board
Soft-Margin Structured SVM

- Two ideas
  - Add slack

\[ w^T \phi(x^1, y) \]
\[ w^T \phi(x^j, y) \]
\[ w^T \phi(x^m, y) \]

\( (x^j, y^j) \)
Soft-Margin Structured SVM

- Two ideas
  - Add slack
  - Re-scale the margin with a loss function

Lemma: The training loss is upper bounded by

$$Err(h) = \frac{1}{m} \sum_{j=1}^{m} \Delta(y^j, h(x^j)) \leq \frac{1}{m} \sum_{j=1}^{m} \xi_j$$
Soft-Margin Structured SVM

- Minimize \( \frac{1}{2} w^2 + \frac{C}{N} \sum_j \xi_j \)

subject to \( w^T \phi(x^j, y^j) \geq w^T \phi(x^j, y) + \Delta(y^j, y) - \xi_j \)

Too many constraints!
Cutting-Plane Method

\[ \frac{1}{2} w^2 + \frac{C}{N} \sum_j \xi_j \]

\[ w^T \phi(x^j, y^j) \geq w^T \phi(x^i, y) + \Delta(y^i, y) - \xi_j \]

• Cutting Plane
  – Suppose we only solve the SVM objective over a small subset of constraints (working set).
  – Some constraints from global set might be violated.
Cutting-Plane Method

![3D graph showing the objective function with axes labeled as $\xi$ and $W$]
Cutting-Plane Method

Original SVM Problem
• Exponential constraints
• Most are dominated by a small set of “important” constraints

Structural SVM Approach
• Repeatedly finds the next most violated constraint…
• …until set of constraints is a good approximation.

(C) Dhruv Batra
Slide Credit: Yisong Yue
Cutting-Plane Method

Original SVM Problem
- Exponential constraints
- Most are dominated by a small set of “important” constraints

Structural SVM Approach
- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.
Cutting-Plane Method

Original SVM Problem
- Exponential constraints
- Most are dominated by a small set of “important” constraints

Structural SVM Approach
- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.
Cutting-Plane Method

Original SVM Problem
• Exponential constraints
• Most are dominated by a small set of “important” constraints

Structural SVM Approach
• Repeatedly finds the next most violated constraint…
• …until set of constraints is a good approximation.

*This is known as a “cutting plane” method.
Cutting-Plane Method

\[ \frac{1}{2} w^2 + \frac{C}{N} \sum_j \xi_j \]

\[ w^T \phi(x^j, y^j) \geq w^T \phi(x^j, y) + \Delta(y^j, y) - \xi_j \]

- Cutting Plane
  - Suppose we only solve the SVM objective over a small subset of constraints (working set).
  - Some constraints from global set might be violated.
  - Degree of violation?

\[ w^T \phi(x^j, y) + \Delta(y^j, y) - \xi_j - w^T \phi(x^j, y^j) \]
Finding Most Violated Constraint

• Finding most violated constraint is equivalent to maximizing the RHS w/o slack:

\[
\text{Violation} = w^T \phi(x, y) + \Delta(y^j, y)
\]

• Requires solving:

\[
\arg\max_y w^T \phi(x, y) + \Delta(y^j, y)
\]

• Highly related to inference:

\[
h(x; w) = \arg\max_{y \in Y} [w^T \phi(x, y)]
\]
**Side note:** What’s the difference between SVMs and logistic regression?

**SVM:**

minimize_{w,b} \[ w \cdot w + C \sum_j \xi_j \]

\[
(w \cdot x_j + b) y_j \geq 1 - \xi_j, \forall j
\]

\[
\xi_j \geq 0, \forall j
\]

**Logistic regression:**

\[
P(Y = 1 | x, w) = \frac{1}{1 + e^{-(w \cdot x + b)}}
\]

Log loss:

\[
- \ln P(Y = 1 | x, w) = \ln \left(1 + e^{-(w \cdot x + b)}\right)
\]