Probabilistic Graphical Models: MRFs and CRFs

CSE628: Natural Language Processing
Guest Lecturer: Veselin Stoyanov
Why PGMs?

- PGMs can model joint probabilities of many events.
- “many ... techniques commonly used as part of a speech recognition system can be described by a graph –this includes Gaussian distributions, mixture models, decision trees, factor analysis, principle component analysis, linear discriminant analysis, and hidden Markov models... [including] many advanced models at the acoustic-, pronunciation-, and language-modeling levels.”

Bilmes (2004)
ACL 2010 Papers

1. Conditional Random Fields for Word Hyphenation
2. Practical Very Large Scale CRFs
3. Jointly Optimizing a Two-Step Conditional Random Field Model for Machine Transliteration and Its Fast Decoding Algorithm
5. Topic Models for Word Sense Disambiguation and Token-Based Idiom Detection
6. PCFGs, Topic Models, Adaptor Grammars and Learning Topical Collocations and the Structure of Proper Names
7. A Latent Dirichlet Allocation Method for Selectional Preferences
8. Latent Variable Models of Selectional Preference
9. A Bayesian Method for Robust Estimation of Distributional Similarities
10. Bayesian Synchronous Tree-Substitution Grammar Induction and Its Application to Sentence Compression
11. Blocked Inference in Bayesian Tree Substitution Grammars
12. Decision Detection Using Hierarchical Graphical Models
Today’s Lecture

• A **very** brief overview of Probabilistic Graphical Models (PGMs)
  – Undirected models:
    • MRFs and CRFs
• Learning and decoding in MRFs and CRFs
• Case study: Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables

For concise introduction to PGMs and their uses in NLP:
https://sites.google.com/site/spring2011pgm/
Markov Random Fields

• Describe joint probability, $P(A,B,C,D)$ using an *undirected* graph
  – Edges describe interacting RVs
  – Include a term (*factor*, $\phi$) in $P(A,B,C,D)$ for all sets of RVs that interact
Markov Random Fields

- Describe joint probability, $P(A,B,C,D)$ using an undirected graph
  - Edges describe interacting RVs
  - Include a term (factor, $\phi$) in $P(A,B,C,D)$ for all sets of RVs that interact

Random variable: a variable whose value results from a measurement on some type of random process. I.e., $A = \{\text{Alice’s grade for homework 2}\}$
Markov Random Fields

- Describe joint probability, \( P(A, B, C, D) \) using an undirected graph
  - Edges describe interacting RVs
  - Include a term (factor, \( \phi \)) in \( P(A, B, C, D) \) for all sets of RVs that interact

Conditional dependencies. For instance, Alice and Bob worked on the homework together, while Alice and Charlie did not.
Markov Random Fields

- Describe joint probability, $P(A,B,C,D)$ using an undirected graph
  - Edges describe interacting RVs
  - Include a term (factor, $\phi$) in $P(A,B,C,D)$ for all sets of RVs that interact

For each configuration of RVs $X_1, X_2,..., X_n$, $\phi(X_1, X_2,..., X_n)$ gives an affinity score.
Markov Random Fields

- Describe joint probability, $P(A, B, C, D)$ using an *undirected* graph
  - Edges describe interacting RVs
  - Include a term *(factor, $\phi$)* in $P(A, B, C, D)$ for all sets of RVs that interact
Joint Probability via Local Factors

\[ P(a,b,c,d) = \frac{1}{Z} \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a) \]

\[ Z = \sum_{abcd} \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a) \]

- Factors map subsets of RVs to non-negative real numbers
- High values for \( P(a,b,c,d) \) mean the local factors have high scores
- Inference: we can use \( P(A,B,C,D) \) to answer queries (conditional probability, marginal probability, most probable (MAP) assignment)
Joint Probability via Local Factors

\[ P(a, b, c, d) = \frac{1}{Z} \phi_1(a, b) \phi_2(b, c) \phi_3(c, d) \phi_4(d, a) \]

\[ Z = \sum_{abcd} \phi_1(a, b) \phi_2(b, c) \phi_3(c, d) \phi_4(d, a) \]

- Factors map subsets of RVs to non-negative real numbers
- High values for \( P(a, b, c, d) \) mean the local factors have high scores
- Inference: we can use \( P(A, B, C, D) \) to answer queries (conditional probability, marginal probability, most probable (MAP) assignment)
Joint Probability via Local Factors

$P(a,b,c,d) = \frac{1}{Z} \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a)$

$Z = \sum_{abcd} \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a)$

- Factors map subsets of RVs to non-negative real numbers
- High values for $P(a,b,c,d)$ mean the local factors have high scores
- Inference: we can use $P(A,B,C,D)$ to answer queries (conditional probability, marginal probability, most probable (MAP) assignment)
Joint Probability via Local Factors

**Factors don’t have to be tables... could be functions (that could be used to produce tables). Often depend on features.**

<table>
<thead>
<tr>
<th></th>
<th>a(^0)</th>
<th>b(^0)</th>
<th>c(^0)</th>
<th>d(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_1(A,B))</td>
<td>a(^0)</td>
<td>b(^0)</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a(^0)</td>
<td>b(^1)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a(^1)</td>
<td>b(^0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a(^1)</td>
<td>b(^1)</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>b(^0)</th>
<th>c(^0)</th>
<th>d(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_2(B,C))</td>
<td>b(^0)</td>
<td>c(^0)</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>b(^0)</td>
<td>c(^1)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>b(^1)</td>
<td>c(^0)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>b(^1)</td>
<td>c(^1)</td>
<td>100</td>
</tr>
</tbody>
</table>

\[
P(a,b,c,d) = \frac{1}{Z} \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a)
\]

\[
Z = \sum_{abcd} \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a)
\]

**Factors map subsets of RVs to non-negative real numbers.**

**High values for** \(P(a,b,c,d)\) **mean the local factors have high scores.**

**Inference:** we can use \(P(A,B,C,D)\) to answer queries (conditional probability, marginal probability, most probable (MAP) assignment)

**Z is just a constant, so is often left out of the discussion. We don’t need don’t even need to calculate it if we just want to “decode” : find most likely Y|X (conditional probability, marginal probability, most probable (MAP) assignment).**
Factorization implies Independencies

\[ P(a,b,c,d) \propto \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a) \]
\[ \propto [\phi_1(a,b) \phi_2(b,c)] [\phi_3(c,d) \phi_4(d,a)] \]
\[ \propto F(b, a,c) G(d, a,c) \]

- Intuition:
  If \((a,c)\) given, knowing \(b\) tells us nothing of \(d\)

- Theorem:
  \(X \perp Y \mid W\) iff.
  \[P(X_1...X_N) = F(X, W) G(Y, W)\]
Factorization implies Independencies

\[
P(a,b,c,d) \propto \phi_1(a,b) \phi_2(b,c) \phi_3(c,d) \phi_4(d,a) \\
\propto [\phi_1(a,b) \phi_2(b,c)] [\phi_3(c,d) \phi_4(d,a)] \\
\propto F(b, a,c) G(d, a,c)
\]

This is how we know the independencies from the factors – can we tell just by looking at the graph?
Formal MN Independencies

- $X \perp Y \mid W$ (X and Y are indep. given W)
  if we can NOT go from $X \in X$ to $Y \in Y$ without passing $W$
- $W$ blocks the interaction of X and Y

Unshaded: unobserved
(X, Y)

Shaded: observed
(W)
Formal MN Independencies

- Consequence: X is independent of everyone else given its neighbors (its Markov blanket)
- To make a MN: connect X and Y if they should never be rendered independent via other rvs

Unshaded: unobserved (X, Y)

Shaded: observed (W)

Simpler than global independencies in BNs (d-separation, see Chapter 3.3 in KF09)

You show me a graph, I can tell you what is independent from what given what.

Consequence: X is independent of everyone else given its neighbors (its Markov blanket).

To make a MN: connect X and Y if they should never be rendered independent via other rvs.

Unshaded: unobserved (X, Y)

Shaded: observed (W)

Simpler than global independencies in BNs (d-separation, see Chapter 3.3 in KF09)

You show me a graph, I can tell you what is independent from what given what.

Consequence: X is independent of everyone else given its neighbors (its Markov blanket).

To make a MN: connect X and Y if they should never be rendered independent via other rvs.

Unshaded: unobserved (X, Y)

Shaded: observed (W)

Simpler than global independencies in BNs (d-separation, see Chapter 3.3 in KF09)

You show me a graph, I can tell you what is independent from what given what.

Consequence: X is independent of everyone else given its neighbors (its Markov blanket).

To make a MN: connect X and Y if they should never be rendered independent via other rvs.

Unshaded: unobserved (X, Y)

Shaded: observed (W)

Simpler than global independencies in BNs (d-separation, see Chapter 3.3 in KF09)

You show me a graph, I can tell you what is independent from what given what.

Consequence: X is independent of everyone else given its neighbors (its Markov blanket).

To make a MN: connect X and Y if they should never be rendered independent via other rvs.

Unshaded: unobserved (X, Y)

Shaded: observed (W)
More on Factors: Factor Product

Figure 4.3, KF09

\[ \phi_1(A,B) \times \phi_2(B,C) = \psi(A,B,C) \]

- every entry in \( \psi(a,b,c) \) is \( \phi_1(a,b) \times \phi_2(b,c) \)
Log-Linear Models

• Factor graphs are more explicit, but still require bulky tables for all the factor values

• Can use *features* to capture patterns that we’d like reflected in clique potentials:

\[ P'_{\Phi}(X_1, \ldots, X_N) = \phi_1(D_1) \phi_2(D_2) \ldots \phi_K(D_K) \]

Define \( \phi_i(D_i) = \exp(-w_i f_i(D_i)) \)

- \( f_i(D_i) \) tell us something indicative about some rvs,
- e.g. \( f_i(Y_1 Y_2) = 1 \) if the vals of \( Y_1 \) & \( Y_2 \) are DT & NN, else \( f_i(Y_1 Y_2) = 0 \)

So \( P'_{\Phi}(X_1, \ldots, X_N) = \exp(-w_1 f_1(D_1)) \ldots \exp(-w_1 f_1(D_1)) \)

\[ = \exp(-\sum w_i f_i(D_i)) \]
Log-Linear Models

• Factor graphs are more explicit, but still require bulky tables for all the factor values
• Can use *features* to capture patterns that we’d like reflected in clique potentials:

\[ P_{\phi}'(X_1, \ldots, X_N) = \phi_1(D_1) \phi_2(D_2) \]

Define \( \phi_i(D_i) = \exp(-w_i f_i(D_i)) \)

- \( f_i(D_i) \) tell us something indicative about some \( \text{rvs} \)
- e.g. \( f_i(Y_1Y_2) = 1 \) if the vals of \( Y_1 \) & \( Y_2 \) are DT & NN, else \( f_i(Y_1Y_2) = 0 \)

So \( P_{\phi}'(X_1, \ldots, X_N) = \exp(-w_1 f_1(D_1)) \cdot \cdots \cdot \exp(-w_K f_K(D_K)) = \exp(-\sum w_i f_i(D_i)) \)

We can have many feature functions for each \( D_i \) that each indicate different properties of our \( \text{rvs} \).
Features in Markov Random Fields

- Conditional tables
- vs. Features:
  - $\phi_1(XY) = 1(X=Y)$
  - $\phi_2(XY) = 1$ if $X$ and $Y$ one grade apart, 0 otherwise
Conditional Random Fields

• CRFs:
  – An MRF over $X$ (input) and $Y$ (output)
  – Model conditional probability $P(Y|X)$
  – Uses features
Conditional Random Fields
Conditional Random Fields

- Does not need to model input interactions
- Less parameters \( \rightarrow \) easier to learn
Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data

John Lafferty†*
Andrew McCallum*†
Fernando Pereira*‡

*WhizBang! Labs–Research, 4616 Henry Street, Pittsburgh, PA 15213 USA
†School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA
‡Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA

Figure 2. Graphical structures of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences. An open circle indicates that the variable is not generated by the model.

3906 citations on Google Scholar
We present conditional random fields, a framework for building probabilistic models to segment and label sequence data. Conditional random fields offer several advantages over hidden Markov models and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models.

Figure 2. Graphical structures of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences. An open circle indicates that the variable is not generated by the model.
Conditional Random Fields for Sequences

- A Markov Random Field
  - With features
  - Trained and used for conditional probabilities

- Useful for modeling sequences
Learning MRF parameters

• Easy when exact inference is possible
• Given some data $D=\{X_i\}_n$, we want to learn MRF parameters
• Maximum likelihood principle
  – Pick the parameters that maximize the (log) probability of the data:
    • $\arg\max_\Theta L(D,\Theta)$
MLE for MRF parameters

• Pick the parameters that maximize the (log) probability of the data:

\[ p(D \mid \theta) = \prod_i p_\theta(X_i) \]

\[ L(D, \Theta) = \log p(D \mid \Theta) = \log(\prod_i p_\theta(X_i)) = \sum_i \log( p_\theta(X_i)) \]

\[ p_\theta(X_i) = \frac{1}{Z} \exp \left\{ \sum_A \sum_{f_j \in A} \Theta_j f_j(x_A) \right\} \]

\[ \log( p(X_i)) = \sum_A \sum_j \Theta_j f_j(x_A) - \log(Z) \]

\[ \log(Z) = \log \left( \sum_X \exp \left\{ \sum_A \sum_{f_j \in A} \Theta_j f_j(x_A) \right\} \right) \]
MLE for MRF parameters

• Compute the gradient of the log-likelihood
• Perform gradient-based optimization method (e.g. stochastic gradient decent) to find the parameters that maximize the log-likelihood
Learning CRF parameters

• Use conditional Log-likelihood instead

• Given some data $D=\{X_i, Y_i\}_n$, we want to learn CRF parameters

• Conditional log-likelihood:

$$\log(p(Y_i | X_i, \theta)) = \sum_A \sum_j \Theta_j f_j(\{x, y\}_A) - \log(Z_{X_i})$$

• $Z$ is now input-dependent:

$$\log(Z_{x_i}) = \log\left(\sum_Y \exp\left(\sum_A \sum_{f_j \in A} \Theta_j f_j(\{x, y\}_A)\right)\right)$$
Learning CRF parameters

• Use conditional Log-likelihood instead
• Given some data $D=\{X_i, Y_i\}_n$, we want to learn CRF parameters
• Conditional log-likelihood:

$$\log( p(Y_i | X_i, \theta)) = \sum \sum \Theta_j f_j(\{x, y\}_A) - \log(Z_{X_i})$$

Compare to MRF:

$$\log( p(X_i)) = \sum \sum A_j f_j(x_A) - \log(Z)$$
Learning CRF parameters

- Use conditional Log-likelihood instead
- Given some data $D=\{X_i, Y_i\}_n$, we want to learn CRF parameters

Compare to MRF:

$$\log(Z) = \log \left( \sum_X \exp \left( \sum_A \sum_{f_j \in A} \Theta_{f_j} f_j(x_A) \right) \right)$$

$$\log(Z_{x_i}) = \log \left( \sum_Y \exp \left( \sum_A \sum_{f_j \in A} \Theta_{f_j} f_j(\{x, y\}_A) \right) \right)$$
Training a CRF

• Use a gradient-based method to optimize the conditional log-likelihood of the data
Training a CRF

• Use a gradient-based method to optimize the conditional log-likelihood of the data
  – The derivative of a single parameter \( \theta_k \) and training example \( \{x_i, y_i\} \)

\[
\frac{\partial L(x_i, y_i)}{\partial \theta_k} = \sum_A f_k(\{x_i, y_i\}_A) - \sum_A f_k(\{x_i, y_i\}_A) p_{\Theta}(\{x_i, y_i\}_A | x_i)
\]

  – i.e., feature counts – expected feature counts
  – Easy to compute when exact inference can be performed
    • Need to count features in the data and compute marginals
CRFs for Classification

• CRFs are models for predicting probabilities:
  – \( p(Y|X) \) or \( p(y_k|X) \)

• In ML we care about decisions (predictions)
  – Is this spam?
  – What is the POS tag of this word?

• How do we turn probabilities into decisions?
  – For a sequence model:
    • Sequence of states with max probability
      – Max Product Algorithm (Viterbi decoding)
CRFs for Classification

• CRFs are models for predicting probabilities:
  – \( p(Y|X) \) or \( p(y_k|X) \)

• In ML we care about decisions (predictions)
  – Is this spam?
  – What is the POS tag of this word?

• How do we turn probabilities into decisions?
  – For a sequence model:
    • Sequence of states with max probability
      – Max Product Algorithm (Viterbi decoding)
Decision Theory

• Concerned with converting probabilities into decisions
• Use the generalized Bayes rule, or minimum Bayes risk (MBR) rule
  – Select the output $y'$ that minimizes the risk or the expected loss:

\[
\text{risk}(y') = E_{p(y|x)}[l(y', y)] = \sum_{y''} l(y', y'')
\]

• MBR decoding depends on the loss function
  – Examples:
    • Accuracy \(\rightarrow\) output the label with the max marginal probability
    – Can be intractable
      • F-measure
Case Study: Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables

[Nakagawa, Inui, Kurohashi, HLT-NAACL 2010]
Sentiment Classification

- Classify documents / sentences / phrases / words as subjective/objective or positive/negative:
  - “The picture quality is unparalleled” → Positive
  - “Watching this movie is like watching paint dry” → Negative
  - “The design is really kewl but this thing is expensive” → ?
Sentiment classification

• Bag-of-words approaches:
  “good” “kewl” “cool” “small” $\rightarrow$ positive
  “bad” “dull” “outrageous” “big” $\rightarrow$ negative

• Cannot handle compositionality:
  great
  killed
  terrorist
Sentiment classification

• Bag-of-words approaches:
  “good” “kewl” “cool” “small” $\rightarrow$ positive
  “bad” “dull” “outrageous” “big” $\rightarrow$ negative

• Cannot handle compositionality:
  great at taking your money
  killed terrorist
Sentiment classification

• Bag-of-words approaches:
  “good” “kewl” “cool” “small” $\rightarrow$ positive
  “bad” “dull” “outrageous” “big” $\rightarrow$ negative

• Cannot handle compositionality:
  not great at taking your money
  killed terrorist targets
Sentiment Composionality

• Annotations only at the sentence (or phrase level)
  – Need to model hidden structure
• Model needs to be compositional
• This paper:
  – Utilize CRF to model dependencies between words
  – Model hidden structure
  – Use dependency parse to indicate sentential structure
Dependency Subtrees and Compositionality

Whole Dependency Tree

Polarities of Dependency Subtrees
CRF Based on the Dependency Tree

It prevents cancer and heart disease.
CRF Based on the Dependency Tree

<root> It prevents cancer and heart disease.
CRF Based on the Dependency Tree

<root> It prevents cancer and heart disease.

Diagram with nodes and edges representing the dependency tree.
Model Specifics

• Inference:
  – Belief propagation

• Decision:
  – Argmax over the root marginal

• Learning:
  – MAP estimation of the parameters
Parameter Estimation

\[ \mathcal{L}_\Lambda = \sum_{l=1}^{L} \log P_\Lambda (p^l | w^l, h^l) - \frac{1}{2\sigma^2} \sum_{k=1}^{K} \lambda_k^2, \]

\[ \hat{\Lambda} = \arg\max_{\Lambda} \mathcal{L}_\Lambda, \]

Gradient computation:

\[ \frac{\partial \mathcal{L}_\Lambda}{\partial \lambda_k} = \sum_{l=1}^{L} \left[ \sum_s P_\Lambda (s | w^l, h^l, p^l) F_k (w^l, h^l, s) \right. \]

\[ - \sum_s P_\Lambda (s | w^l, h^l) F_k (w^l, h^l, s) \left. \right] - \frac{1}{\sigma^2} \lambda_k. \]
### Features

#### Node Features

<table>
<thead>
<tr>
<th></th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>$s_i$</td>
<td>polarity</td>
</tr>
<tr>
<td>b</td>
<td>$s_i &amp; q_i$</td>
<td>prior polarity</td>
</tr>
<tr>
<td>c</td>
<td>$s_i &amp; q_i &amp; r_i$</td>
<td>polarity reversal</td>
</tr>
<tr>
<td>d</td>
<td>$s_i &amp; u_{i,1}, \ldots, s_i &amp; u_{i,m_i}$</td>
<td>number of words</td>
</tr>
<tr>
<td>e</td>
<td>$s_i &amp; c_{i,1}, \ldots, s_i &amp; c_{i,m_i}$</td>
<td>surface form</td>
</tr>
<tr>
<td>f</td>
<td>$s_i &amp; f_{i,1}, \ldots, s_i &amp; f_{i,m_i}$</td>
<td>base form</td>
</tr>
<tr>
<td>g</td>
<td>$s_i &amp; u_{i,1} &amp; u_{i,2}, \ldots, s_i &amp; u_{i,m_i}$</td>
<td>coarse POS tag</td>
</tr>
<tr>
<td>h</td>
<td>$s_i &amp; b_{i,1} &amp; b_{i,2}, \ldots, s_i &amp; b_{i,m_i}$</td>
<td>finePOS tag</td>
</tr>
</tbody>
</table>

#### Edge Features

<table>
<thead>
<tr>
<th></th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$s_i &amp; s_j$</td>
<td>i ranges over all phrases</td>
</tr>
<tr>
<td>B</td>
<td>$s_i &amp; s_j &amp; r_j$</td>
<td>k ranges over all words in phrase i</td>
</tr>
<tr>
<td>C</td>
<td>$s_i &amp; s_j &amp; r_j &amp; q_j$</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>$s_i &amp; b_{i,1} &amp; b_{i,2}, \ldots, s_i &amp; s_j &amp; b_{i,m_i}$</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>$s_i &amp; s_j &amp; b_{j,1}, \ldots, s_i &amp; s_j &amp; b_{j,m_j}$</td>
<td></td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Japanese</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACP</td>
<td>KNB</td>
<td>NTC-J</td>
<td>50 Topics</td>
<td>CR</td>
<td>MPQA</td>
<td>MR</td>
<td>NTC-E</td>
<td></td>
</tr>
<tr>
<td>Voting-w/o Rev.</td>
<td>0.686</td>
<td>0.764</td>
<td>0.665</td>
<td>0.727</td>
<td>0.714</td>
<td>0.804</td>
<td>0.629</td>
<td>0.730</td>
<td></td>
</tr>
<tr>
<td>Voting-w/ Rev.</td>
<td>0.732</td>
<td>0.792</td>
<td>0.714</td>
<td>0.765</td>
<td>0.742</td>
<td>0.817</td>
<td>0.631</td>
<td>0.740</td>
<td></td>
</tr>
<tr>
<td>Rule</td>
<td>0.734</td>
<td>0.792</td>
<td>0.742</td>
<td>0.764</td>
<td>0.743</td>
<td>0.818</td>
<td>0.629</td>
<td>0.750</td>
<td></td>
</tr>
<tr>
<td>BoF-no Dic.</td>
<td>0.798</td>
<td>0.758</td>
<td>0.754</td>
<td>0.761</td>
<td>0.793</td>
<td>0.818</td>
<td>0.757</td>
<td>0.768</td>
<td></td>
</tr>
<tr>
<td>BoF-w/o Rev.</td>
<td>0.812</td>
<td>0.823</td>
<td>0.794</td>
<td>0.805</td>
<td>0.802</td>
<td>0.840</td>
<td>0.761</td>
<td>0.793</td>
<td></td>
</tr>
<tr>
<td>BoF-w/ Rev.</td>
<td>0.822</td>
<td>0.830</td>
<td>0.804</td>
<td>0.819</td>
<td>0.814</td>
<td>0.841</td>
<td>0.764</td>
<td>0.797</td>
<td></td>
</tr>
<tr>
<td>Tree-CRF</td>
<td><strong>0.846</strong>*</td>
<td><strong>0.847</strong>*</td>
<td><strong>0.826</strong>*</td>
<td><strong>0.841</strong>*</td>
<td><strong>0.814</strong>*</td>
<td><strong>0.861</strong>*</td>
<td><strong>0.773</strong>*</td>
<td><strong>0.804</strong></td>
<td></td>
</tr>
</tbody>
</table>
Summary

• Probabilistic graphical models: Models of joint probability
• CRF: A discriminative model for conditional probabilities
• Training through MLE
• Decisions using CRFs