Machine Learning for Code Analytics

Martin Vechev    Veselin Raychev

Department of Computer Science
ETH Zurich
Machine Learning for Programming

Probabilistically likely solutions to problems impossible to solve otherwise

Publications:
- Predicting Program Properties from “Big Code”, ACM POPL’15
- Programming with Big Code, SNAPL 2015
- Code Completion with Statistical Language Models, ACM PLDI’14
- Machine Translation for Programming Languages, ACM Onward’14
- Bug Localization with Statistical Language Models, ETH TR
- Fast and Precise Statistical Code Completion, ETH TR

Tools:
- JSNICE (used worldwide)
  statistical de-obfuscation
- SLANG
  statistical code synthesis
- SAGE
  statistical feedback generation

More information: http://www.srl.inf.ethz.ch/
Tutorial Outline

• Motivation
  • Potential applications

• Statistical language models
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• Graphical Models
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
Machine Learning for Programming

Applications

Intermediate Representation

Analyze Program (PL)

Train Model (ML)

Query Model (ML)
# Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Deobfuscation</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Representation</td>
<td>Sequences (sentences)</td>
<td>Translation Table</td>
<td>Trees</td>
<td>Graphical Models (CRFs)</td>
<td>Feature Vectors</td>
</tr>
<tr>
<td>Analyze Program (PL)</td>
<td>typestate analysis</td>
<td>scope analysis</td>
<td>control-flow analysis</td>
<td>alias analysis</td>
<td></td>
</tr>
<tr>
<td>Train Model (ML)</td>
<td>Neural Networks</td>
<td>SVM</td>
<td>Structured SVM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Query Model (ML)           | \[
| \text{argmax} \ P(y \mid x) \\
| y \in \Omega
| Greedy MAP inference |
Number of repositories

10M repos

How can we leverage “Big Code”?
Scene Completion

Input
Scene Completion

Camera camera = Camera.open();
camera.setDisplayOrientation(90);
Camera camera = Camera.open();
camera.setDisplayOrientation(90);

Camera camera = Camera.open();
camera.setDisplayOrientation(90);
camera.unlock();

SurfaceHolder holder = getHolder();
holder.addCallback(this);
holder.setType(SurfaceHolder.STP);
MediaRecorder r = new MediaRecorder();
r.setCamera(camera);
r.setAudioSource(MediaRecorder.AS);
r.setVideoSource(MediaRecorder.VS);
r.setOutFormat(MediaRecorder.MPEG4);
Martin is talking at the TCE now.
Programming Language Translation

C#  Java  Translate

Console.WriteLine("Hi");
...

System.out.println("Hi");
...

“Phrase-based statistical translation of programming languages”, S. Karaivanov, V. Raychev, M. V., Onward’14
Image de-noisification
function FZ(e, t) {
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t) {
        if (i + t < r) n.push(e.substring(i, i + t));
        else n.push(e.substring(i, r));
    }
    return n;
}

function chunkData(str, step) {
    var colNames = [];
    var len = str.length;
    var i = 0;
    for (; i < len; i += step) {
        if (i + step < len) colNames.push(str.substring(i, i + step));
        else colNames.push(str.substring(i, len));
    }
    return colNames;
}

“Predicting program properties from Big Code”, V. Raychev, M. V., A. Krause, ACM POPL’15
**JSNice.org: Impact**

- Used in **191 countries**
- **1,000+ Tweets**
- **Top ranked tool** for JavaScript
- **30,000 users** in 1st week

---

"Predicting program properties from Big Code", V. Raychev, M. V., A. Krause, ACM POPL'15
Tutorial Outline

• Motivation
  • Potential applications

• Statistical language models
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• Graphical Models
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
Probability Refresher

• (Discrete) probability distribution: \[ \sum_x P(x) = 1 \]

• For a joint probability distribution \( P(x,y) \), the distribution of a single variable, called the marginal, is: \[ \sum_y P(x,y) = 1 \]

• Conditional distribution: \( P(y \mid x) = \frac{P(y, x)}{P(x)} \)

• Bayes’s rule: \( P(y \mid x) = \frac{P(x \mid y) \times P(y)}{P(x)} \)
Probability Refresher

- **Independence:** \( P(x, y) = P(x) \times P(y) \)
  - written as: \( x \perp y \)

- **Conditional Independence:** \( P(X, Y \mid Z) = P(X \mid Z) \times P(Y \mid Z) \)
  - Set \( X \) is independent of set \( Y \) provided we know the value of variables in set \( Z \).
  - written as \( X \perp Y \mid Z \)
# Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Representation</td>
<td>Sequences (sentences)</td>
<td>Translation Table</td>
<td>Trees</td>
<td>Graphical Models (CRFs) Feature Vectors</td>
</tr>
<tr>
<td>Analyze Program (PL)</td>
<td>typestate analysis</td>
<td>control-flow analysis</td>
<td>alias analysis</td>
<td></td>
</tr>
<tr>
<td>Train Model (ML)</td>
<td>Neural Networks</td>
<td>SVM</td>
<td>Structured SVM</td>
<td></td>
</tr>
<tr>
<td>Query Model (ML)</td>
<td>$\text{argmax } P(y \mid x)$</td>
<td>$y \in \Omega$</td>
<td>Greedy MAP inference</td>
<td></td>
</tr>
</tbody>
</table>
Motivation: Working with APIs
Camera camera = Camera.open();
camera.setDisplayOrientation(90);

MediaRecorder rec = new MediaRecorder();

rec.setAudioSource(MediaRecorder.AudioSource.MIC);
rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);
rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
rec.setAudioEncoder(1);
rec.setVideoEncoder(3);

rec.setOutputFile("file.mp4");
...

[Statistical Code Completion: Capabilities]
[V. Raychev, M. Vechev, E. Yahav, PLDI’14]
Camera camera = Camera.open();
camera.setDisplayOrientation(90);
camera.unlock();

MediaRecorder rec = new MediaRecorder();

rec.setCamera(camera);
rec.setAudioSource(MediaRecorder.AudioSource.MIC);
rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);
rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
rec.setAudioEncoder(1);
rec.setVideoEncoder(3);
rec.setOutputFile("file.mp4");
...
Camera camera = Camera.open();
camera.setDisplayOrientation(90);
camera.unlock();

MediaRecorder rec = new MediaRecorder();

rec.setCamera(camera);
rec.setAudioSource(MediaRecorder.AudioSource.MIC);
rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);
rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
rec.setAudioEncoder(1);
rec.setVideoEncoder(3);
rec.setOutputFile("file.mp4");
...

Handles multiple objects
Camera camera = Camera.open();
camera.setDisplayOrientation(90);
camera.unlock();

MediaRecorder rec = new MediaRecorder();
rec.setCamera(camera);
rec.setAudioSource(MediaRecorder.AudioSource.MIC);
rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);
rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
rec.setAudioEncoder(1);
rec.setVideoEncoder(3);
rec.setOutputFile("file.mp4");
...

Handles multiple objects

Infers multiple statements
Camera camera = Camera.open();
camera.setDisplayOrientation(90);
camera.unlock();

MediaRecorder rec = new MediaRecorder();

rec.setCamera(camera);

rec.setAudioSource(MediaRecorder.AudioSource.MIC);
rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);
rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);

rec.setAudioEncoder(1);
rec.setVideoEncoder(3);
rec.setOutputFile("file.mp4");

...
Key Insight

Regularities in code are similar to regularities in natural language
Key Insight

Regularities in code are similar to regularities in natural language

We want to learn that
    MediaRecorder rec = new MediaRecorder();
is before
    rec.setCamera(camera);
Regularities in code are similar to regularities in natural language.

We want to learn that

```java
MediaRecorder rec = new MediaRecorder();
rec.setCamera(camera);
```

is before

```
like in natural languages
Hello
is before
World!
```
The SLANG System

- **Incomplete Program**: Language model
- **Completion Phase**: Program Analysis → Query → Combine
- **Training Phase**: Program Analysis → Train Language Model

**Diagram Components**:
- Camera: `Camera camera = Camera.open(); camera.setDisplayOrientation(90);`
- MediaRecorder
- dioSource.MIC
- rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT)
- rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4)
- rec.setAudioEncoder(1)
- rec.setOutputFile("file.mp4")
The SLANG System
she = new X();
me = new Y();

me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);
From Programs to Sentences

```java
she = new X();
me = new Y();

me.sleep();
if (random()) {
    me.eat();
}

she.enter();
me.talk(she);
```

Typestate analysis
From Programs to Sentences

she = new X();
me = new Y();
me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);

Typestate analysis
Alias analysis
From Programs to Sentences

```
she = new X();
me = new Y();
me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);
```

Typestate analysis
Alias analysis

for abstract object me:
From Programs to Sentences

```
she = new X();
me = new Y();
me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);
```

Typestate analysis
Alias analysis

```
for abstract object me:
    Y_init sleep talk
```
From Programs to Sentences

```
she = new X();
me = new Y();

me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);
```

Typestate analysis
Alias analysis

for abstract object `me`:

```
Y_{init}  sleep  talk
Y_{init}  sleep  eat  talk
```
she = new X();
me = new Y();
me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);

Typestate analysis
Alias analysis

for abstract object **me**:  
Y_{init} sleep talk  
Y_{init} sleep eat talk

for abstract object **she**:  

From Programs to Sentences

```java
she = new X();
me = new Y();
me.sleep();
if (random()) {
    me.eat();
}
she.enter();
me.talk(she);
```

Typestate analysis
Alias analysis

for abstract object `me`:

- `Y_{init} sleep talk`
- `Y_{init} sleep eat talk`

for abstract object `she`:

- `X_{init} enter talk_{param1}`
Learn Regularities

Learn regularities in obtained sentences

Regularities in sentences $\Leftrightarrow$ regularities in API usage

If we see many sequences like:

$$x_{\text{init}} \text{ enter} \ talk_{\text{param1}}$$

then we should learn that $talk_{\text{param1}}$ is often after enter
**Statistical Language Models**

Given a sentence \( s = w_1 w_2 w_3 ... w_n \)

estimate \( P( w_1 w_2 w_3 ... w_n ) \)

Decomposed to conditional probabilities

\[
P( w_1 w_2 w_3 ... w_n ) = \prod_{i=1..n} P( w_i | w_1 ... w_{i-1} )
\]
Statistical Language Models

Given a sentence $s = w_1 w_2 w_3 \ldots w_n$

estimate $P( w_1 w_2 w_3 \ldots w_n )$

Decomposed to conditional probabilities

$$P( w_1 w_2 w_3 \ldots w_n ) = \prod_{i=1\ldots n} P( w_i \mid w_1 \ldots w_{i-1} )$$

$$P( \text{The quick brown fox jumped} ) =$$

$$P( \text{The} ) \ P( \text{quick} \mid \text{The} ) \ P( \text{brown} \mid \text{The quick} )$$

$$P( \text{fox} \mid \text{The quick brown} ) \ P( \text{jumped} \mid \text{The quick brown fox} )$$
N-gram language model

Conditional probability only on previous $n-1$ words

$$P( w_i \mid w_1 \ldots w_{i-1} ) \approx P( w_i \mid w_{i-n+1} \ldots w_{i-1} )$$
N-gram language model

Conditional probability only on previous $n-1$ words

$$P(w_i | w_1 \ldots w_{i-1}) \approx P(w_i | w_{i-n+1} \ldots w_{i-1})$$

$n-1$ words
N-gram language model

Conditional probability only on previous n-1 words

\[
P(w_i \mid w_1 \ldots w_{i-1}) \approx P(w_i \mid w_{i-n+1} \ldots w_{i-1})
\]

Training is achieved by counting n-grams. E.g., with 3-gram language model, we get:

\[
P(\text{jumped} \mid \text{The quick brown fox}) \approx P(\text{jumped} \mid \text{brown fox}) \approx \frac{\#(\text{brown fox jumped})}{\#(\text{brown fox})}
\]

\#(n-gram) - number of occurrences of n-gram in training data

Time complexity for each word encountered in training is constant, so training is usually fast.
Tri-gram language model

\[ P (w_1 \cdot w_2 \cdot w_3 \cdot \ldots \cdot w_n) \sim = P (w_1) \times P (w_2 \mid w_1) \times P (w_3 \mid w_1 \cdot w_2) \times \ldots \times P (w_n \mid w_{n-2} \cdot w_{n-1}) \]

<table>
<thead>
<tr>
<th>3-grams</th>
<th># of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>brown fox jumped</td>
<td>125</td>
</tr>
<tr>
<td>brown fox walked</td>
<td>45</td>
</tr>
<tr>
<td>brown fox snapped</td>
<td>30</td>
</tr>
</tbody>
</table>

\[
P \text{ (jumped} \mid \text{ brown fox)} \sim = \frac{125}{200} \sim = 0.625
\]

\[
P \text{ (brown fox jumped)} \sim =
\frac{P \text{ (brown)}}{600} \times \frac{P \text{ (fox} \mid \text{ brown)}}{200} \times \frac{P \text{ (jumped} \mid \text{ brown fox)}}{125} \sim = 0.208
\]
Key Problem: Sparsity of Data

What if this number is 0?

\[
P(\text{jumped} \mid \text{brown fox}) \approx \frac{\#(\text{brown fox jumped})}{\#(\text{brown fox})}
\]

The problem of sparsity gets worse as the size of the n-gram becomes larger.

We need to handle n-grams with 0 or few occurrences in the training data. Techniques that can do that are: smoothing, discounting
Solution: Smoothing Techniques

• Smoothing techniques give non-zero probability to n-grams not in the training data.

• Essentially, they try to estimate how likely it is that the n-gram is missing due to the limited size of the training data.

• They work by taking the probability mass of the existing n-grams and redistributing that mass over n-grams that occur zero times.

• Typically, probability of such n-grams is estimated by looking at the probabilities of n-1 grams, n-2 grams, unigrams, etc.
Smoothing: Intuitively

Distribution of probability mass before smoothing

Training Data

All other sentences get 0 probability

Smoothing

Distribution of probability mass after smoothing

Training Data

All other sentences get non-zero probability
Querying the Probability Distribution

All scoring techniques that use smoothing follow this template:

\[
P( w_i | w_{i-N+1} ) = \begin{cases} 
P( w_i | w_{i-N+1}^{i-1} ) & \text{if } w_{i-N+1}^i \text{ in training data} \\
B(w_{i-N+1}^{i-1}) \times P( w_i | w_{i-N+2}^{i-1} ) & \text{otherwise}
\end{cases}
\]

Smoothing techniques differ by how they define this function

Example smoothing techniques are: Witten-Bell Interpolated (WBI), Witten-Bell Backoff (WBB), Natural Discounting (ND), Stupid-Backoff (SB)
Training: Witten-Bell Interpolated

\[ P_{wbi}(w_i \mid w_{i-N+1}^{i-1}) = F_{wbi}(w_{i-N+1}^i) + B_{wbi}(w_{i-N+1}^{i-1}) \times P_{wbi}(w_i \mid w_{i-N+2}^{i-1}) \]

\[ F_{wbi}(w_{i-N+1}^i) = (1 - B_{wbi}(w_{i-N+1}^{i-1})) \times P_{ML}(w_i \mid w_{i-N+1}^{i-1}) \]

\[ B_{wbi}(w_{i-N+1}^{i-1}) = \frac{N(w_{i-N+1}^{i-1})}{N(w_{i-N+1}^{i-1}) + \#(w_{i-N+1}^{i-1})} \]

\[ N(w_s) = \left| \{ w : \#(w_s, w) > 0 \} \right| \]

Number of distinct followers of \( w_s \)

In the training data
LM Training Algorithm

1. Count all grams of length 1,..., N found in the training data

2. For each K-gram (0 < K < N) $w_{i-K+1}$ in training data, calculate:

   $B_{wbi} (w_{i-K+1})$, $P_{wbi} (w_i | w_{i-K+1})$

3. For the largest N-grams $w_{i-N+1}$ in training data, calculate:

   $P_{wbi} (w_i | w_{i-N+1})$
N-gram language model

Conditional probability only on previous \( n-1 \) words

\[
P( w_i \mid w_1 \ldots w_{i-1} ) \approx P( w_i \mid w_{i-n+1} \ldots w_{i-1} )
\]

Existing library implementing language models is SRILM:

http://www.speech.sri.com/projects/srilm/
Recurrent Neural Networks (RNN)

RNNs can learn dependencies beyond the prior several words

A neural network with internal state that stores probabilistic information about all previous words in a sentence. With this, it can capture relationship between \textit{quick} and \textit{jumped} in:

\[ P(\text{jumped} | \text{The quick brown fox}) \]
Recurrent Neural Networks (RNN)

RNNs can learn dependencies beyond the prior several words

A neural network with internal state that stores probabilistic information about all previous words in a sentence. With this, it can capture relationship between quick and jumped in:

\[ P( \text{jumped} | \text{The quick brown fox}) \]

Matrices U, V and W learned through back propagation with time and some unfolding.

Time complexity for training RNNLM quadratic in size of vocabulary (for each word encountered in training), can be slow

Evaluation of a sentence \( s = w_1 \cdot \ldots w_n \) proceeds on a word-by-word basis keeping the last \( s(t) \).
Recurrent Neural Networks (RNNs) can learn dependencies beyond the prior several words.

A neural network with internal state that stores probabilistic information about all previous words in a sentence. With this, it can capture the relationship between quick and jumped in:

\[ P(\text{jumped} \mid \text{The quick brown fox}) \]

Matrices U, V and W are learned through back propagation with time and some unfolding.

Time complexity for training RNNLM is quadratic in size of vocabulary (for each word encountered in training), can be slow.

Evaluation of a sentence \( s = w_1 \cdot \ldots \cdot w_n \) proceeds on a word-by-word basis keeping the last \( s(t) \).
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
Code Completion

Abstract object `smsMgr`:
- `getDefault_result divideMessage H1`
- `getDefault_result H2`

```java
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
```
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
Code Completion

```java
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
```

Abstract object **smsMgr**:
- `getDefault_result` divideMessage H1
- `getDefault_result` H2

Abstract object **list**:
- `divideMessage_result` H1
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    ? {smsMgr, list} // (Hole H1)
} else {
    ? {smsMgr, message} // (Hole H2)
}
Code Completion

```
getDefault_result  divideMessage H1
```

```
getDefault_result H2
```

```
divideMessage_result H1
```

```
length H2
```
## Code Completion

<table>
<thead>
<tr>
<th>Function</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getDefault(result divideMessage sendMultipartTextMessage)</code></td>
<td>0.0033</td>
</tr>
<tr>
<td><code>getDefault(result divideMessage sendTextMessage)</code></td>
<td>0.0016</td>
</tr>
<tr>
<td><code>getDefault(result sendTextMessage)</code></td>
<td>0.0073</td>
</tr>
<tr>
<td><code>getDefault(result sendMultipartTextMessage)</code></td>
<td>0.0010</td>
</tr>
<tr>
<td><code>divideMessage(result sendMultipartTextMessage) param3</code></td>
<td>0.0821</td>
</tr>
<tr>
<td><code>length length</code></td>
<td>0.0132</td>
</tr>
<tr>
<td><code>length split</code></td>
<td>0.0080</td>
</tr>
<tr>
<td><code>length sendTextMessage param3</code></td>
<td>0.0017</td>
</tr>
<tr>
<td>Default</td>
<td>Result</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>sendMultipartTextMessage</td>
<td>0.0033</td>
</tr>
<tr>
<td>sendTextMessage</td>
<td>0.0016</td>
</tr>
<tr>
<td>sendTextMessage</td>
<td>0.0073</td>
</tr>
<tr>
<td>sendMultipartTextMessage</td>
<td>0.0010</td>
</tr>
<tr>
<td>sendMultipartTextMessage</td>
<td>0.0821</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Completion Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>length length</td>
<td>0.0132</td>
</tr>
<tr>
<td>length split</td>
<td>0.0080</td>
</tr>
<tr>
<td>length sendTextMessage</td>
<td>0.0017</td>
</tr>
</tbody>
</table>
## Code Completion

<table>
<thead>
<tr>
<th>Method</th>
<th>Completion Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getDefault</code></td>
<td></td>
</tr>
<tr>
<td><code>divideMessage</code></td>
<td></td>
</tr>
<tr>
<td><code>sendMultipartTextMessage</code></td>
<td>0.0033</td>
</tr>
<tr>
<td><code>getDefault</code></td>
<td></td>
</tr>
<tr>
<td><code>divideMessage</code></td>
<td></td>
</tr>
<tr>
<td><code>sendTextMessage</code></td>
<td>0.0016</td>
</tr>
<tr>
<td><code>getDefault</code></td>
<td></td>
</tr>
<tr>
<td><code>sendTextMessage</code></td>
<td>0.0073</td>
</tr>
<tr>
<td><code>divideMessage</code></td>
<td></td>
</tr>
<tr>
<td><code>sendMultipartTextMessage</code></td>
<td></td>
</tr>
<tr>
<td><code>length</code></td>
<td></td>
</tr>
<tr>
<td><code>length</code></td>
<td></td>
</tr>
<tr>
<td><code>length</code></td>
<td></td>
</tr>
<tr>
<td><code>split</code></td>
<td>0.0080</td>
</tr>
<tr>
<td><code>length</code></td>
<td></td>
</tr>
<tr>
<td><code>sendTextMessage</code></td>
<td></td>
</tr>
<tr>
<td><code>param3</code></td>
<td>0.0017</td>
</tr>
</tbody>
</table>

Not a feasible solution: completions disagree on selected method

The solution must satisfy program constraints
<table>
<thead>
<tr>
<th>Code Completion</th>
<th>Completion probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>getDefault\text{result} divideMessage \textbf{sendMultipartTextMessage}</td>
<td>0.0033</td>
</tr>
<tr>
<td>getDefault\text{result} divideMessage \textbf{sendTextMessage}</td>
<td>0.0016</td>
</tr>
<tr>
<td>getDefault\text{result} \textbf{sendTextMessage}</td>
<td>0.0073</td>
</tr>
<tr>
<td>getDefault\text{result} \textbf{sendMultipartTextMessage}</td>
<td>0.0010</td>
</tr>
<tr>
<td>\textbf{divideMessage}\text{result} \textbf{sendMultipartTextMessage}_{\text{param3}}</td>
<td>0.0821</td>
</tr>
<tr>
<td>\textbf{length} \textbf{length}</td>
<td>0.0132</td>
</tr>
<tr>
<td>\textbf{length} \textit{split}</td>
<td>0.0080</td>
</tr>
<tr>
<td>\textbf{length} \textbf{sendTextMessage}_{\text{param3}}</td>
<td>0.0017</td>
</tr>
<tr>
<td>Code Completion</td>
<td>Completion probability</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td><code>getDefault</code> result <code>divideMessage</code> <code>sendMultipartTextMessage</code></td>
<td>0.0033</td>
</tr>
<tr>
<td><code>getDefault</code> result <code>divideMessage</code> <code>sendTextMessage</code></td>
<td>0.0016</td>
</tr>
<tr>
<td><code>getDefault</code> result <code>sendTextMessage</code></td>
<td>0.0073</td>
</tr>
<tr>
<td><code>getDefault</code> result <code>sendMultipartTextMessage</code></td>
<td>0.0010</td>
</tr>
<tr>
<td><code>divideMessage</code> result <code>sendMultipartTextMessage</code>param3</td>
<td>0.0821</td>
</tr>
<tr>
<td><code>length length</code></td>
<td>0.0132</td>
</tr>
<tr>
<td><code>length split</code></td>
<td>0.0080</td>
</tr>
<tr>
<td><code>length sendTextMessage param3</code></td>
<td>0.0017</td>
</tr>
</tbody>
</table>
smsMgr = SmsManager.getDefault();
int length = message.length();
if (length > MAX_SMS_MESSAGE_LENGTH) {
    list = smsMgr.divideMessage(message);
    smsMgr.sendMultipartTextMessage(...list...);
} else {
    smsMgr.sendTextMessage(...message...);
}
The SLANG System

Completion phase

- Program Analysis
- Query
- Combine

Language model

Training phase

- Program Analysis
- Train Language Model

Correct completion in top 3 for ~90% of cases

~700MB

~100MB

Camera camera = Camera.open();
camera.setDisplayOrientation(90);
MediaRecorder();

84 testing samples

Object detection

1M Java methods

Sentences with holes

Sentences

Completed sentences

Completed program
Lessons

• **Language Models**
  • Easy and cheap to train
  • Important to deal with sparsity via smoothing
  • Connection to PL: decide what to "compile" into the word
  • Program Analysis need not be sound
  • Can be tricky to capture long distance relationships

• **Recurrent Networks**
  • Can capture longer distance relationships
  • More expensive to train
  • Experimentally similar to language models
Tutorial Outline

• **Motivation**
  • Potential applications

• **Statistical language models**
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• **Graphical Models**
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
# Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Deobfuscation</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Representation</td>
<td>Sequences (sentences)</td>
<td>Translation Table Trees</td>
<td>Graphical Models (CRFs) Feature Vectors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze Program (PL)</td>
<td>typestate analysis</td>
<td>scope analysis</td>
<td>control-flow analysis</td>
<td>alias analysis</td>
<td></td>
</tr>
<tr>
<td>Train Model (ML)</td>
<td>Neural Networks</td>
<td>N-gram language model</td>
<td>SVM</td>
<td>Structured SVM</td>
<td></td>
</tr>
<tr>
<td>Query Model (ML)</td>
<td>argmax $P(y \mid x)$</td>
<td>$y \in \Omega$</td>
<td>Greedy MAP inference</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deobfuscation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intermediate Representation</th>
<th>Sequences (sentences)</th>
<th>Translation Table</th>
<th>Graphical Models (CRFs)</th>
<th>Feature Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trees</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analyze Program (PL)</th>
<th>typestate analysis</th>
<th>scope analysis</th>
<th>control-flow analysis</th>
<th>alias analysis</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Train Model (ML)</th>
<th>Neural Networks</th>
<th>SVM</th>
<th>Structured SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N-gram language model</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Model (ML)</th>
<th>[ \text{argmax} \ P(y \mid x) ]</th>
<th>y \in \Omega</th>
<th>Greedy MAP inference</th>
</tr>
</thead>
</table>

# Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deobfuscation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intermediate Representation</th>
<th></th>
<th></th>
<th>Graphical Models (CRFs)</th>
<th>Feature Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequences (sentences)</td>
<td>Translation Table</td>
<td>Trees</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analyze Program (PL)</th>
<th></th>
<th></th>
<th>alias analysis</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>typestate analysis</td>
<td>control-flow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scope analysis</td>
<td>analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Train Model (ML)</th>
<th></th>
<th>SVM</th>
<th>Structured SVM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Networks</td>
<td></td>
<td>N-gram language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>language model</td>
<td></td>
<td>model</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Model (ML)</th>
<th></th>
<th></th>
<th>Greedy MAP inference</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[\arg\max_{y \in \Omega} P(y</td>
<td>x)]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

function \( f(a) \) {
    var \( b = \) document.getElementById(\( a \));
    return \( b \);
}
function $f(a)$ {
    var $b = document.getElementById(a);$ 
    return $b$;
}

unknown facts: $a$ $b$

known facts: $f$ $document$ $getElementById$
function f(a) {
    var b = document.getElementById(a);
    return b;
}

unknown facts:  a  b

known facts:  f  document  getElementById

Predict unknown facts given some known facts
Challenges

Facts to be predicted are dependent

Millions of candidate choices

Must quickly learn from huge codebases

Prediction should be fast (real time)
Key Idea

Phrase the problem of predicting program facts as

Structured Prediction for Programs
Structured Prediction for Programs
[V. Raychev, M. V., A. Krause, ACM POPL’15]

First connection between Programs and Conditional Random Fields
Factors

• A factor (or potential) $\varphi$ is a function from a set of random variables $D$ to a real number $R$

$$\varphi: D \rightarrow R$$

• The set of variables $D$ is the scope of the factor $\varphi$
  – we are typically concerned with non-negative factors

• Intuition: captures affinity, agreement, compatibility of the variables in $D$
Factors: Example

- Assume we have 4 (binary) random variables:
  - Alice, Bob, Ceco and Dobri

- Example factors
  - $\varphi_1 (0,0) = 30$ says we believe Alice and Bob agree on 0 with belief 30
  - $\varphi_3 (C,D)$ says that Ceco and Dobri argue all the time 😊

<table>
<thead>
<tr>
<th>$\varphi_1$ (A,B)</th>
<th>$\varphi_2$ (B,C)</th>
<th>$\varphi_3$ (C,D)</th>
<th>$\varphi_4$ (D,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A</td>
<td>B</td>
<td>value]</td>
<td>[B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>


# Factors: Example

- Assume we have 4 (binary) random variables:
  - Alice, Bob, Ceco and Dobri

- Example factors
  - $\varphi_1 (0,0) = 30$ says we believe Alice and Bob agree on 0 with belief 30
  - $\varphi_3 (C,D)$ says that Ceco and Dobri argue all the time :)
Defining a Global Probabilistic Model

\[ P(A, B, C, D) = \frac{F(A, B, C, D)}{Z(A, B, C, D)} \]

\[ F(A, B, C, D) = \varphi_1(A, B) \times \varphi_2(B, C) \times \varphi_3(C, D) \times \varphi_4(D, A) \]

\[ Z(A, B, C, D) = \sum_{a \in A, b \in B, c \in C, d \in D} F(a, b, c, d) \]
\[ P(A,B,C,D) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Unnormalized (F)</th>
<th>Normalized (P = F/Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300,000</td>
<td>0.04</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>300,000</td>
<td>0.04</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>300,000</td>
<td>0.04</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>(4.1 \times 10^{-6})</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

\[ Z(A,B,C,D) = 7,201,840 \]
\[ P(A,B,C,D) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Unnormalized ((F))</th>
<th>Normalized ((P = F/Z))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300,000</td>
<td>0.04</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>300,000</td>
<td>0.04</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>300,000</td>
<td>0.04</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td>(4.1 \times 10^{-6})</td>
</tr>
</tbody>
</table>

\[ Z(A,B,C,D) = 7,201,840 \]

We can now answer probability queries on \( P(A, B, C, D) \):

For example: \[ P(B = 1) = \sum_{a \in A} \sum_{c \in C} \sum_{d \in D} P(A, 1, C, D) \sim = 0.732 \]
Markov Network

Let $X = \{X_1, X_2, ..., X_n\}$ be a set of random variables and $D_1, D_2, ..., D_m \subseteq X$

Then, a Markov Network is defined as:

$$P(X_1, X_2, ..., X_n) = \frac{\varphi_1(D_1) \times \varphi_2(D_2) \times ... \times \varphi_m(D_m)}{Z(X_1, X_2, ..., X_n)}$$

$$Z(X_1, X_2, ..., X_n) = \sum \varphi_1(D_1) \times \varphi_2(D_2) \times ... \times \varphi_m(D_m)$$
Markov Network

Let $X = \{X_1, X_2, ..., X_n\}$ be a set of random variables and $D_1, D_2, ..., D_m \subseteq X$

Then, a Markov Network is defined as:

$$P(X_1, X_2, ..., X_n) = \frac{\varphi_1(D_1) \times \varphi_2(D_2) \times ... \times \varphi_m(D_m)}{Z(X_1, X_2, ..., X_n)}$$

$$Z(X_1, X_2, ..., X_n) = \sum \varphi_1(D_1) \times \varphi_2(D_2) \times ... \times \varphi_m(D_m)$$

If the factors are strictly positive, then $P$ is called a Gibbs distribution.

If the domains of all factors is of size 2, the network is called pairwise.
Graphs vs. Probability Distributions

We will next relate graphs and probability distributions. This is important for two reasons.

First, it allows us to determine properties (e.g., independence of variables) of a probability distribution directly from the graph.

Second, it allows us to answer queries (e.g., MAP inference) on the probability distribution by working with the graph.
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$. 
Graph Factorization

A distribution $P$ equipped with a set of factors $\phi_1(D_1), \ldots, \phi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$

$$P(A, B, C, D) = \frac{\phi_1(A, B) \times \phi_2(B, D) \times \phi_3(D, C) \times \phi_4(C, A)}{Z}$$
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

$$P(A, B, C, D) = \varphi_1(A, B) \times \varphi_2(B, D) \times \varphi_3(D, C) \times \varphi_4(C, A) / Z$$

Here, $P$ factorizes over $G$. 
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

\[
P(A, B, C, D) = \frac{\varphi_1(A, B) \times \varphi_2(B, D) \times \varphi_3(D, C) \times \varphi_4(C, A)}{Z}
\]

Here, $P$ factorizes over $G$.

\[
P(A, B, C, D) = \frac{\varphi_1(A, B, C) \times \varphi_2(B, D, C)}{Z}
\]
Graph Factorization

A distribution P equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph G if each $D_i$ is a complete subgraph of G.

$$P(A, B, C, D) = \varphi_1(A, B) \times \varphi_2(B, D) \times \varphi_3(D, C) \times \varphi_4(C, A) / Z$$

Here, $P$ factorizes over G

$$P(A, B, C, D) = \varphi_1(A, B, C) \times \varphi_2(B, D, C) / Z$$

Here, $P$ does not factorize over G
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

$P(A, B, C, D) = \varphi_1(A, B, C) \times \varphi_2(B, C, D) / Z$
A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

$$P(A, B, C, D) = \varphi_1(A, B, C) \times \varphi_2(B, C, D) / Z$$

Here, $P$ factorizes over $G$. 
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

$P(A, B, C, D) = \varphi_1(A, B, C) \times \varphi_2(B, C, D) / Z$

$P(A, B, C, D) = \varphi_1(A, B, C, D) / Z$

Here, $P$ factorizes over $G$. 
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

\[ P(A, B, C, D) = \frac{\varphi_1(A, B, C) \times \varphi_2(B, C, D)}{Z} \]

Here, $P$ factorizes over $G$.

\[ P(A, B, C, D) = \frac{\varphi_1(A, B, C, D)}{Z} \]

Here, $P$ factorizes over $G$. 
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$

$$P(A, B, C, D) = \varphi_1(A, B, C) \times \varphi_2(B, C, D) \times \varphi_3(B, E) \times \varphi_4(B, F) / Z$$
Graph Factorization

A distribution $P$ equipped with a set of factors $\varphi_1(D_1), \ldots, \varphi_m(D_m)$ factorizes over a graph $G$ if each $D_i$ is a complete subgraph of $G$.

$$\mathbf{P}(A, B, C, D) = \varphi_1(A, B, C) \times \varphi_2(B, C, D) \times \varphi_3(B, E) \times \varphi_4(B, F) / Z$$

Here, $P$ factorizes over $G$. 
Graphs vs. Probabilities

There is an important connection between a probability distribution that factorizes over a graph and the properties of the graph.

In particular, we can discover various independence properties of the probabilistic distribution directly from the graph.
Graph Separation

Two sets of disjoint nodes A and B in a graph G are separated by a set S, if every path between A and B goes through a node in S. If there is no path between A and B then A and B are separated. A and B are also separated if $S = \emptyset$ and no path between A and B exists.

The Global Markov Property says that if A and B are separated by S, then

$$A \perp B \mid S \quad \text{that is} \quad P(A, B \mid S) = P(A \mid S) \times P(B \mid S)$$
Graph Separation

Two sets of disjoint nodes A and B in a graph G are separated by a set S, if every path between A and B goes through a node in S. If there is no path between A and B then A and B are separated. A and B are also separated if \( S = \emptyset \) and no path between A and B exists.

The Global Markov Property says that if A and B are separated by S, then

\[
A \perp B \mid S \quad \text{that is} \quad P(A, B \mid S) = P(A \mid S) \times P(B \mid S)
\]

An algorithm to determine all global independencies \( A \perp B \mid S \) is:

1. Given S, remove all edges adjacent to nodes in S resulting in a new graph G′
2. Check if a path between nodes in A and B exists in G′
Graph Separation: Example
Graph Separation: Example

Example of global independencies found in the above graph:

\[
\begin{align*}
\{A\} & \perp \{G\} \mid \{D\} \\
\{B\} & \perp \{F\} \mid \{E,G\}
\end{align*}
\]
Independence Assertions

We use the notation $\mathbf{I}(P)$ to denote the set of conditional independence assertions of the form $X \perp Y \mid Z$ for the probability distribution $P$.

Similarly, we use $\mathbf{I}(G)$ to denote the set of assertions of the above kind exhibited by a graph $G$ – these are called the **Global Markov assertions**.

$G$ is called an **I-map** of $P$ if $\mathbf{I}(G) \subseteq \mathbf{I}(P)$.
Independence Assertions

We use the notation $I(P)$ to denote the set of conditional independence assertions of the form $X \perp Y \mid Z$ for the probability distribution $P$.

Similarly, we use $I(G)$ to denote the set of assertions of the above kind exhibited by a graph $G$ – these are called the Global Markov assertions.

$G$ is called an $I$-map of $P$ if $I(G) \subseteq I(P)$.

For positive distributions (every value in the range is $> 0$), we have:

**Theorem:** $P$ factorizes over $G$ iff $I(G) \subseteq I(P)$.

If the distribution is not positive, then there is a counter example to the above theorem in one direction.
Counter-example to IFF

Consider the distribution $P$ over 4 variables $A$, $B$, $C$ and $D$ where:

\[
\begin{align*}
P(0, 0, 0, 0) &= 1/8 & P(1, 0, 0, 0) &= 1/8 & P(1, 1, 0, 0) &= 1/8 & P(1, 1, 1, 0) &= 1/8 \\
P(0, 0, 0, 1) &= 1/8 & P(0, 0, 1, 1) &= 1/8 & P(0, 1, 1, 1) &= 1/8 & P(1, 1, 1, 1) &= 1/8
\end{align*}
\]

Here, $P$ is not a positive distribution as for all other values of $A,B,C,D$, it is 0.

Consider the graph $G$:

- $I(G)$:
  - $\{A\} \perp \{C\} \mid \{B, D\}$
  - $\{B\} \perp \{D\} \mid \{A, C\}$
Counter-example to IFF

Consider the distribution $P$ over 4 variables $A$, $B$, $C$ and $D$ where:

- $P(0, 0, 0, 0) = 1/8$
- $P(1, 0, 0, 0) = 1/8$
- $P(1, 1, 0, 0) = 1/8$
- $P(1, 1, 1, 0) = 1/8$
- $P(0, 0, 0, 1) = 1/8$
- $P(0, 0, 1, 1) = 1/8$
- $P(0, 1, 1, 1) = 1/8$
- $P(1, 1, 1, 1) = 1/8$

Here, $P$ is not a positive distribution as for all other values of $A,B,C,D$, it is 0.

Consider the graph $G$:

Here we have that $I(G) \subseteq I(P)$, yet $P$ does not factorize according to $G$.
Independence Assertions on Graphs

**Pairwise Independence:**
\[ I_p(G) = \{(X_1 \perp X_2 \mid X \setminus \{X_1, X_2\}) : \text{edge } (X_1, X_2) \notin G\} \]

**Local Independence:**
\[ I_l(G) = \{X_1 \perp X \mid X \setminus \{X_1\} - MB_G(X_1) \mid MB_G(X_1)\} \]

\( MB_G(X_1) \) is the Markov Blanket of node \( X_1 \): the neighbors of node \( X_1 \) in \( G \)

For **positive** distributions, the following statements are equivalent:
\[ P \models I_p(G) \quad P \models I_l(G) \quad P \models I(G) \]

For **general** distributions, we have a weaker relationship:
\[ \text{If } P \models I(G) \text{ then } P \models I_l(G) \]
\[ \text{If } P \models I_l(G) \text{ then } P \models I_p(G) \]
Graph Separation: Examples

Example of global independence $l(G)$: $\{A\} \perp \{G\} | \{D\}$

Example of pairwise independence $l_p(G)$: $A \perp G | \{B,C,D,E,F\}$

Example of local independence $l_l(G)$: $A \perp \{D, E,F,G\} | \{B,C\}$
Graphical Models: So far

So far we learned what a Markov Network is, what it means for a probability distribution to factor over a graph and how to extract information about independence of variables from the graph.

We next look at two equivalent representation of Markov Networks, one of which is amenable to inference queries (factor graphs) and another which is amenable to learning from data (e.g., log-linear form).
Log-Linear Models

Another representation of a Markov Network is that of a log-linear model.

Here, we have a set of feature functions $f_1(D_1), \ldots, f_k(D_k)$ where each $D_i$ is a complete subgraph of $G$. An $f_i(D_i)$ can return any value, i.e., can be negative.

$$P(X_1, \ldots, X_n) = \frac{1}{Z(X_1, \ldots, X_n)} \exp - \sum_{i=1}^{k} (w_i \times f_i(D_i))$$

$$Z(X_1, \ldots, X_n) = \sum \exp - \sum_{i=1}^{k} (w_i \times f_i(D_i))$$

Any Markov Network whose factors are positive can be converted to a log-linear model.
Log-Linear Models to Markov Networks

Let \( \varphi_i(D_i) = \exp^{-w_i \times f_i(D_i)} \)

Then
\[
\exp^{-\sum_{i=1}^{k}(w_i \times f_i(D_i))} = \\
\exp(-w_1 \times f_1(D_1) - w_2 \times f_2(D_2) \ldots - w_k \times f_k(D_k)) = \\
\exp^{-w_1 \times f_1(D_1)} \times \exp^{-w_2 \times f_2(D_2)} \ldots \times \exp(-w_k \times f_k(D_k)) = \\
\varphi_1(D_1) \times \varphi_2(D_2) \ldots \times \varphi_k(D_k)
\]

By substitution:
\[
P(X_1, X_2, \ldots, X_n) = \frac{\varphi_1(D_1) \times \varphi_2(D_2) \times \ldots \times \varphi_k(D_k)}{Z(X_1, X_2, \ldots, X_n)}
\]
Log-Linear Representation: Benefits

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

\[
Z(X_1, \ldots, X_n) = \sum \exp\left(-\sum_{i=1}^{k} (w_i \times f_i(D_i))\right)
\]

Log-linear models have few benefits over factors. They:

1. Make certain relationships more explicit

2. Offer a much more compact representation for many distributions, especially for variables with large domains (e.g., names in JSNice)

3. They are useful for learning where we are given the feature functions and the learning phase simply figures out the weights.
Feature Function: Example

For example, suppose that for the factor \( \varphi_1 \):

\[
\varphi_1 (A, B) = 148.41 \quad \text{if} \quad A = B
\]
\[
1 \quad \text{otherwise}
\]

If A and B range over \( m \) and \( n \) values respectively, we have \( m \times n \) possible values to store in order to keep this factor.

Instead, we can use a single feature function \( f_{A \neq B}(A, B) \) where

\[
f_1(A, B) = 1 \quad \text{if} \quad A = B \quad \text{and} \quad w_1 = -5
\]
\[
0 \quad \text{else}
\]
Factors vs. Features + Weights

<table>
<thead>
<tr>
<th>$\varphi_1$ (A,B)</th>
<th>$\varphi_2$ (B,C)</th>
<th>$\varphi_3$ (C,D)</th>
<th>$\varphi_4$ (D,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>value</strong></td>
<td><strong>value</strong></td>
<td><strong>value</strong></td>
<td><strong>value</strong></td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>A</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>
Factors vs. Features + Weights

Recall \( \varphi_i(D_i) = \exp^{-wi} \times f_i(D_i) \)

\( \varphi_1(A,B) \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>
Factors vs. Features + Weights

Recall $\varphi_i(D_i) = \exp^{-w_i \times f_i(D_i)}$

$\varphi_1(A,B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

$w_1 \times f_1(A,B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

$w_1 = 1$

$f_1(A,B) = \text{value}$
## Factors vs. Features + Weights

Recall \( \varphi_i(D_i) = \exp^{-w_i \times f_i(D_i)} \)

<table>
<thead>
<tr>
<th>( \varphi_1 ) (A,B)</th>
<th>( \varphi_2 ) (B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td><strong>B</strong></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th><strong>A</strong></th>
<th><strong>B</strong></th>
<th><strong>value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]
### Factors vs. Features + Weights

Recall \( \phi_i(D_i) = \exp^{-w_i \times f_i(D_i)} \)

<table>
<thead>
<tr>
<th>( \varphi_1 ) (A,B)</th>
<th>( \varphi_2 ) (B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( w_1 \times f_1(A, B) )</th>
<th>( w_2 \times f_2(B, C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\( w_1 = 1 \)
\( f_1(A, B) = \text{value} \)

\( w_2 = -4.61 \)
\( f_2(B, C) = 1 \) if B = C
\( 0 \) else
Factors vs. Features + Weights

Recall $\varphi_i(D_i) = \exp^{-w_i \times f_i(D_i)}$

<table>
<thead>
<tr>
<th>$\varphi_1$ (A,B)</th>
<th>$\varphi_2$ (B,C)</th>
<th>$\varphi_3$ (C,D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>value</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

$w_1 \times f_1(A, B)$

$w_2 \times f_2(B, C)$

| $w_1 = 1$ | $f_1(A, B) = \text{value}$ | $w_2 = -4.61$ | $f_2(B, C) = 1$ if $B = C$ | $0$ else |

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
<th>B</th>
<th>C</th>
<th>value</th>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
<td>0</td>
<td>0</td>
<td>-4.61</td>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
<td>1</td>
<td>1</td>
<td>-4.61</td>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>
Factors vs. Features + Weights

Recall $\varphi_i(D_i) = \exp^{-w_i \times f_i(D_i)}$

<table>
<thead>
<tr>
<th>$\varphi_1$ (A,B)</th>
<th>$\varphi_2$ (B,C)</th>
<th>$\varphi_3$ (C,D)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>value</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>value</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>value</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$w_1 \times f_1(A, B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

$w_2 \times f_2(B, C)$

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

$w_3 \times f_3(C, D)$

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_1 = 1$
$f_1(A, B) = value$

$w_2 = -4.61$
$f_2(B, C) = 1$ if $B = C$
$0$ else

$w_3 = -4.61$
$f_3(C, D) = 1$ if $C \neq D$
$0$ else
Factors vs. Features + Weights

Recall $\varphi_i(D_i) = \exp^{-w_i \times f_i(D_i)}$

<table>
<thead>
<tr>
<th>$\varphi_1$ (A,B)</th>
<th>$\varphi_2$ (B,C)</th>
<th>$\varphi_3$ (C,D)</th>
<th>$\varphi_4$ (D,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>value</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

$w_1 \times f_1(A, B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
<th>B</th>
<th>C</th>
<th>value</th>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
<td>0</td>
<td>0</td>
<td>-4.61</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
<td>1</td>
<td>1</td>
<td>-4.61</td>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

$w_1 = 1$

$f_1(A, B) = \text{value}$

$w_2 \times f_2(B, C)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
<th>B</th>
<th>C</th>
<th>value</th>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-4.61</td>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-4.61</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_2 = -4.61$

$f_2(B, C) = 1$ if $B = C$

$w_3 \times f_3(C, D)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
<th>B</th>
<th>C</th>
<th>value</th>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
<td>1</td>
<td>0</td>
<td>-4.61</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_3 = -4.61$

$f_3(C, D) = 1$ if $C \neq D$

$w_3 = 0$ else
# Factors vs. Features + Weights

Recall \( \varphi_i(D_i) = e^{\text{weight}_i \times f_i(D_i)} \)

<table>
<thead>
<tr>
<th>( \varphi_1 ) (A,B)</th>
<th>( \varphi_2 ) (B,C)</th>
<th>( \varphi_3 ) (C,D)</th>
<th>( \varphi_4 ) (D,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>value</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

\( w_1 \times f_1(A, B) \)

| A | B | value | B | C | value |
|----------------|----------------|----------------|
| 0 | 0 | -3.4 | 0 | 0 | -4.61 |
| 0 | 1 | -1.61 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | -2.3 | 1 | 1 | -4.61 |

\( w_2 \times f_2(B, C) \)

| B | C | value |
|----------------|
| 0 | 0 | -4.61 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | -4.61 |

\( w_3 \times f_3(C, D) \)

| C | D | value |
|----------------|
| 0 | 0 | 0 |
| 0 | 1 | -4.61 |
| 1 | 0 | -4.61 |
| 1 | 1 | 0 |

\( w_4 \times f_4(D, A) \)

| D | A | value |
|----------------|
| 0 | 0 | -4.61 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | -4.61 |

\( w_1 = 1 \)

\( f_1(A, B) = \text{value} \)

\( w_2 = -4.61 \)

\( f_2(B, C) = 1 \text{ if } B = C \)

\( f_2(B, C) = 0 \text{ else} \)

\( w_3 = -4.61 \)

\( f_3(C, D) = 1 \text{ if } C \neq D \)

\( f_3(C, D) = 0 \text{ else} \)

\( w_4 = -4.61 \)

\( f_4(D, A) = 1 \text{ if } D = A \)

\( f_4(D, A) = 0 \text{ else} \)
Indicator Feature Function

There is a special, restricted kind of a simple feature function which is particularly important in practice, called the indicator feature function

This function is one which takes the value of 1 for particular values of the arguments $D_i$ and 0 for all other values.

We use the notation $f^{ab}(A, B)$ to denote that $f(A, B)$ returns 1 when $A = a$ and $B = b$ and 0 otherwise.

Indicator functions can be directly extracted from data during learning (later)
Example: Using Indicator Function

$$w_1 \times f_1(A, B)$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

$$w_1 = 1$$
$$f_1(A, B) = \text{value}$$

$$w_2 \times f_2(B, C)$$

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

$$w_2 = -4.61$$
$$f_2(B, C) = 1 \text{ if } B = C$$
$$0 \text{ else}$$

$$w_3 \times f_3(C, D)$$

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$$w_3 = -4.61$$
$$f_3(C, D) = 1 \text{ if } C \neq D$$
$$0 \text{ else}$$

$$w_4 \times f_4(D, A)$$

<table>
<thead>
<tr>
<th>D</th>
<th>A</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

$$w_4 = -4.61$$
$$f_4(D, A) = 1 \text{ if } D = A$$
$$0 \text{ else}$$
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]
Example: Using Indicator Function

$$w_1 \times f_1(A, B)$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

$$w_1 = 1$$
$$f_1(A, B) = \text{value}$$

$$f_1^{00}(A, B) = 1 \text{ if } A = 0, B = 0$$

$$f_1^{01}(A, B) = 1 \text{ if } A = 0, B = 1$$

$$f_1^{10}(A, B) = 1 \text{ if } A = 1, B = 0$$

$$f_1^{11}(A, B) = 1 \text{ if } A = 1, B = 1$$
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]

\[ f_{100}(A, B) = 1 \text{ if } A = 0, B = 0 \]
\[ f_{101}(A, B) = 1 \text{ if } A = 0, B = 1 \]
\[ f_{110}(A, B) = 1 \text{ if } A = 1, B = 0 \]
\[ f_{111}(A, B) = 1 \text{ if } A = 1, B = 1 \]

\[ f_1(A, B) = -3.4 \times f_{100}(A, B) + -1.61 \times f_{101}(A, B) + 1 \times f_{110}(A, B) + -2.3 \times f_{111}(A, B) \]
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \]

\[ -3.4 \times f_1^{00}(A, B) + \]
\[ -1.61 \times f_1^{01}(A, B) + \]
\[ 1 \times f_1^{10}(A, B) + \]
\[ -2.3 \times f_1^{11}(A, B) \]
Example: Using Indicator Function

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]

\[ f_1(A, B) = -3.4 \times f_1^{00}(A, B) + -1.61 \times f_1^{01}(A, B) + 1 \times f_1^{10}(A, B) + -2.3 \times f_1^{11}(A, B) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_2 = -4.61 \]
\[ f_2(B, C) = 1 \text{ if } B = C \]
\[ 0 \text{ else} \]
### Example: Using Indicator Function

**$w_1 \times f_1(A, B)$**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

- $w_1 = 1$
- $f_1(A, B) = \text{value}$

**$w_2 \times f_2(B, C)$**

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

- $w_2 = -4.61$
- $f_2(B, C) = 1 \text{ if } B = C$
- $0 \text{ else}$

**$f_1(A, B) =$**

- $-3.4 \times f_1^{00}(A, B) +$  
- $-1.61 \times f_1^{01}(A, B) +$  
- $1 \times f_1^{10}(A, B) +$  
- $-2.3 \times f_1^{11}(A, B)$

**$f_2(B, C) =$**

- $f_2^{00}(B, C) = 1 \text{ if } B = 0, C = 0$
- $f_2^{01}(B, C) = 1 \text{ if } B = 0, C = 1$
- $f_2^{10}(B, C) = 1 \text{ if } B = 1, C = 0$
- $f_2^{11}(B, C) = 1 \text{ if } B = 1, C = 1$
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]

\[ f_1(A, B) = -3.4 \times f_1^{00}(A, B) + -1.61 \times f_1^{01}(A, B) + 1 \times f_1^{10}(A, B) + -2.3 \times f_1^{11}(A, B) \]

\[ w_2 \times f_2(B, C) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_2 = -4.61 \]
\[ f_2(B, C) = 1 \text{ if } B = C \]
\[ 0 \text{ else} \]

\[ f_2^{00}(B, C) = 1 \text{ if } B = 0, C = 0 \]
\[ f_2^{01}(B, C) = 1 \text{ if } B = 0, C = 1 \]
\[ f_2^{10}(B, C) = 1 \text{ if } B = 1, C = 0 \]
\[ f_2^{11}(B, C) = 1 \text{ if } B = 1, C = 1 \]
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]

\[ f_1(A, B) =
\]
\[-3.4 \times f_1^{00}(A, B) +
\-1.61 \times f_1^{01}(A, B) +
\1 \times f_1^{10}(A, B) +
\-2.3 \times f_1^{11}(A, B) \]

\[ w_2 \times f_2(B, C) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_2 = -4.61 \]
\[ f_2(B, C) = 1 \text{ if } B = C \]
\[ 0 \text{ else} \]

\[ f_2(B, C) =
\]
\[-4.61 \times f_2^{00}(B, C) + f_2^{11}(B, C) \]
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]

\[ f_1(A, B) = -3.4 \times f_1^{00}(A, B) + -1.61 \times f_1^{01}(A, B) + 1 \times f_1^{10}(A, B) + -2.3 \times f_1^{11}(A, B) \]

\[ w_2 \times f_2(B, C) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_2 = -4.61 \]
\[ f_2(B, C) = 1 \text{ if } B = C \]
\[ 0 \text{ else} \]

\[ f_2(B, C) = f_2^{00}(B, C) + f_2^{11}(B, C) \]

Rest are Similar
### Example: Using Indicator Function

#### $w_1 \times f_1(A, B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

$w_1 = 1$

$f_1(A, B) =$

\[-3.4 \times f_1^{00}(A, B) + \]
\[-1.61 \times f_1^{01}(A, B) + \]
\[1 \times f_1^{10}(A, B) + \]
\[-2.3 \times f_1^{11}(A, B)\]

#### $w_2 \times f_2(B, C)$

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

$w_2 = -4.61$

$f_2(B, C) = 1$ if $B = C$

0 else

#### $w_3 \times f_3(C, D)$

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_3 = -4.61$

$f_3(C, D) = 1$ if $C \neq D$

0 else

#### $w_4 \times f_4(D, A)$

<table>
<thead>
<tr>
<th>D</th>
<th>A</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

$w_4 = -4.61$

$f_4(D, A) = 1$ if $D = A$

0 else

---

$f_1(A, B) =$

\[-3.4 \times f_1^{00}(A, B) + \]
\[-1.61 \times f_1^{01}(A, B) + \]
\[1 \times f_1^{10}(A, B) + \]
\[-2.3 \times f_1^{11}(A, B)\]

$f_2(B, C) =$

\[f_2^{00}(B, C) + f_2^{11}(B, C)\]

$f_3^{00}(C, D) = 1$ if $C = 0$, $D = 0$

$f_3^{01}(C, D) = 1$ if $C = 0$, $D = 1$

$f_3^{10}(C, D) = 1$ if $C = 1$, $D = 0$

$f_3^{11}(C, D) = 1$ if $C = 1$, $D = 1$

$f_4^{00}(D, A) = 1$ if $D = 0$, $A = 0$

$f_4^{01}(C, D) = 1$ if $D = 0$, $A = 1$

$f_4^{10}(C, D) = 1$ if $D = 1$, $A = 0$

$f_4^{11}(C, D) = 1$ if $D = 1$, $A = 1$
**Example: Using Indicator Function**

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>value</th>
<th>$w_1 \times f_1(A, B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
<td></td>
</tr>
</tbody>
</table>

$w_1 = 1$

$f_1(A, B) = -3.4 \times f_{100}(A, B) + -1.61 \times f_{101}(A, B) + 1 \times f_{110}(A, B) + -2.3 \times f_{111}(A, B)$

<table>
<thead>
<tr>
<th>$B$</th>
<th>$C$</th>
<th>value</th>
<th>$w_2 \times f_2(B, C)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
<td></td>
</tr>
</tbody>
</table>

$w_2 = -4.61$

$f_2(B, C) = 1$ if $B = C$

$0$ else

<table>
<thead>
<tr>
<th>$C$</th>
<th>$D$</th>
<th>value</th>
<th>$w_3 \times f_3(C, D)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

$w_3 = -4.61$

$f_3(C, D) = 1$ if $C \neq D$

$0$ else

<table>
<thead>
<tr>
<th>$D$</th>
<th>$A$</th>
<th>value</th>
<th>$w_4 \times f_4(D, A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
<td></td>
</tr>
</tbody>
</table>

$w_4 = -4.61$

$f_4(D, A) = 1$ if $D = A$

$0$ else

$f_1(A, B) = f_{100}(B, C) + f_{111}(B, C)$

$f_2(B, C) = f_{200}(B, C) + f_{211}(B, C)$

$f_3(C, D) = f_{301}(C, D) + f_{310}(C, D)$

$f_4(D, A) = f_{400}(D, A) + f_{411}(D, A)$
Example: Using Indicator Function

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]
\[ f_1(A, B) = \text{value} \]

\[ f_1(A, B) =
-3.4 \times f_{100}(A, B) +
-1.61 \times f_{101}(A, B) +
1 \times f_{110}(A, B) +
-2.3 \times f_{111}(A, B) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_2 = -4.61 \]
\[ f_2(B, C) = 1 \text{ if } B = C \]
\[ 0 \text{ else} \]

\[ f_2(B, C) =
-4.61 \times f_{200}(B, C) + f_{211}(B, C) \]

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ w_3 = -4.61 \]
\[ f_3(C, D) = 1 \text{ if } C \neq D \]
\[ 0 \text{ else} \]

\[ f_3(C, D) =
f_{301}(C, D) + f_{310}(C, D) \]

<table>
<thead>
<tr>
<th>D</th>
<th>A</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_4 = -4.61 \]
\[ f_4(D, A) = 1 \text{ if } D = A \]
\[ 0 \text{ else} \]

\[ f_4(D, A) =
f_{400}(D, A) + f_{411}(D, A) \]

\[
P(X_1, \ldots, X_n) = \frac{1}{Z(X_1, \ldots, X_n)} \exp^{-\sum_{i=1}^{k} (w_i \times f_i(D_i))}
\]
Example: Using Indicator Function

\[ w_1 \times f_1(A, B) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

\[ w_1 = 1 \]

\[ f_1(A, B) = \]

\[-3.4 \times f_1^{00}(A, B) + -1.61 \times f_1^{01}(A, B) + 1 \times f_1^{10}(A, B) + -2.3 \times f_1^{11}(A, B) \]

\[ w_2 \times f_2(B, C) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_2 = -4.61 \]

\[ f_2(B, C) = 1 \text{ if } B = C \]

\[ f_2(B, C) = f_2^{00}(B, C) + f_2^{11}(B, C) \]

\[ w_3 \times f_3(C, D) \]

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ w_3 = -4.61 \]

\[ f_3(C, D) = 1 \text{ if } C \neq D \]

\[ f_3(C, D) = f_3^{01}(C, D) + f_3^{10}(C, D) \]

\[ w_4 \times f_4(D, A) \]

<table>
<thead>
<tr>
<th>D</th>
<th>A</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-4.61</td>
</tr>
</tbody>
</table>

\[ w_4 = -4.61 \]

\[ f_4(D, A) = 1 \text{ if } D = A \]

\[ f_4(D, A) = f_4^{00}(D, A) + f_4^{11}(D, A) \]

\[ P(A, B, C, D) = \frac{\exp^{-SP}}{Z(A, B, C, D)} \]

\[ SP = -3.4 \times f_1^{00}(A, B) + -1.61 \times f_1^{01}(A, B) + 1 \times f_1^{10}(A, B) + -2.3 \times f_1^{11}(A, B) + -4.61 \times f_2^{00}(B, C) + -4.61 \times f_2^{11}(B, C) + -4.61 \times f_3^{01}(C, D) + -4.61 \times f_3^{10}(C, D) + -4.61 \times f_4^{00}(D, A) + -4.61 \times f_4^{11}(D, A) \]
Graphical Models: So far

In the last segment we learned about log-linear models and indicator functions.

Log-linear models are important because they allow to capture factors more concisely, the indicator functions can be directly extracted from data, and the weights can be learned (to fit the optimization objective, discussed later).

So far we have been discussing the joint probability distribution $P(X_1, ..., X_n)$. We next focus on conditional distributions.
Conditional Random Field

[J. Lafferty, A. McCallum, F. Pereira, ICML 2001]

Let \( X \cup Y \) be a set of random variables where \( X = \{X_1, ..., X_n\} \), \( Y = \{Y_1, ..., Y_n\} \). Here, \( X \) are the observed variables and \( Y \) are the target variables. Let \( D_1, D_2, ..., D_m \subseteq X \cup Y \) where \( D_i \not\subseteq X \)

A Conditional Random Field is defined as:

\[
P(Y_1, ..., Y_n \mid X_1, ..., X_n) = \frac{\varphi_1 (D_1) \times \varphi_2 (D_2) \times ... \times \varphi_m (D_m)}{Z(X_1, ..., X_n)}
\]

\[
Z(X_1, ..., X_n) = \sum_Y \varphi_1 (D_1) \times \varphi_2 (D_2) \times ... \times \varphi_m (D_m)
\]

Note that the \( Z \) function ranges over variables in \( X \) only.
Conditional Random Field

[J. Lafferty, A. McCallum, F. Pereira, ICML 2001]

**Key advantage:** avoids encoding distribution over the variables X. Thus, we can use many observed variables with complex dependencies without needing to model any joint distributions over them.

\[
P(Y_1, \ldots, Y_n \mid X_1, \ldots, X_n) = \frac{\varphi_1(D_1) \times \varphi_2(D_2) \times \ldots \times \varphi_m(D_m)}{Z(X_1, \ldots, X_n)}
\]

\[
Z(X_1, \ldots, X_n) = \sum_{Y} \varphi_1(D_1) \times \varphi_2(D_2) \times \ldots \times \varphi_m(D_m)
\]
Conditional Random Field: Example
Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location
Conditional Random Field: Example
Text Analytics

Problem Statement: Given a sentence, label each word with whether it is a person or a location.

5 Labels: B-per, I-per, B-loc, I-loc, N
Conditional Random Field: Example
Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location

5 Labels: B-per, I-per, B-loc, I-loc, N

Given:
Mr. Smith came today to Portland
Conditional Random Field: Example
Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location

5 Labels: B-per, I-per, B-loc, I-loc, N

Predict:
B-per  I-per  N  N  N  N  B-loc

Given:
Mr. Smith came today to Portland
Conditional Random Field: Example
Text Analytics

**Problem Statement:**
Given a sentence, label each word with whether it is a person or a location

5 Labels: B-per, l-per, B-loc, l-loc, N
Conditional Random Field: Example

Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location.

5 Labels: B-per, I-per, B-loc, I-loc, N
Conditional Random Field: Example
Text Analytics

**Problem Statement:**
Given a sentence, label each word with whether it is a **person** or a **location**

5 Labels: B-per, l-per, B-loc, l-loc, N

Two kinds of factors for each word $X_t$

- Captures relationship between **neighboring** predictions
- Relationship between prediction and the neighbors of the word

\[
\varphi^1_t (Y_t, Y_{t+1}) \quad \varphi^2_t (Y_t, X_1, \ldots, X_t)
\]
Conditional Random Field: Example

Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location

5 Labels: B-per, I-per, B-loc, I-loc, N

Two kinds of factors for each word $X_t$  

$\phi_1^t (Y_t, Y_{t+1})$  $\phi_2^t (Y_t, X_1, \ldots, X_t)$

Diagram:

1. $Y_1$ to $X_1$: Mr.
2. $Y_2$ to $X_2$: Smith
3. $Y_3$ to $X_3$: came
4. $Y_4$ to $X_4$: today
5. $Y_5$ to $X_5$: to
6. $Y_6$ to $X_6$: Portland
7. $Y_7$ to $X_7$:
Conditional Random Field: Example

Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location

\[
P(Y_1, \ldots, Y_7 \mid X_1, \ldots, X_7) = \frac{\prod_{i=1}^{7} \varphi(Y_i, Y_{i+1}) \times \varphi(Y_6, Y_7) \times \prod_{i=1}^{7} \varphi(X_i, X_{i+1}) \times \varphi(Y_6, X_6, X_7)}{\sum \prod_{i=1}^{7} \varphi(Y_i, Y_{i+1}) \times \varphi(Y_6, Y_7) \times \prod_{i=1}^{7} \varphi(X_i, X_{i+1}) \times \varphi(Y_6, X_6, X_7)} \]

\(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7\)

Diagram:
- \(Y_1\) to \(X_1\): Mr.
- \(Y_2\) to \(X_2\): Smith
- \(Y_3\) to \(X_3\): came
- \(Y_4\) to \(X_4\): today
- \(Y_5\) to \(X_5\): to
- \(Y_6\) to \(X_6\): Portland
- \(Y_7\) to \(X_7\):
Conditional Random Field: Example

Text Analytics

Problem Statement:
Given a sentence, label each word with whether it is a person or a location

\[
P(Y_1, \ldots, Y_7 \mid X_1, \ldots, X_7) =
\]

\[
\sum \varphi^1_1 (Y_1, Y_2) \times \ldots \times \varphi^7_6 (Y_6, Y_7) \times \varphi^2_1 (Y_1, X_1, X_2) \times \ldots \times \varphi^2_7 (Y_7, X_6, X_7)
\]

\[
\varphi^1_1 (Y_1, Y_2) \times \ldots \times \varphi^7_6 (Y_6, Y_7) \times \varphi^2_1 (Y_1, X_1, X_2) \times \ldots \times \varphi^2_7 (Y_7, X_6, X_7)
\]

\[
Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7
\]

---

B-per  I-per  N  N  N  B-loc

X_1  X_2  X_3  X_4  X_6  X_7

Mr.  Smith  came  today  to  Portland
Conditional Random Field: Log-Linear Form

\[ P(y | x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]
CRF: Example

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Example: Let \( y = i, r \) and \( x = t \)
CRF: Example

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Example: Let \( y = \text{first, last} \) and \( x = \text{town} \)

<table>
<thead>
<tr>
<th>first</th>
<th>town</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pencho</td>
<td>Portland</td>
<td>0.1</td>
</tr>
<tr>
<td>James</td>
<td>Portland</td>
<td>0.3</td>
</tr>
</tbody>
</table>

\[ P(y \mid x) = \exp(w^T f(y, x)) \]

<table>
<thead>
<tr>
<th>first</th>
<th>last</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pencho</td>
<td>Smith</td>
<td>0.7</td>
</tr>
<tr>
<td>James</td>
<td>Chandra</td>
<td>0.4</td>
</tr>
</tbody>
</table>
CRF: Example

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Example: Let \( y = \text{first, last} \) and \( x = \text{town} \)

<table>
<thead>
<tr>
<th>first</th>
<th>town</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pencho</td>
<td>Portland</td>
<td>0.1</td>
</tr>
<tr>
<td>James</td>
<td>Portland</td>
<td>0.3</td>
</tr>
</tbody>
</table>

\[
P(\text{first, last} \mid \text{town}) = \frac{\exp (0.1 \cdot f_1 + 0.3 \cdot f_2 + 0.7 \cdot f_3 + 0.4 \cdot f_4)}{Z(\text{town})}
\]
Tutorial Outline

• Motivation
  • Potential applications

• Statistical language models
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• Graphical Models
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
Queries: MAP vs. Max Marginals

MAP Inference: \((y_1, ..., y_n)^{\text{MAP}} = \arg\max_{y_1, ..., y_n} P(y_1, ..., y_n)\)

VS.

Max Marginals: \((y_1, ..., y_n)^{\text{ML}} = (y_1^{\text{ML}}, ..., y_n^{\text{ML}})\)

\[ y_1^{\text{ML}} = \arg\max_{y_1} P(y_1) \quad \text{......} \quad y_n^{\text{ML}} = \arg\max_{y_n} P(y_n) \]

Consider the following probability distribution:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.35</td>
</tr>
</tbody>
</table>
Queries: MAP vs. Max Marginals

**MAP Inference:**

\[
(y_1, \ldots, y_n)^{\text{MAP}} = \arg\max_{y_1, \ldots, y_n} P(y_1, \ldots, y_n)
\]

**Max Marginals:**

\[
(y_1, \ldots, y_n)^{\text{ML}} = (y_1^{\text{ML}}, \ldots, y_n^{\text{ML}})
\]

\[
y_1^{\text{ML}} = \arg\max_{y_1} P(y_1) \quad \ldots \quad y_n^{\text{ML}} = \arg\max_{y_n} P(y_n)
\]

Consider the following probability distribution:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.35</td>
</tr>
</tbody>
</table>
Factor Graph Representation

A third representation of a Markov Network, which makes the factors explicit in the graph is the factor graph. Here, factors are nodes in the graph. Edges only exist between factor nodes and variable nodes.

Factor graphs are often useful when performing inference.
Example: Compute $P(B=0 \mid G = 1, D = 0, A = 1)$

<table>
<thead>
<tr>
<th>$\varphi_0 (B)$</th>
<th>$\varphi_1 (S,B)$</th>
<th>$\varphi_2 (A,B)$</th>
<th>$\varphi_3 (G,S)$</th>
<th>$\varphi_4 (D,S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>$S$</td>
<td>$A$</td>
<td>$G$</td>
<td>$D$</td>
</tr>
<tr>
<td>$B$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.99</td>
<td>0.5</td>
<td>0.99</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>0.01</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>0.5</td>
<td>0.01</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>0.9</td>
<td></td>
<td>0.9</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$P(B, S \mid G, D, A) = \frac{\varphi_0 (B) \times \varphi_1 (S, B) \times \varphi_2 (A, B) \times \varphi_3 (G, S) \times \varphi_4 (D, S)}{Z(G, D, A)}$$

$$Z(G, D, A) = \sum_{B,S} \varphi_0 (B) \times \varphi_1 (S, B) \times \varphi_2 (A, B) \times \varphi_3 (G, S) \times \varphi_4 (D, S)$$
Example: Compute $P(B=0 \mid G = 1, D = 0, A = 1)$

\[
P(B=0 \mid G, D, A) = \sum_S P(B=0, S \mid G, D, A) = P(B, S=0 \mid G, D, A) + P(B, S=1 \mid G, D, A)
\]

\[
P(B=0 \mid G=1, D=0, A=1) = P(B=0, S = 0 \mid G=1, D=0, A=1) + P(B=0, S=1 \mid G=1, D=0, A=1)
\]
Example: Compute $P(B=0 \mid G = 1, D = 0, A = 1)$

$$Z(G=1, D=0, A=1) = \varphi_0 (0) \times \varphi_1 (0, 0) \times \varphi_2 (1, 0) \times \varphi_3 (1, 0) \times \varphi_4 (0, 0) + \varphi_0 (1) \times \varphi_1 (0, 1) \times \varphi_2 (1, 1) \times \varphi_3 (1, 0) \times \varphi_4 (0, 0) + \varphi_0 (0) \times \varphi_1 (1, 0) \times \varphi_2 (1, 0) \times \varphi_3 (1, 1) \times \varphi_4 (0, 1) + \varphi_0 (1) \times \varphi_1 (1, 1) \times \varphi_2 (1, 1) \times \varphi_3 (1, 1) \times \varphi_4 (0, 1) +$$

$$= 0.99 \times 0.5 \times 0.01 \times 0.3 \times 0.8 + 0.01 \times 0.1 \times 0.9 \times 0.3 \times 0.8 + 0.99 \times 0.5 \times 0.01 \times 0.6 \times 0.3 + 0.01 \times 0.9 \times 0.9 \times 0.6 \times 0.7$$

$$= 0.001188 + 0.000216 + 0.000891 + 0.003402 = 0.005697$$
Example: Compute $P(B=0 \mid G = 1, D = 0, A = 1)$

$P(B=0, S = 0 \mid G=1, D=0, A=1) = \varphi_0 (B) \times \varphi_1 (S, B) \times \varphi_2 (A, B) \times \varphi_3 (G, S) \times \varphi_4 (D, S)/Z = \varphi_0 (0) \times \varphi_1 (0, 0) \times \varphi_2 (1, 0) \times \varphi_3 (1, 0) \times \varphi_4 (0, 0)/Z = 0.99 \times 0.5 \times 0.01 \times 0.3 \times 0.8)/Z = 0.001188/0.005697$

$P(B=0, S = 1 \mid G=1, D=0, A=1) = \varphi_0 (B) \times \varphi_1 (S, B) \times \varphi_2 (A, B) \times \varphi_3 (G, S) \times \varphi_4 (D, S)/Z = \varphi_0 (0) \times \varphi_1 (1, 0) \times \varphi_2 (1, 0) \times \varphi_3 (1, 1) \times \varphi_4 (0, 1)/Z = 0.99 \times 0.5 \times 0.01 \times 0.6 \times 0.3)/Z = 0.000891/0.005697$

$+ 0.002079/0.005697$
Belief Propagation

An inference algorithm for computing queries on distributions represented as factor graphs.

The algorithm is exact if the graph is in the shape of a tree. If the graph contains loops, the algorithm is referred to as loopy belief propagation and in this case, there are no guarantees on exactness. Nonetheless, the algorithm is often used for loopy graphs as well.

Algorithm is based on passing two kinds of messages on the factor graph.
Belief Propagation: Message Kinds

**Variable-to-Factor**

\[ \mu_{\varphi \rightarrow \phi} : \text{Val} \rightarrow \text{MsgVal} \]

**Factor-to-Variable**

\[ \mu_{\phi \rightarrow \varphi} : \text{Val} \rightarrow (\text{MsgVal}, \rho(V \times \text{Val})) \]

\[ \begin{align*}
\mu_{x \rightarrow \phi_4(x)} &= \mu_{\phi_1 \rightarrow x(x)} \times \\
& \quad \times \mu_{\phi_2 \rightarrow x(x)} \times \\
& \quad \times \mu_{\phi_3 \rightarrow x(x)} \\
\end{align*} \]
Belief Propagation: Operation

• Assume the factor graph is a tree.

• Pick an arbitrary node to be the root in the tree (i.e., a factor or a variable).

• Initialize as follows:
  if $\phi$ is a leaf node (whose parent is variable $v$) then $\mu_{\phi \rightarrow v}(v) = \phi(v)$

  if $v$ is a leaf node (whose parent is a factor $\phi$) then $\mu_{v \rightarrow \phi}(v) = 1$

• Proceed by sending messages around. A node can send out a message if all incoming messages have arrived

• Once a node $X$ receives all messages, we can compute the marginal:

\[ P(X) = \prod \mu_{\phi \rightarrow v}(v) \quad (\phi \text{ is a neighbor of } X) \]
Belief Propagation: Compute $P(B=0 \mid G = 1, D = 0, A = 1)$

$\varphi_0(B)$

<table>
<thead>
<tr>
<th>B</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.99</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
</tr>
</tbody>
</table>

$\varphi_1(S,B)$

<table>
<thead>
<tr>
<th>S</th>
<th>B</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

$\varphi_2(A,B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.99</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

$\varphi_3(G,S)$

<table>
<thead>
<tr>
<th>G</th>
<th>S</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
</tr>
</tbody>
</table>

$\varphi_4(D,S)$

<table>
<thead>
<tr>
<th>D</th>
<th>S</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.7</td>
</tr>
</tbody>
</table>
Belief Propagation: Compute \( P(B=0 \mid G = 1, D = 0, A = 1) \)

\[ \begin{align*}
\varphi_0 (B) & \quad \varphi_1 (S,B) & \quad \varphi_2 (A,B) & \quad \varphi_3 (G,S) & \quad \varphi_4 (D,S) \\
\begin{array}{c|c}
B & \text{val} \\
\hline
0 & 0.99 \\
1 & 0.01 \\
\end{array} & \begin{array}{c|c|c}
S & B & \text{val} \\
\hline
0 & 0 & 0.5 \\
0 & 1 & 0.1 \\
1 & 0 & 0.5 \\
1 & 1 & 0.9 \\
\end{array} & \begin{array}{c|c|c}
A & B & \text{val} \\
\hline
0 & 0 & 0.99 \\
0 & 1 & 0.1 \\
1 & 0 & 0.01 \\
1 & 1 & 0.9 \\
\end{array} & \begin{array}{c|c|c}
G & S & \text{val} \\
\hline
0 & 0 & 0.7 \\
0 & 1 & 0.4 \\
1 & 0 & 0.3 \\
1 & 1 & 0.6 \\
\end{array} & \begin{array}{c|c|c}
D & S & \text{val} \\
\hline
0 & 0 & 0.8 \\
0 & 1 & 0.3 \\
1 & 0 & 0.2 \\
1 & 1 & 0.7 \\
\end{array}
\]
Belief Propagation: Compute \( P(B=0 \mid G = 1, D = 0, A = 1) \)

**Factor-to-Variable Messages:**

\[
\begin{align*}
\mu_{\varphi_3} & \rightarrow S(S = 0) = \sum_G \varphi_3(G, S = 0) \times \mu_G \rightarrow \varphi_3(G) \quad & \mu_{\varphi_3} & \rightarrow S(S = 1) = \sum_G \varphi_3(G, S = 1) \times \mu_G \rightarrow \varphi_3(G) \\
\mu_{\varphi_4} & \rightarrow S(S = 0) = \sum_D \varphi_4(D, S = 0) \times \mu_D \rightarrow \varphi_4(D) \quad & \mu_{\varphi_4} & \rightarrow S(S = 1) = \sum_D \varphi_4(D, S = 1) \times \mu_D \rightarrow \varphi_4(D) \\
\mu_{\varphi_2} & \rightarrow B(B = 0) = \sum_A \varphi_2(A, B = 0) \times \mu_A \rightarrow \varphi_2(A) \quad & \mu_{\varphi_2} & \rightarrow B(B = 1) = \sum_A \varphi_2(A, B = 1) \times \mu_A \rightarrow \varphi_2(A) \\
\mu_{\varphi_1} & \rightarrow B(B = 0) = \sum_S \varphi_1(S, B = 0) \times \mu_S \rightarrow \varphi_1(S) \quad & \mu_{\varphi_1} & \rightarrow B(B = 1) = \sum_S \varphi_1(S, B = 1) \times \mu_S \rightarrow \varphi_1(S) \\
\mu_{\varphi_0} & \rightarrow B(B = 0) = \varphi_0(B = 0) \quad & \mu_{\varphi_0} & \rightarrow B(B = 1) = \varphi_0(B = 1)
\end{align*}
\]

**Variable-to-Factor Messages:**

\[
\begin{align*}
\mu_G & \rightarrow \varphi_3(G = 0) = 0 \quad & \mu_D & \rightarrow \varphi_4(D = 0) = 1 \quad & \mu_A & \rightarrow \varphi_2(A = 0) = 0 \\
\mu_G & \rightarrow \varphi_3(G = 1) = 1 \quad & \mu_D & \rightarrow \varphi_4(D = 1) = 0 \quad & \mu_A & \rightarrow \varphi_2(A = 1) = 1 \\
\mu_S & \rightarrow \varphi_1(S = 0) = \mu_{\varphi_3} \rightarrow S(S = 0) \times \mu_{\varphi_4} \rightarrow S(S = 0) \\
\mu_S & \rightarrow \varphi_1(S = 1) = \mu_{\varphi_3} \rightarrow S(S = 1) \times \mu_{\varphi_4} \rightarrow S(S = 1)
\end{align*}
\]
Belief Propagation: Compute \( P(B=0 \mid G = 1, D = 0, A = 1) \)

\[
P(B = 0 \mid G=1, D=0, A=1) = \mu_{\varphi_0 \rightarrow B}(B = 0) \times \mu_{\varphi_1 \rightarrow B}(B = 0) \times \mu_{\varphi_2 \rightarrow B}(B = 0) \\
= \varphi_0(B = 0) \times \left( \sum_S \varphi_1(S, B = 0) \times \mu_{S \rightarrow \varphi_1(S)} \right) \times \left( \sum_A \varphi_2(A, B = 0) \times \mu_{A \rightarrow \varphi_2(A)} \right) \\
= \varphi_0(B = 0) \times \left( \sum_S \varphi_1(S, B = 0) \times \left( \mu_{\varphi_3 \rightarrow S}(S) \times \mu_{\varphi_4 \rightarrow S}(S) \right) \right) \times \left( \sum_A \varphi_2(A, B = 0) \times \left( \mu_{A \rightarrow \varphi_2(A)} \right) \right) \\
= \varphi_0(B = 0) \times \left( \sum_S \varphi_1(S, B = 0) \times \left( \sum_G \varphi_3(G, S) \times \mu_{G \rightarrow \varphi_3(G)} \right) \times \left( \sum_D \varphi_4(D, S) \times \mu_{D \rightarrow \varphi_4(D)} \right) \right) \\
\times \left( \sum_A \varphi_2(A, B = 0) \times \left( \mu_{A \rightarrow \varphi_2(A)} \right) \right)
\]
Belief Propagation: Compute \( P(B=0 \mid G = 1, D = 0, A = 1) \)

\[
P(B = 0 \mid G=1, D=0, A=1) =
\begin{align*}
&= 0.99 \times \\
&(\varphi_1(S=0, B = 0) \times \\
&(\varphi_3(G=0, S=0) \times \mu_{G \to \varphi_3(G=0)} + \varphi_3(G=1, S=0) \times \mu_{G \to \varphi_3(G=1)}) \times \\
&(\varphi_4(D=0, S=0) \times \mu_{D \to \varphi_4(D=0)} + \varphi_4(D=1, S=0) \times \mu_{D \to \varphi_4(D=1)})) + \\
&(\varphi_1(S=1, B = 0) \times \\
&(\varphi_3(G=0, S=1) \times \mu_{G \to \varphi_3(G=0)} + \varphi_3(G=1, S=1) \times \mu_{G \to \varphi_3(G=1)}) \times \\
&(\varphi_4(D=0, S=1) \times \mu_{D \to \varphi_4(D=0)} + \varphi_4(D=1, S=1) \times \mu_{D \to \varphi_4(D=1)})) \times \\
&(\varphi_2(A=0, B = 0) \times (\mu_{A \to \varphi_2(A=0)} + \varphi_2(A=1, B = 0) \times (\mu_{A \to \varphi_2(A=1))))
\end{align*}
\]

\[
= 0.99 \times ((0.5 \times (0.7 \times 0 + 0.3 \times 1) \times (0.8 \times 1 + 0.2 \times 0) + \\
(0.5 \times (0.4 \times 0 + 0.6 \times 1) \times (0.3 \times 1 + 0.7 \times 0)) \times \\
(0.99 \times 0 + 0.01 \times 1) = 0.99 \times (0.5 \times 0.3 \times 0.8 + 0.5 \times 0.6 \times 0.3) \times 0.01 = 0.002079
\]
Querying the model: MAP Inference

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given \( x \), we would like to predict \( y = y_1, y_2, \ldots, y_n \) that maximizes \( P(y \mid x) \).

This requires us to make a joint prediction, together for all \( y_1, y_2, \ldots, y_n \).
Querying the model: MAP Inference

\[ P(y | x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given \( x \), we would like to predict \( y = y_1, y_2, \ldots, y_n \) that maximizes \( P(y | x) \).

This requires us to make a joint prediction, together for all \( y_1, y_2, \ldots, y_n \).

\[ y^{\text{best}} = \arg\max_{y \in \Omega_x} P(y | x) = \arg\max_{y \in \Omega_x} w^T f(y, x) \]
Querying the model: MAP Inference

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given \( x \), we would like to predict \( y = y_1, y_2 \ldots y_n \) that maximizes \( P(y \mid x) \)

This requires us to make a joint prediction, together for all \( y_1, y_2 \ldots y_n \)

\[ y^{\text{best}} = \arg\max_{y \in \Omega_x} P(y \mid x) = \arg\max_{y \in \Omega_x} w^T f(y, x) \]

We designed an approximate one to fit our needs, i.e., deal with many values
Querying the model: Max-marginals

\[ y_{1}^{\text{best}} = \arg\max P(y_{1} | x) \quad \ldots \quad y_{n}^{\text{best}} = \arg\max P(y_{n} | x) \]

\[ y^{\text{best}} = (y_{1}^{\text{best}}, \ldots, y_{n}^{\text{best}}) \]

\[ \Sigma \Pi \] belief propagation algorithms answer max-marginal queries
Querying the model: Max-marginals

\[ y_1^{\text{best}} = \arg\max P(y_1 | x) \]

\[ y^{\text{best}} = (y_1^{\text{best}}, \ldots, y_n^{\text{best}}) \]

\[ \Sigma \prod \text{bet} \]

Proper propagation algorithms answer max-marginal queries.
function chunkData(e, t)
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;
function chunkData(e, t)
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;
function chunkData(e, t)
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;
function chunkData(e, t)
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;
function chunkData(e, t)
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;

argmax \mathbf{w}^T f(i, t, r, \text{length})

<table>
<thead>
<tr>
<th>i</th>
<th>t</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>step</td>
<td>0.5</td>
</tr>
<tr>
<td>j</td>
<td>step</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>r</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>len</td>
<td>0.6</td>
</tr>
<tr>
<td>i</td>
<td>length</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>r</th>
<th>length</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>length</td>
<td>0.5</td>
</tr>
<tr>
<td>len</td>
<td>length</td>
<td>0.3</td>
</tr>
</tbody>
</table>
function chunkData(e, t)
    var n = [];
    var r = e.length;
    var i = 0;
    for (; i < r; i += t)
        if (i + t < r)
            n.push(e.substring(i, i + t));
        else
            n.push(e.substring(i, r));
    return n;

function chunkData(str, step)
    var colNames = [];
    var len = str.length;
    var i = 0;
    for (; i < len; i += step)
        if (i + step < len)
            colNames.push(str.substring(i, i + step));
        else
            colNames.push(str.substring(i, len));
    return colNames;
Tutorial Outline

• Motivation
  • Potential applications

• Statistical language models
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• Graphical Models
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
Structured SVM Training
(N. Ratliff, J. Bagnell, M. Zinkevich, AISTATS 2007)

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given a data set: \( D = \{ x^i, y^i \}_{j=1..n} \) learn weights \( w^r \)
Structured SVM Training
(N. Ratliff, J. Bagnell, M. Zinkevich, AISTATS 2007)

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given a data set: \( D = \{ x^i, y^i \}_{j=1..n} \) learn weights \( w^T \)

Optimization objective (max-margin training):
\[ \forall j \forall y \sum w_i f_i(x^{(j)}, y^{(j)}) \geq \sum w_i f_i(x^{(j)}, y) + \Delta(y, y^{(j)}) \]
Structured SVM Training
(N. Ratliff, J. Bagnell, M. Zinkevich, AISTATS 2007)

\[ P(y | x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given a data set: \( D = \{ x^i, y^i \}_{j=1..n} \) learn weights \( w^T \)

**Optimization objective (max-margin training):**

\[ \forall j \forall y \sum w_i f_i(x^{(j)}, y^{(j)}) \geq \sum w_i f_i(x^{(j)}, y) + \Delta(y, y^{(j)}) \]

for all samples

Given prediction is better than any other prediction by a margin
Structured SVM Training
(N. Ratliff, J. Bagnell, M. Zinkevich, AISTATS 2007)

\[ P(y \mid x) = \frac{1}{Z(x)} \exp(w^T f(y, x)) \]

Given a data set: \( D = \{ x^i, y^i \}_{j=1..n} \) learn weights \( w^T \)

Optimization objective (max-margin training):

\[ \forall j \forall y \sum w_i f_i(x^{(j)}, y^{(j)}) \geq \sum w_i f_i(x^{(j)}, y) + \Delta(y,y^{(j)}) \]

for all samples

Given prediction is better than any other prediction by a margin

Avoids expensive computation of the partition function \( Z(x) \)
Tutorial Outline

• Motivation
  • Potential applications

• Statistical Language Models
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• Graphical Models
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
Tutorial Outline

• Motivation
  • Potential applications

• Statistical language models
  • N-gram and Recurrent Networks, Smoothing
  • Application: code completion

• Graphical Models
  • Markov Networks, Conditional Random Fields
  • Inference in Markov Networks
  • Learning in Markov Networks
  • Application: predicting names and types

• Hands-on session with Nice2Predict
Structured Prediction for Programs

(V. Raychev, M. V., A. Krause, ACM POPL’15)

First connection between Programs and Conditional Random Fields
Structured Prediction for Programs
(V. Raychev, M. V., A. Krause, ACM POPL’15)

var n = [];
var r = e.length;
var i = 0;
for (; i < r; i += t)
  if (i + t < r)
    n.push(e.subs(i, i + t));
  else
    n.push(e.subs(i, r));
return n;

var colNames = [];
var len = str.length;
var i = 0;
for (; i < len; i += step)
  if (i + step < len)
    colNames.push(str.subs(i, i + step));
  else
    colNames.push(str.subs(i, len));
return colNames;

Time: milliseconds
Prediction Phase
program analysis
MAP inference
transform

Learning Phase
program analysis
SSVM learning
max-margin training

30 nodes, 400 edges

Conditional Random Field
\( P(y \mid x) \)

150MB

Names: 63%
Types: 81%
(helps typechecking)

alias, call analysis
7M feature functions for names
70K feature functions for types
# Machine Learning for Programming

<table>
<thead>
<tr>
<th>Applications</th>
<th>Code completion</th>
<th>Program synthesis</th>
<th>Feedback generation</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deobfuscation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intermediate Representation</td>
<td>Sequences (sentences)</td>
<td>Translation Table</td>
<td>Graphical Models (CRFs)</td>
<td>Feature Vectors</td>
</tr>
<tr>
<td></td>
<td>Trees</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze Program (PL)</td>
<td>typestate analysis</td>
<td>control-flow analysis</td>
<td>alias analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>scope analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train Model (ML)</td>
<td>Neural Networks</td>
<td>SVM</td>
<td>Structured SVM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N-gram language model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query Model (ML)</td>
<td>$\text{argmax } P(y \mid x)$</td>
<td>$y \in \Omega$</td>
<td>Greedy MAP inference</td>
<td></td>
</tr>
</tbody>
</table>