A Statistical Learning Method for Logic Programs with Distribution Semantics

Taisuke Sato, 1995

Presented by Michael Litvak, Technion
Outline for section 1

1. Introduction
2. Distribution semantics
   - Preliminaries
   - The existence of $P_F$
   - From $P_F$ to $P_{DB}$
   - Properties of $P_{DB}$
3. EM learning
4. BS-programs
   - BS-programs
   - A learning algorithm for BS-programs
5. A learning experiment
6. Conclusion
Introduction

Figure: Hidden Markov Model of Weather [?]
Overview

1. Introduction
2. Distribution semantics
   • Preliminaries
   • The existence of $P_F$
   • From $P_F$ to $P_{DB}$
   • Properties of $P_{DB}$
3. EM learning
4. BS-programs
   • BS-programs
   • A learning algorithm for BS-programs
5. A learning experiment
6. Conclusion
Outline for section 2

1. Introduction

2. Distribution semantics
   - Preliminaries
   - The existence of $P_F$
   - From $P_F$ to $P_{DB}$
   - Properties of $P_{DB}$

3. EM learning

4. BS-programs
   - BS-programs
   - A learning algorithm for BS-programs

5. A learning experiment

6. Conclusion
Logic programming

Let $DB$ be a definite clause program in a first order language

$$DB = F \cup R$$

$$F = \{ A_1, A_2, \ldots \}$$

$$R = \{ B_1 \leftarrow W_1, B_2 \leftarrow W_2, \ldots \}$$

$$\text{head}(R) = \{ B_1, B_2, \ldots \}$$

$A_i, B_i$ - atomic formulas (atoms)

$F$ - unit clauses (facts)

$$\text{edge}(a, b)$$

$R$ - non-unit clauses (rules)

$$\text{path}(X, Y) \leftarrow \text{edge}(X, Y)$$
Logic program example

\[
DB = F \cup R
\]

\[
F = \{edge(a, b), edge(b, c)\}
\]

\[
R = \{path(X, Y) \leftarrow edge(X, Y), \quad path(X, Y) \leftarrow path(X, Z) \land edge(Z, Y)\}\n\]

There is a unique minimal model

\[
M^*_DB = \{edge(a, b), edge(b, c), \quad path(a, b), path(b, c), path(a, c)\}\n\]
Distribution semantics

- Distribution semantics = logic program + distribution on facts
- Each atom $A_i \in F$ is a random variable with possible values $\{0, 1\}$
  - 1 if $A_i$ is true
  - 0 if $A_i$ is false
Probabilistic logic program example

Assign each edge the probability it exists (independently).

![Diagram showing edges a to b with probability 0.6, b to c with probability 0.3]
Probabilistic logic program example

Assign each edge the probability it exists (independently).

![Graph](image)

Table: $M_{DB}$

<table>
<thead>
<tr>
<th>$\omega = \langle x_1, x_2 \rangle$</th>
<th>$F_1\omega$</th>
<th>$M_{DB}(\omega)$</th>
<th>$P_{DB}(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle 0, 0 \rangle$</td>
<td>${}$</td>
<td>${}$</td>
<td>0.28</td>
</tr>
<tr>
<td>$\langle 1, 0 \rangle$</td>
<td>${e(a, b)}$</td>
<td>${e(a, b), p(a, b)}$</td>
<td>0.42</td>
</tr>
<tr>
<td>$\langle 0, 1 \rangle$</td>
<td>${e(b, c)}$</td>
<td>${e(b, c), p(b, c)}$</td>
<td>0.12</td>
</tr>
<tr>
<td>$\langle 1, 1 \rangle$</td>
<td>${e(a, b), e(b, c)}$</td>
<td>$M^*_DB$</td>
<td>0.18</td>
</tr>
</tbody>
</table>
Probabilistic logic program example

Assign each edge the probability it exists (independently).

![Graph with nodes a, b, c and probabilities 0.6 and 0.3]

**Table:** $M_{DB}$

<table>
<thead>
<tr>
<th>$\omega = \langle x_1, x_2 \rangle$</th>
<th>$F_{1\omega}$</th>
<th>$M_{DB}(\omega)$</th>
<th>$P_{DB}(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle 0, 0 \rangle$</td>
<td>${ }$</td>
<td>${ }$</td>
<td>0.28</td>
</tr>
<tr>
<td>$\langle 1, 0 \rangle$</td>
<td>${ e(a, b) }$</td>
<td>${ e(a, b), p(a, b) }$</td>
<td>0.42</td>
</tr>
<tr>
<td>$\langle 0, 1 \rangle$</td>
<td>${ e(b, c) }$</td>
<td>${ e(b, c), p(b, c) }$</td>
<td>0.12</td>
</tr>
<tr>
<td>$\langle 1, 1 \rangle$</td>
<td>${ e(a, b), e(b, c) }$</td>
<td>$M_{DB}^*$</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Extend the probability space to various formulas:

$P(path(a, c)) = 0.18$

$P(\exists x \ path(a, x)) = 0.42 + 0.18 = 0.6$
Preliminaries

We make the following assumptions

- DB is ground (no free variables)
- DB is denumerably infinite
- Atoms in $F$ and $\text{head}(R)$ are disjoint

\[
DB = F \cup R
\]

\[
F = \{A_1, A_2, \ldots\}
\]

\[
R = \{B_1 \leftarrow W_1, B_2 \leftarrow W_2, \ldots\}
\]

\[
\text{head}(R) = \{B_1, B_2, \ldots\}
\]
Fix an enumeration of atoms in $F$

$$F = \{ A_1, A_2, \ldots \}$$

- an interpretation $\omega$ for $F$ - an assignment of truth values to atoms in $F$. $x_i \in \{0, 1\}$ is the value of $A_i$

$$\omega = \langle x_1, x_2, \ldots \rangle$$

- The set of all possible interpretations for $F$

$$\Omega_F = \prod_{i=1}^{\infty} \{0, 1\}_i$$

- $P_F$ - a probability measure on $\Omega_F$
Each sample \( \omega \in \Omega_F \) determines a set \( F_\omega \subset F \) of true ground atoms. We can speak of a logic program \( F_\omega \cup R \) and its least model \( M_{DB}(\omega) \). \( M_{DB}(\omega) \) decides all truth values of atoms in \( DB \).
The existence of \( P_F \)

How to construct a basic distribution \( P_F \) for \( F \)?
The existence of $P_F$

How to construct a basic distribution $P_F$ for $F$?

Define a series of finite distributions $P_F^{(n)}(A_1 = x_1, \ldots, A_n = x_n)$ such that

\[
0 \leq P_F^{(n)}(A_1 = x_1, \ldots, A_n = x_n) \leq 1
\]
\[
\sum_{x_1, \ldots, x_n} P_F^{(n)}(A_1 = x_1, \ldots, A_n = x_n) = 1
\]
\[
\sum_{x_{n+1}} P_F^{(n+1)}(A_1 = x_1, \ldots, A_{n+1} = x_{n+1}) = P_F^{(n)}(A_1 = x_1, \ldots, A_n = x_n)
\]

Theorem

There exists a completely additive probability measure $P_F$ over $\Omega_F$ satisfying for any $n$

\[
P_F(A_1 = x_1, \ldots, A_n = x_n) = P_F^{(n)}(A_1 = x_1, \ldots, A_n = x_n).
\]
Let $A_1, A_2, \ldots$ be an enumeration of all atoms appearing in $DB$. 
$\Omega_{DB} = \prod_{i=1}^{\infty} \{0, 1\};$ The set of all possible interpretations for atoms in $DB$.

**Goal**

Extend $P_F$ to a completely additive probability measure $P_{DB}$ over $\Omega_{DB}$. 
From $P_F$ to $P_{DB}$

Definitions:

$A^x = A$ if $x = 1$

$A^x = \neg A$ if $x = 0$

$[A_1^{x_1} \land \cdots \land A_n^{x_n}]_F \overset{\text{def}}{=} \{\omega \in \Omega_F \mid M_{DB}(\omega) \models A_1^{x_1} \land \cdots \land A_n^{x_n}\}$

Define a series of finite distributions $P_{DB}^{(n)}(A_1 = x_1, \ldots, A_n = x_n)$ by

$P_{DB}^{(n)}(A_1 = x_1, \ldots, A_n = x_n) \overset{\text{def}}{=} P_F([A_1^{x_1} \land \cdots \land A_n^{x_n}]_F)$

Similarly, it follows that there exists a completely additive measure $P_{DB}$ over $\Omega_{DB}$. 
Example

\[ DB_1 = F_1 \cup R_1 \]
\[ F_1 = \{ A_1, A_2 \} \]
\[ R_1 = \{ B_1 \leftarrow A_1, B_1 \leftarrow A_2, B_2 \leftarrow A_2 \} \]

Table: \( P_{F_1} \)

<table>
<thead>
<tr>
<th>( \omega = \langle x_1, x_2 \rangle )</th>
<th>( P_{F_1}(x_1, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle 0, 0 \rangle )</td>
<td>0.2</td>
</tr>
<tr>
<td>( \langle 1, 0 \rangle )</td>
<td>0.3</td>
</tr>
<tr>
<td>( \langle 0, 1 \rangle )</td>
<td>0.4</td>
</tr>
<tr>
<td>( \langle 1, 1 \rangle )</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Example

\[ DB_1 = F_1 \cup R_1 \]

\[ F_1 = \{ A_1, A_2 \} \]

\[ R_1 = \{ B_1 \leftarrow A_1, B_1 \leftarrow A_2, B_2 \leftarrow A_2 \} \]

Table: \( P_{F_1} \)

<table>
<thead>
<tr>
<th>( \omega = \langle x_1, x_2 \rangle )</th>
<th>( P_{F_1}(x_1, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle 0, 0 \rangle</td>
<td>0.2</td>
</tr>
<tr>
<td>\langle 1, 0 \rangle</td>
<td>0.3</td>
</tr>
<tr>
<td>\langle 0, 1 \rangle</td>
<td>0.4</td>
</tr>
<tr>
<td>\langle 1, 1 \rangle</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table: \( P_{DB_1} \)

<table>
<thead>
<tr>
<th>( \omega = \langle x_1, x_2, y_1, y_2 \rangle )</th>
<th>( P_{DB_1}(x_1, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle 0, 0, 0, 0 \rangle</td>
<td>0.2</td>
</tr>
<tr>
<td>\langle 1, 0, 1, 0 \rangle</td>
<td>0.3</td>
</tr>
<tr>
<td>\langle 0, 1, 1, 1 \rangle</td>
<td>0.4</td>
</tr>
<tr>
<td>\langle 1, 1, 1, 1 \rangle</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Example

$$DB_1 = F_1 \cup R_1$$

$$F_1 = \{A_1, A_2\}$$

$$R_1 = \{B_1 \leftarrow A_1, B_1 \leftarrow A_2, B_2 \leftarrow A_2\}$$

Table: $P_{F_1}$

<table>
<thead>
<tr>
<th>$\omega = \langle x_1, x_2 \rangle$</th>
<th>$P_{F_1}(x_1, x_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle 0, 0 \rangle$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\langle 1, 0 \rangle$</td>
<td>0.3</td>
</tr>
<tr>
<td>$\langle 0, 1 \rangle$</td>
<td>0.4</td>
</tr>
<tr>
<td>$\langle 1, 1 \rangle$</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table: $P_{DB_1}$

<table>
<thead>
<tr>
<th>$\omega = \langle x_1, x_2, y_1, y_2 \rangle$</th>
<th>$P_{DB_1}(x_1, x_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle 0, 0, 0, 0 \rangle$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\langle 1, 0, 1, 0 \rangle$</td>
<td>0.3</td>
</tr>
<tr>
<td>$\langle 0, 1, 1, 1 \rangle$</td>
<td>0.4</td>
</tr>
<tr>
<td>$\langle 1, 1, 1, 1 \rangle$</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>

$P_{DB_1}$ is a distribution over minimal models. Is this always the case?
Assigning probabilities to formulas

Let $G$ be an arbitrary sentence.

$$[G] \stackrel{\text{def}}{=} \{ \omega \in \Omega_{DB} \mid \omega \models G \}$$

The probability of $G$ is defined as $P_{DB}(\lfloor G \rfloor)$. 
Assigning probabilities to formulas

Let $G$ be an arbitrary sentence.

$$[G] \overset{\text{def}}{=} \{ \omega \in \Omega_{DB} \mid \omega \models G \}$$

The probability of $G$ is defined as $P_{DB}([G])$.

Table: $P_{DB_1}$

<table>
<thead>
<tr>
<th>$\omega = \langle x_1, x_2, y_1, y_2 \rangle$</th>
<th>$P_{DB_1}(x_1, x_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle 0, 0, 0, 0 \rangle$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\langle 1, 0, 1, 0 \rangle$</td>
<td>0.3</td>
</tr>
<tr>
<td>$\langle 0, 1, 1, 1 \rangle$</td>
<td>0.4</td>
</tr>
<tr>
<td>$\langle 1, 1, 1, 1 \rangle$</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>

$$P(B_1 \land B_2) = 0.4 + 0.1 = 0.5$$
Properties of $P_{DB}$

\[ DB = F \cup R \]

\[ F = \{ A_1, A_2, \ldots \} \]

\[ R = \{ B_1 \leftarrow W_1, B_2 \leftarrow W_2, \ldots \} \]

\[ \text{head}(R) = \{ B_1, B_2, \ldots \} \]

**Definition**

A support set for an atom $B \in \text{head}(R)$ is a finite subset $S$ of $F$ such that $S \cup R \vdash B$. A minimal support set for $B$ is a support set minimal with respect to set inclusion.
Support set examples

\[ DB_1 = F_1 \cup R_1 \]
\[ F_1 = \{A_1, A_2\} \]
\[ R_1 = \{B_1 \leftarrow A_1, B_1 \leftarrow A_2, B_2 \leftarrow A_2\} \]

- \(\{A_1\}, \{A_2\}\) are minimal support sets for \(B_1\)
- \(\{A_1, A_2\}\) is a support set for \(B_1\), but not minimal since \(\{A_1\}\) is also a (minimal) support set for \(B_1\).
Finite support condition

**Definition**

$DB$ satisfies the **finite support condition** when there are only a finite number of minimal support sets for every $B \in head(R)$.

Usual programs seem to satisfy the finite support condition.
Finite support condition

Definition

$DB$ satisfies the **finite support condition** when there are only a finite number of minimal support sets for every $B \in head(R)$.

Usual programs seem to satisfy the finite support condition.

Counter example:

\[
DB_2 = F_2 \cup R_2
\]

\[
F_2 = \{ A_i \mid i \in \mathbb{N} \}
\]

\[
R_2 = \{ B_1 \leftarrow A_i \mid i \in \mathbb{N} \}
\]

$DB_2$ does **not** satisfy the finite support condition since for each $i \in \mathbb{N}$, \{A_i\} is a minimal support set for $B_1$. 
Properties of $P_{DB}$

Is $P_{DB}$ a distribution over minimal models?

Table: $P_{DB_1}$

<table>
<thead>
<tr>
<th>$\omega = \langle x_1, x_2, y_1, y_2 \rangle$</th>
<th>$P_{DB_1}(x_1, x_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle 0, 0, 0, 0 \rangle$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\langle 1, 0, 1, 0 \rangle$</td>
<td>0.3</td>
</tr>
<tr>
<td>$\langle 0, 1, 1, 1 \rangle$</td>
<td>0.4</td>
</tr>
<tr>
<td>$\langle 1, 1, 1, 1 \rangle$</td>
<td>0.1</td>
</tr>
<tr>
<td>others</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Remark

$\omega \in \Omega_{DB}$ represents a possible model if and only if

$\omega = M_{DB}(\omega|_{F})$
Properties of $P_{DB}$

Let

$$fix(DB) \overset{\text{def}}{=} \{ M_{DB}(\omega) \mid \omega \in \Omega_F \}$$

The collections of least models derived from possible interpretations for $F$. Also,

$$fix(DB) = \{ \omega \in \Omega_{DB} \mid \omega = M_{DB}(\omega|_F) \}$$

**Theorem 2.1**

If $DB$ satisfies the finite support condition, $fix(DB)$ is $P_{DB}$-measurable. Also $P_{DB}(fix(DB)) = 1$. 
Properties of $P_{DB}$

Proof idea:

**Lemma 2.1**

Suppose $DB$ satisfies the finite support condition. For $\omega = \langle x_1, x_2, \ldots \rangle$,

$$\omega = M_{DB}(\omega|_F) \iff \forall n \left[ A_1^{x_1} \land \cdots \land A_n^{x_n} \right]_F \neq \emptyset$$

For a finite vector $\langle y_1, \ldots, y_n \rangle$ define $E_{y_1,\ldots,y_n} \subset \Omega_{DB}$ by

$$E_{y_1,\ldots,y_n} \overset{\text{def}}{=} \{ \langle y_1, \ldots, y_n, *, *, \ldots \rangle \in \Omega_{DB} \mid \left[ A_1^{y_1} \land \cdots \land A_n^{y_n} \right]_F = \emptyset \}$$

where * is a don’t care symbol.

$$\omega \neq M_{DB}(\omega|_F) \iff \exists n \left[ A_1^{x_1} \land \cdots \land A_n^{x_n} \right]_F = \emptyset$$

$$\iff \omega \in \bigcup_{n=1}^{\infty} \bigcup_{y_1,\ldots,y_n} E_{y_1,\ldots,y_n}$$

$$\Rightarrow \{ \omega \in \Omega_{DB} \mid \omega \neq M_{DB}(\omega|_F) \} \text{ is a null set}$$
Theorem 2.2

If $P_F$ gives probability 1 to $\{\omega_0\} \subset \Omega_F$, $P_{DB}$ gives probability 1 to $\{M_{DB}(\omega_0)\} \subset \Omega_{DB}$. 
Properties of $P_{DB}$

**Definition**

$\{A_1, \ldots, A_n\} \subset F$ finitely determines $\{B_1, \ldots, B_k\}$ if $\{A_1, \ldots, A_n\}$ includes all minimal support sets for every atom in $\{B_1, \ldots, B_k\}$

**Lemma 2.2**

If $\{A_1, \ldots, A_n\} \subset F$ finitely determines $\{B_1, \ldots, B_k\}$, the truth values of $\{B_1, \ldots, B_k\}$ are uniquely determined by those of $\{A_1, \ldots, A_n\}$

Let $\varphi_{DB}(x_1, \ldots, x_n)$ designate this functional relationship.
Properties of $P_{DB}$

Proposition 2.1

Suppose $\{A_1, \ldots, A_n\} \subset F$ finitely determines $\{B_1, \ldots, B_k\}$, Then

$$P_{DB}(A_1 = x_1, \ldots, A_n = x_n, B_1 = y_1, \ldots, B_k = y_k)$$

$$\begin{cases} 
P_F(A_1 = x_1, \ldots, A_n = x_n) & \text{if } \varphi_{DB}(x_1, \ldots, x_n) = \langle y_1, \ldots, y_k \rangle \\
0 & \text{otherwise}
\end{cases}$$

$$P_{DB}(B_1 = y_1, \ldots, B_k = y_k)$$

$$= \sum_{\varphi_{DB}(x_1, \ldots, x_n) = \langle y_1, \ldots, y_k \rangle} P_F(A_1 = x_1, \ldots, A_n = x_n)$$
Outline for section 3

1. Introduction
2. Distribution semantics
   - Preliminaries
   - The existence of $P_F$
   - From $P_F$ to $P_{DB}$
   - Properties of $P_{DB}$
3. EM learning
4. BS-programs
   - BS-programs
   - A learning algorithm for BS-programs
5. A learning experiment
6. Conclusion
A distribution $P_F$ to facts $F$

induces

A distribution $P_{DB}$ for $\text{head}(R)$
A distribution $P_F$ to facts $F$ learns a distribution $P_{DB}$ for $head(R)$.
EM learning

1. Observe truth values of some atoms $B_1, \ldots, B_k$ and obtain an empirical distribution $P_{\text{obs}}(B_1 = y_1, \ldots, B_k = y_k)$

2. Write a logic program $DB = F \cup R$ such that
   \[
   \{B_1, \ldots, B_k\} \subseteq \operatorname{head}(R)
   \]

3. Set an initial basic distribution $P_F$ to $F$ and try to make $P_{DB}(B_1 = y_1, \ldots, B_k = y_k)$ as similar to $P_{\text{obs}}(B_1 = y_1, \ldots, B_k = y_k)$ as possible by adjusting $P_F$
Maximum likelihood estimation (MLE) - given the observations $\langle B_1 = y_1, \ldots, B_k = y_k \rangle$, adjust the parameter $\theta$ of a parameterized distribution $P_F(\theta)$ so that $P_{DB}(B_1 = y_1, \ldots, B_k = y_k \mid \theta)$ attains the optimum.
Maximum likelihood estimation (MLE) - given the observations $\langle B_1 = y_1, \ldots, B_k = y_k \rangle$, adjust the parameter $\theta$ of a parameterized distribution $P_F(\theta)$ so that $P_{DB}(B_1 = y_1, \ldots, B_k = y_k \mid \theta)$ attains the optimum.

Problem: We cannot apply MLE to $P_{DB}$ because $\theta$ does not govern $P_{DB}$ directly.
MLE

Figure: Maximum likelihood

<table>
<thead>
<tr>
<th></th>
<th>Coin A</th>
<th>Coin B</th>
</tr>
</thead>
<tbody>
<tr>
<td>H T T T H H T H T H</td>
<td>5 H, 5 T</td>
<td></td>
</tr>
<tr>
<td>H H H H T H H H H H</td>
<td>9 H, 1 T</td>
<td></td>
</tr>
<tr>
<td>H T H H H H T H T H</td>
<td>8 H, 2 T</td>
<td></td>
</tr>
<tr>
<td>H T H T T T H T H T</td>
<td>4 H, 6 T</td>
<td></td>
</tr>
<tr>
<td>T H H H T H H H T H</td>
<td>7 H, 3 T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24 H, 6 T</td>
<td>9 H, 11 T</td>
</tr>
</tbody>
</table>

\[ \hat{\theta}_A = \frac{24}{24 + 6} = 0.80 \]

\[ \hat{\theta}_B = \frac{9}{9 + 11} = 0.45 \]
Expectation-Maximization (EM) algorithm - an iterative method used in statistics to compute maximum likelihood estimates with incomplete data.

Suppose

- $f(x, y \mid \theta)$ is a parameterized distribution function
- We could not observe complete data $\langle x, y \rangle$, but only observed $y$

The EM algorithm performs MLE by estimating both missing data $x$ and parameter $\theta$

- **E-step** - Estimate missing data
- **M-step** - Calculate new parameters using MLE
EM algorithm

Figure: Expectation maximization
EM algorithm

In our case

- We have incomplete observations \( \langle B_1 = y_1, \ldots, B_k = y_k \rangle \)
- We need to supplement them by "missing observations" \( \langle A_1 = x_1, A_2 = x_2, \ldots \rangle \)
- We need to estimate parameters \( \theta_1, \theta_2, \ldots \) lurking in \( P_{DB}(A_1 = x_1, A_2 = x_2, \ldots, B_1 = y_1, \ldots, B_k = y_k) \)

The EM algorithm applies when DB satisfies the finite support condition.
An EM learning schema for $\vec{B} = \vec{y}$

Suppose $\vec{A} = \langle A_1, \ldots, A_n \rangle \subset F$ finitely determines $\vec{B} = \langle B_1, \ldots, B_k \rangle \subset head(R)$. Also suppose the distribution of $\vec{A}$ is parameterized by $\vec{\theta} = \langle \theta_1, \ldots, \theta_h \rangle$.

EM learning schema

1. Start from $\vec{\theta}_0$ such that $P_{DB}(\vec{y} \mid \vec{\theta}_0) > 0$
2. Suppose $\vec{\theta}_n$ has been computed. Find $\theta_{n+1}$ such that $Q(\theta_{n+1} \mid \vec{\theta}_n) > Q(\vec{\theta}_n \mid \vec{\theta}_n)$
3. Repeat Step 2 until $Q(\vec{\theta}_n \mid \vec{\theta}_n)$ saturates

Where

$$Q(\vec{\theta} \mid \vec{\theta}_n) \overset{\text{def}}{=} \sum_{\vec{x} : P_{DB}(\vec{x} \mid \vec{y}, \vec{\theta}_n) > 0} P_{DB}(\vec{x} \mid \vec{y}, \vec{\theta}_n) \ln P_F(\vec{x} \mid \vec{\theta})$$

---

1 we abbreviate $P_{DB}(\vec{A} = \vec{x}, \vec{B} = \vec{y} \mid \vec{\theta})$ to $P_{DB}(\vec{x}, \vec{y} \mid \vec{\theta})$ etc
Outline for section 4

1. Introduction
2. Distribution semantics
   • Preliminaries
   • The existence of $P_F$
   • From $P_F$ to $P_{DB}$
   • Properties of $P_{DB}$
3. EM learning
4. BS-programs
   • BS-programs
   • A learning algorithm for BS-programs
5. A learning experiment
6. Conclusion
DB = F \cup R \text{ is a BS-program if}

- atoms in $F$ take the form $bs(i, n, 1)$ or $bs(i, n, 0)$
  - $i$ - group identifier
- exactly one of $bs(i, n, 0), bs(i, n, 1)$ is true. The probability of $bs(i, n, 1)$ being true is $\theta_i$
- if $n \neq n'$, $bs(i, n, x)$ and $bs(i', n', x)$ are independent and identically distributed
- if $i \neq i'$, $bs(i, \cdot, \cdot)$ and $bs(i', \cdot, \cdot)$ are independent
BS-programs

For example, a series of 3 coin tosses is described by

\[ A_1 = bs(\text{coin}, 1, 1) \]
\[ A_2 = bs(\text{coin}, 2, 1) \]
\[ A_3 = bs(\text{coin}, 3, 1) \]

Assume coin probability \( \theta \), then

\[
P_F(A_1 = 1, A_2 = 1, A_3 = 0) = P_F(A_1 = 1)P_F(A_2 = 1)P_F(A_3 = 0)
\]
\[
= \theta^2(1 - \theta)
\]
BS-programs

To specify $P_F$, it suffices to define $P_F(A_1 = x_1, \ldots, A_n = x_n)$ for $n \in \mathbb{N}$ where $A_i$ is a bs-atom of the form $bs(i, \cdot, 1)$.

For an equation $\vec{A} = \vec{x}$

- $|\vec{A} = \vec{x}|_1$ denotes the number of equations of the form $bs(i, \cdot, 1) = 1$
- $|\vec{A} = \vec{x}|_0$ denotes the number of equations of the form $bs(i, \cdot, 1) = 0$
- $G_{id}(\vec{A})$ denotes the set of group identifiers in $\vec{A}$

For example,

$$\vec{A} = \langle bs(\text{coin}, 1, 1), bs(\text{coin}, 2, 1), bs(\text{coin}, 3, 1) \rangle = \langle 1, 1, 0 \rangle$$

$$G_{id}(\vec{A}) = \{ \text{coin} \}$$

$$|\vec{A}^{\text{coin}} = \vec{x}|_1 = 2$$

$$|\vec{A}^{\text{coin}} = \vec{x}|_0 = 1$$
Let $\vec{A}$ be a vector of bs-atoms of the form $bs(i, \cdot, 1)$ and $\theta_i$ the probability of $bs(i, \cdot, 1)$ being true. $P_F$ is given by

$$P_F(\vec{A} = \vec{x} \mid \vec{\theta}) = \prod_{i \in G_{id}(\vec{A})} \theta_i^{\vec{A} = \vec{x} \mid 1} \cdot (1 - \theta_i)^{\vec{A} = \vec{x} \mid 0}$$
Suppose $DB = F \cup R$ is a BS-program which satisfies the finite support condition. For $\vec{B} \subset head(\mathcal{R})$ let $\vec{A}$ be the (finite) set of atoms in $F$ that determine $\vec{B}$. By Proposition 2.1,

$$P_{DB}(\vec{y} \mid \vec{\theta}) = \sum_{\varphi(\vec{x}) = \vec{y}} P_{F}(\vec{x} \mid \vec{\theta})$$
A learning algorithm for BS-program

Let \( \langle \vec{B}_1 = \vec{y}_1, \ldots, \vec{B}_M = \vec{y}_M \rangle \) be the result of \( M \) independent observations

A learning algorithm for BS-programs

1. Choose any \( \vec{\theta}^{(0)} \) such that \( P_{DB}(\vec{y}_m \mid \vec{\theta}^{(0)}) > 0 \) for \( \forall m (1 \leq m \leq M) \)

2. Until \( \prod_{m=1}^{M} P_{DB}(\vec{y}_m \mid \vec{\theta}) \) saturates, Repeat

   1. Renew \( \vec{\theta}^{(n)} \) to \( \vec{\theta}^{(n+1)} \) by
      
      For \( i \in \{i_1, \ldots, i_\ell\} \), renew \( \theta_i^{(n)} \) to \( \theta_i^{(n+1)} \) where

      \[
      \theta_i^{(n+1)} = \frac{\text{ON}_i(\vec{\theta}^{(n)})}{\text{ON}_i(\vec{\theta}^{(n)}) + \text{OFF}_i(\vec{\theta}^{(n)})}
      \]

      \[
      \text{ON}_i(\vec{\theta}) \overset{\text{def}}{=} \sum_{m=1}^{M} \sum_{\varphi_{DB}(\vec{x}_m) = \vec{y}_m} \frac{P_F(\vec{x}_m \mid \vec{\theta})|\vec{A} = \vec{x}|_1}{P_{DB}(\vec{y}_m \mid \vec{\theta})}
      \]

      \[
      \text{OFF}_i(\vec{\theta}) \overset{\text{def}}{=} \sum_{m=1}^{M} \sum_{\varphi_{DB}(\vec{x}_m) = \vec{y}_m} \frac{P_F(\vec{x}_m \mid \vec{\theta})|\vec{A} = \vec{x}|_0}{P_{DB}(\vec{y}_m \mid \vec{\theta})}
      \]
Outline for section 5

1. Introduction
2. Distribution semantics
   - Preliminaries
   - The existence of $P_F$
   - From $P_F$ to $P_{DB}$
   - Properties of $P_{DB}$
3. EM learning
4. BS-programs
   - BS-programs
   - A learning algorithm for BS-programs
5. A learning experiment
6. Conclusion
Table: The result of an experiment

<table>
<thead>
<tr>
<th>bs-atom</th>
<th>original value</th>
<th>estimated-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$bs(0, T, 1)$</td>
<td>0.3</td>
<td>0.348045</td>
</tr>
<tr>
<td>$bs(1, T, 1)$</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>$bs(2, T, 1)$</td>
<td>0.5</td>
<td>0.496143</td>
</tr>
<tr>
<td>$bs(3, T, 1)$</td>
<td>0.2</td>
<td>0.15693</td>
</tr>
<tr>
<td>$bs(4, T, 1)$</td>
<td>0.0</td>
<td>$4.5499e-06$</td>
</tr>
<tr>
<td>$bs(5, T, 1)$</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
A learning experiment

Figure: Convergence of parameters
Outline for section 6

1. Introduction
2. Distribution semantics
   - Preliminaries
   - The existence of $P_F$
   - From $P_F$ to $P_{DB}$
   - Properties of $P_{DB}$
3. EM learning
4. BS-programs
   - BS-programs
   - A learning algorithm for BS-programs
5. A learning experiment
6. Conclusion
Conclusion

- Distribution semantics
  - Generalization of the least model semantics
  - Can model Turing machines, Bayesian Networks, Hidden Markov Chains
Conclusion

- Distribution semantics
  - Generalization of the least model semantics
  - Can model Turing machines, Bayesian Networks, Hidden Markov Chains
- EM learning enables us to learn parameters from observations
Conclusion

- Distribution semantics
  - Generalization of the least model semantics
  - Can model Turing machines, Bayesian Networks, Hidden Markov Chains
- EM learning enables us to learn parameters from observations
- BS-programs - simpler and practical subclass of logic programs
Conclusion

- **Distribution semantics**
  - Generalization of the least model semantics
  - Can model Turing machines, Bayesian Networks, Hidden Markov Chains
- **EM learning** enables us to learn parameters from observations
- **BS-programs** - simpler and practical subclass of logic programs
- **Later**: Probabilistic programming languages
  - PRISM
  - ProbLog
Conclusion

- Distribution semantics
  - Generalization of the least model semantics
  - Can model Turing machines, Bayesian Networks, Hidden Markov Chains
- EM learning enables us to learn parameters from observations
- BS-programs - simpler and practical subclass of logic programs
- Later: Probabilistic programming languages
  - PRISM
  - ProbLog

Questions?
Predicting the weather with hidden markov models.  

Chuong B. Do and Serafim Batzoglou.  
What is the expectation maximization algorithm?  

M Nishio.  
*Probability theory (in J).*  
1978.