COMP 424 - Artificial Intelligence
Lecture 17: Learning with missing values

Instructors: Joelle Pineau (jpineau@cs.mcgill.ca)

Class web page: www.cs.mcgill.ca/~jpineau/comp424

Unless otherwise noted, all material posted for this course are copyright of the instructor, and cannot be reused or reposted without the instructor’s written permission.

---

Google DeepMind computer AlphaGo beaten by human after 3 tries
South Korea’s Lee Se-dol was victorious after dropping 3 straight matches against AlphaGo

South Korean professional Go player Lee Se-dol, left, shakes hands with Google co-founder Sergey Brin, right, after winning the fourth match of the Google DeepMind Challenge Match against the company’s artificial intelligence program, AlphaGo, in Seoul, South Korea, Sunday. (The Associated Press)
Today’s overview

- Topic: Parameter learning with missing values
- Gradient-based methods
- Expectation maximization (EM)
- K-means clustering

Learning in Bayesian networks

- Given data in the form of instances:

<table>
<thead>
<tr>
<th>Tampering</th>
<th>Fire</th>
<th>Smoke</th>
<th>Alarm</th>
<th>Leaving</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

- Goal: Find parameters of the Bayes net.
- We discussed how to do this using maximum likelihood.
Learning in Bayesian networks

- Suppose now that the values of some variables are not recorded in some instances:

<table>
<thead>
<tr>
<th>Tampering</th>
<th>Fire</th>
<th>Smoke</th>
<th>Alarm</th>
<th>Leaving</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>??</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>??</td>
<td>No</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Can we still estimate the parameters using maximum likelihood?
  - Should we count instances with missing values? If so, how?

Why do we get incomplete data?

- Some variables may **not be assigned values in some instances**.
  - E.g. not all patients undergo all medical tests.

- Some variables may **not be observed in any of the data items**.
  - E.g. viewer preferences for a show may depend on their metabolic cycle (what time they are awake) - which is not usually measured.

- **Problem**: the fact that a value is missing may be indicative of what the value actually is.
  - E.g. patient did not undergo X-ray because she had no bone problems; so X-ray would likely have come out negative.
A case study: CATIE study

- **CATIE** = Clinical Antipsychotic Trial of Intervention and Effectiveness.
  - 18 months, 1460 patients with schizophrenia.

- Data collected in a **Sequential Multiple Assignment Randomized Trial (SMART)**.
  - Each patient is repeatedly randomized over time.
  - Each randomization occurs at a critical decision point (e.g., milestone in the disease process).
  - Timing and number of randomizations may vary across patients and depend on evolving patient-specific information.

---

**Participants**

| Age 18-65 | No first episode or refractory schizophrenia
| Adequate decision-making capacity |
| Medical illnesses and substance use disorders allowed |
| Concomitant medications (except additional AFDs) allowed |

**Phase I**

- Participants randomly assigned to double-blind, adjustable-dose treatment with one of the following: olanzapine, perphenazine*, quetiapine, haloperidol, ziprasidone

  - Responders: Continue assigned treatment for duration of 18-month period
  - Non-responders

**Phase II**

- Participants choose one of two random assignment pathways:
  - Olanzapine or one of the following: clozapine, quetiapine, risperidone
  - Ziprasidone or one of the following: clozapine, quetiapine, risperidone

  - Responders: Continue assigned treatment for duration of 18-month period
  - Non-responders

**Phase III**

- Participants choose one of the following open-label treatments:
  - aripiprazole, olanzapine, fluphenazine decanoate, olanzapine, perphenazine, quetiapine, risperidone, ziprasidone, co-treatment with any two of the above

---

*Copyright © 2008 Nature Publishing Group*
### List of variables collected during CATIE

#### Variables with no missing information:

**Time independent variables:**
- Age (cont), Sex (dich), Race (cat), Tardive dyskinesia status at baseline (dich), Marital status at baseline (dich), Patient education (cat), Hospitalization history in 3 months prior to CATIE (dich), Clinical setting in which patient received CATIE treatment (cat), Treatment prior to CATIE enrollment (cat), stage 1 randomized treatment assignment (cat)

#### Variables with missing information:

**Time independent variables:**
- Employment status at baseline (cat), Years since first prescribed anti-psychotic medication at baseline (cont), Neurocognitive composite score at baseline (cont)

**Variables recorded at all months 1-18 and at end-of-stage visits:**
- Adherence measured by the proportion of capsules taken since last visit (cont)
- Variables recorded at months 0, 1, 3, 6, 9, 12, 15, 18 and at end-of-stage visits:
  - Body mass index (cont), Clinical drug use scale (cat), Clinical alcohol use scale (cat), Clinical Global Impressions of Severity of illness score (cat), Positive and Negative Syndrome Scale (cont), Calgary Depression Score (cont), Simpson-Angus EPS scale (cont), Barnes Akathisia scale (cont), Total movement severity score (cont)

**Variables recorded at months 0, 6, 12, 18 and at end-of-stage visits:**
- Quality of Life total score (cont), SF-12 Mental health summary (cont), SF-12 Physical health summary (cont), Illicit drug use (dich)

**Variables recorded only at end-of-stage visits:**
- Reason for discontinuing treatment (cat), Stage 2 randomization arm (dich, when applicable), Stage 2 treatment (cat, when applicable)

---

### Artificial CATIE dataset

<table>
<thead>
<tr>
<th>( G_0 )</th>
<th>( W_0 )</th>
<th>( P_0 )</th>
<th>( A_1 )</th>
<th>( W_1 )</th>
<th>( P_1 )</th>
<th>( C_1 )</th>
<th>( A_2 )</th>
<th>( P_2 )</th>
<th>( W_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>31.8</td>
<td>103</td>
<td>Perphenazine</td>
<td>23.4</td>
<td>77</td>
<td>SWITCHED</td>
<td>Ziprasidone</td>
<td>86</td>
<td>26.9</td>
</tr>
<tr>
<td>Male</td>
<td>29.4</td>
<td>108</td>
<td>Risperidone</td>
<td>18.2</td>
<td>102</td>
<td>STAYED</td>
<td>NA</td>
<td>88</td>
<td>19</td>
</tr>
<tr>
<td>Male</td>
<td>32.6</td>
<td>63</td>
<td>Olanzapine</td>
<td>35.2</td>
<td>99</td>
<td>STAYED</td>
<td>Olanzapine</td>
<td>85</td>
<td>38.2</td>
</tr>
<tr>
<td>Female</td>
<td>102</td>
<td>Quetiapine</td>
<td>34.6</td>
<td>96</td>
<td>SWITCHED</td>
<td>Ziprasidone</td>
<td>77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>38.1</td>
<td>86</td>
<td>Risperidone</td>
<td>20.8</td>
<td>96</td>
<td>SWITCHED</td>
<td>Olanzapine</td>
<td>71</td>
<td>31.6</td>
</tr>
<tr>
<td>Male</td>
<td>31.1</td>
<td>80</td>
<td>Risperidone</td>
<td>22.8</td>
<td>89</td>
<td>SWITCHED</td>
<td>Clozapine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>31.6</td>
<td>71</td>
<td>Olanzapine</td>
<td>21.1</td>
<td>74</td>
<td>STAYED</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>25.1</td>
<td>64</td>
<td>Perphenazine</td>
<td>19.7</td>
<td>74</td>
<td>STAYED</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>37.9</td>
<td>Risperidone</td>
<td>36</td>
<td>STAYED</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>28.7</td>
<td>91</td>
<td>Perphenazine</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( W = \text{Body-Mass Index} \)

\( P = \text{PANSS score (measure of symptom intensity)} \)

\( A = \text{Treatment assigned} \)
Missing data in CATIE

- **High study attrition**: only 705 of 1460 stayed for full 18 months; 509 dropped out before entering stage 2.
  - High attrition is not unusual for studies of antipsychotics.

- Majority of missing data (78.1%) was due to attrition.

Why missing values make life hard

- Consider a simple network $X \rightarrow Y$, and suppose we want to learn its parameters from samples $<x_1, y_1>, \ldots, <x_m, y_m>$.

- Which parameters do we need?

- Given all samples values, maximize log-likelihood of:
  $$L(\theta_X, \theta_{Y|X=0}, \theta_{Y|X=1}) = (\theta_X)^N (1-\theta_X)^{N0} (\theta_{Y|X=0})^{N0} (1-\theta_{Y|X=0})^{N0} (\theta_{Y|X=1})^{N1} (1-\theta_{Y|X=1})^{N0}$$

- Suppose now that $x_1$ is missing and $y_1=1$. What can we do?
Why missing values make life hard (2)

• We can consider both settings, \(x_1=0, x_1=1\).

• For each setting we get a different likelihood.

• Overall likelihood combines both settings (weighted by probability of that setting).

\[
L(\theta_x, \theta_{y|x}, \theta_{y|x'}) = (1-\theta_x) \Pr(<0,y_1>, <x_2,y_2>, ..., <x_m,y_m> | \theta_x, \theta_{y|x}, \theta_{y|x'}) + \\
\theta_x \Pr(<1,y_1>, <x_2,y_2>, ..., <x_m,y_m> | \theta_x, \theta_{y|x}, \theta_{y|x'})
\]

• **Problem**: If we have values missing for \(x_1\) and \(x_2\), we have to consider all possible values for both instances! Etc.

---

Missing at random assumption

• The probability that the value of \(X_i\) is missing is independent of its actual value, given the observed data.

• If this is not true, for variable \(X_i\), we can introduce an additional Boolean variable, \(X_i^{\text{Observ}}\) and satisfy the assumption.
Effects of missing data

<table>
<thead>
<tr>
<th>Complete data</th>
<th>Missing data</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Parameters of model can be estimated locally and independently.</td>
<td>• Parameters cannot be estimated independently.</td>
</tr>
<tr>
<td>• Log-likelihood has a unique maximum.</td>
<td>• Many local maxima. Maximizing likelihood becomes non-linear optimization problem.</td>
</tr>
<tr>
<td>• Under certain assumptions, there is a nice closed-form solution for parameters.</td>
<td>• No closed-form solution.</td>
</tr>
</tbody>
</table>

Two solutions for maximizing likelihood

1. **Gradient ascent**: Use hill-climbing search through the space of parameters, following the gradient of the likelihood with respect to the parameters.
Gradient ascent

- **Basic idea:** Move parameters in the direction of the log-likelihood.

- **Pros:**
  - Flexible: allows different forms for the Conditional Prob. Distributions.
  - Easy to compute the gradient at any parameter setting.
  - Closely related to other learning methods (e.g., neural nets).

- **Cons:**
  - Solution needs to be projected on space of legal parameters (in our case, need to ensure that we get probability distributions.)
  - Sensitive to parameters (e.g. learning rate).
  - Slow!

Two solutions for maximizing likelihood

1. **Gradient ascent:** hill-climbing search through the space of parameters, following the gradient of the likelihood with respect to the parameters.

2. **Expectation maximization:** use the current parameter settings to construct a local approximation of the likelihood which is “nice” and can be optimized easily.
Expectation Maximization (EM)

- General purpose method for learning from incomplete data (not only Bayes nets), **whenever an underlying distribution is assumed.**

- **Main idea:** Alternate between two steps

  1. For all the instances of missing data, we will “fantasize” how the data should look based on the current parameter setting.
     - This means we compute **expected sufficient statistics.**

  2. Then **maximize parameter setting**, based on these statistics.

Outline of EM

- **Initialization:**
  - Start with some initial parameter setting (e.g. $P(T)$, $P(F)$, $P(A|F,T)$, etc.).
  - These can be estimated from all complete data instances.

- **Repeat:**
  1. **Expectation (E-step):** Complete the data by assigning “values” to the missing items based on current parameter setting.
  2. **Maximization (M-step):** Compute the maximum likelihood parameter setting based on the completed data. This is what we did in Lecture #16.

- **Convergence:**
  - Nothing changes in E-step or M-step between 2 consecutive rounds.
EM in our example

• To start, guess the parameters of the network $\theta$ (using the known data):

  E.g.
  \[
  \theta_x = \frac{N_{x=1}(2:m)}{(m-1)} \\
  \theta_{y|x=0} = \frac{N_{y=1,x=0}(2:m)}{N_{x=0}(2:m)} \\
  \theta_{y|x=1} = \frac{N_{y=1,x=1}(2:m)}{N_{x=1}(2:m)}
  \]

• **E-step:** Using initial $\theta$, compute: $P(x_1=0 \mid y_1), P(x_1=1 \mid y_1)$
  (Note that this step requires *exact inference* - so not cheap!)
  Complete dataset with most likely value of $x_1$. Call new dataset $D^*$.

• **M-step:** Compute new parameter vector $\theta$, which maximizes the likelihood given the completed data: $L(\theta|D^*) = P(D^* \mid \theta)$

  E.g.
  \[
  \theta_x = \frac{N_{x=1}}{m} \\
  \theta_{y|x=0} = \frac{N_{y=1,x=0}}{N_{x=0}} \\
  \theta_{y|x=1} = \frac{N_{y=1,x=1}}{N_{x=1}}
  \]

• Repeat E-step and M-step until the parameter vector converges.

Two versions of the algorithm

• **Hard EM:** for each missing data point, assign the value that is most likely.  
  (This is the version we just saw.)

• **Soft EM:** for each missing data point, put a weight on each value, equal to its probability, and use the weights as counts.  
  (This is the most common version.)

  Then these numbers are used as real counts, to provide a maximum likelihood estimate for $\theta$. 
Soft EM in our example

• To start, guess the parameters of the network \( \theta \) (using the known data):

\[
\begin{align*}
\theta_X &= \frac{N_{x=1}(2,m)}{(m-1)} \\
\theta_{Y|x=0} &= \frac{N_{y=1,x=0}(2,m)}{N_{x=0}(2,m)} \\
\theta_{Y|x=1} &= \frac{N_{y=1,x=1}(2,m)}{N_{x=1}(2,m)}
\end{align*}
\]

• **E-step**: Using initial \( \theta \), compute:

\[
\begin{align*}
w_0 &= P(x_1=0 \mid y_i) \quad w_1 = P(x_1=1 \mid y_i)
\end{align*}
\]

Now hypothesize two datasets:

\[
\begin{align*}
D_0 &= \langle w_0, y_i \rangle, \langle x_2, y_2 \rangle, \ldots, \langle x_m, y_m \rangle \\
D_1 &= \langle w_1, y_i \rangle, \langle x_2, y_2 \rangle, \ldots, \langle x_m, y_m \rangle
\end{align*}
\]

• **M-step**: Compute new parameter vector \( \theta \), which maximizes the expected likelihood given the completed data:

\[
\begin{align*}
L(\theta | D^*) &= w_0 P(D_0 \mid \theta) + w_1 P(D_1 \mid \theta)
\end{align*}
\]

\[
\begin{align*}
\theta_X &= \frac{(N_{x=1}(2,m) + w_1)}{m} \\
\theta_{Y|x=0} &= \frac{(N_{y=1,x=0}(2,m) + w_0)}{(N_{x=0}(2,m) + w_0)} \\
\theta_{Y|x=1} &= \frac{(N_{y=1,x=1}(2,m) + w_1)}{(N_{x=1}(2,m) + w_1)}
\end{align*}
\]

Repeat!
Comparison of hard EM and soft EM

• Soft EM does not commit to specific value for the missing item.
  – Instead, it considers all possible values, with some probability.
  – This is a pleasing property, given the uncertainty in the value.

• Complexity:
  – Hard EM requires computing most probable values.
  – Soft EM requires computing conditional probabilities for completing the missing values.

Same complexity: both require full probabilistic inference - which can be expensive!

Properties of EM

• Likelihood function is guaranteed to improve (or stay the same) with each iteration.
  – Algorithm can be stopped when no more improvement is achieved between iterations.

• EM is guaranteed to converge to a local optimum of the likelihood function.
  – Starting with different values of initial parameters is necessary (random re-starts, to avoid local optimum).

• EM is a widely used algorithm in practice!
A harder example

Suppose we have the simple Bayes net $A \rightarrow B \rightarrow C$, where each node is associated with a Bernoulli random variable. Further suppose we have the following sample data:

(i) $A=1, B=1, C=1$

$P_{\text{data}}(A=1, B=1, C=1) = P(B=1 | A=1, C=1) P(A=1) P(C=1)$

(ii) $A=0, B=1, C=0$

$P_{\text{data}}(A=0, B=1, C=0) = P(B=1 | A=0, C=0) P(A=0) P(C=0)$

(iii) $A=1, B=0, C=0$

$P_{\text{data}}(A=1, B=0, C=0) = P(B=0 | A=1, C=0) P(A=1) P(C=0)$

(iv) $A=1, B=1, C=0$

$P_{\text{data}}(A=1, B=1, C=0) = P(B=1 | A=1, C=0) P(A=1) P(C=0)$

(v) $A=1, B=1, C=1$

$P_{\text{data}}(A=1, B=1, C=1) = P(B=1 | A=1, C=1) P(A=1) P(C=1)$

(vi) $A=0, B=0, C=0$

$P_{\text{data}}(A=0, B=0, C=0) = P(B=0 | A=0, C=0) P(A=0) P(C=0)$

And repeat…

A seemingly harder problem

- What if one of your variables is always missing?

  Data = $<x_1, ?>, <x_2, ?>, ..., <x_m, ?>$

- Can you estimate a maximum-likelihood parameter for this case?

- We call this learning from unlabeled data.
An example

- A fruit merchant approaches you, with a set of apples to classify according to their variety.
  - Tells you there are five varieties of apples in the dataset.
  - Tells you the weight and colour of each apple in the dataset.

- Can you label each apple with the correct variety?
  - What would you need to know / assume?

Plotting the data

What if you observe that the data looks like this?
(One axis is weight, the other is colour.)
K-means clustering

- K-means clustering: cluster instances into K distinct classes.
- Need to know K in advance.
- Need to assume a parametric distribution for each class.

- Back to our apples ...
  - You know there are 5 varieties.
  - Assume each variety generates apples according to a (variety-specific) 2-D Gaussian distribution, \( N(\mu_i, \sigma^2_i) \)
  - If you know \( \mu_i, \sigma^2_i \) for each class, it’s easy to classify the apples!
  - If you know the class of each apple, it’s easy to estimate \( \mu_i, \sigma^2_i \)

  What if we know neither?

K-means algorithm

This data could easily be modeled by Gaussians.

1. Ask user how many clusters.
K-means algorithm

This data could easily be modeled by Gaussians.

1. Ask user how many clusters.
2. Randomly guess k centers:
   \[ \{ \mu_1, \ldots, \mu_k \} \text{ (assume } \sigma^2 \text{ is known).} \]
3. Assign each data point to the closest center.
K-means algorithm

This data could easily be modeled by Gaussians.

1. Ask user how many clusters.
2. Randomly guess \( k \) centers:
   \( \{ \mu_1, \ldots, \mu_k \} \) (assume \( \sigma^2 \) is known).
3. Assign each data point to the closest center.
4. Each center finds the centroid of the points it owns.

and jumps there.
K-means algorithm starts

K-means algorithm continues (2)
K-means algorithm continues (3)

Image courtesy of Andrew Moore, Carnegie Mellon U.

K-means algorithm continues (4)

Image courtesy of Andrew Moore, Carnegie Mellon U.
K-means algorithm continues (5)

Image courtesy of Andrew Moore, Carnegie Mellon U.

K-means algorithm continues (6)

Image courtesy of Andrew Moore, Carnegie Mellon U.
K-means algorithm continues (7)

Image courtesy of Andrew Moore, Carnegie Mellon U.

K-means algorithm continues (8)

Image courtesy of Andrew Moore, Carnegie Mellon U.
K-means algorithm continues (9)

K-means algorithm terminates
Properties of K-means

- **Time complexity?** \( O(nkd) \) where
  - \( k \) = #centers
  - \( n \) = #datapoints
  - \( d \) = dimensionality of data

- **Optimality?**
  - Converges to a local optimum.
  - Can use random re-starts to get better local optimum.
  - Alternately, can choose your initial centers carefully:
    - Place \( \mu_1 \) on top of a randomly chosen datapoint.
    - Place \( \mu_2 \) on top of datapoint that is furthest from \( \mu_1 \).
    - Place \( \mu_3 \) on top of datapoint that is furthest from both \( \mu_1 \) and \( \mu_2 \).

- **Maximum number of iterations?** *(This is harder…)*

What you should know

- Learning maximum-likelihood parameters with missing data.
  - EM algorithm in general case.
  - Properties of EM (complexity, convergence).

- Clustering of data using K-means algorithm
  - Basic procedure.
  - Properties of K-means (complexity, convergence).

- Relation between EM and K-means.
Final notes

• Homework 4 out today, due Mar.25.

• Final project due Apr.7 (code) and Apr.8 (report).