

**CS 2750 Machine Learning**  
**Lecture 16**

**Learning BBNs with hidden variables and missing values. Expectation Maximization (EM).**

Milos Hauskrecht  
[milos@cs.pitt.edu](mailto:milos@cs.pitt.edu)  
5329 Sennott Square

---

CS 2750 Machine Learning

**Learning probability distribution**

**Basic learning settings:**

- A set of random variables  $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$
- **A model of the distribution** over variables in  $\mathbf{X}$  with parameters  $\Theta$
- **Data**  $D = \{D_1, D_2, \dots, D_N\}$   
    **s.t.**  $D_i = (x_1^i, x_2^i, \dots, x_n^i)$

**Objective:** find parameters  $\hat{\Theta}$  that describe the data

**Assumptions considered so far:**

- Known parameterizations
- No hidden variables
- No-missing values

---

CS 2750 Machine Learning

## Hidden variables

### Modeling assumption:

Variables  $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$

- Additional variables are hidden – never observed in data

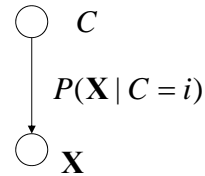
### Why to add hidden variables?

- **More flexibility in describing the distribution**  $P(\mathbf{X})$
- **Smaller parameterization of**  $P(\mathbf{X})$ 
  - **New independences can be introduced via hidden variables**

### Example:

- Latent variable models
  - hidden classes (categories)

Hidden class variable



CS 2750 Machine Learning

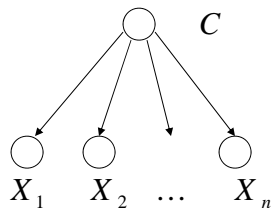
## Naïve Bayes with a hidden class variable

**Introduction of a hidden variable can reduce the number of parameters defining**  $P(\mathbf{X})$

### Example:

- Naïve Bayes model with a hidden class variable

Hidden class variable



Attributes are independent given the class

- **Useful in customer profiles**
  - Class value = type of customers

CS 2750 Machine Learning

## Missing values

A set of random variables  $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$

• **Data**  $D = \{D_1, D_2, \dots, D_N\}$

• **But some values are missing**

$$D_i = (x_1^i, x_3^i, \dots, x_n^i)$$

Missing value of  $x_2^i$

$$D_{i+1} = (x_3^{i+1}, \dots, x_n^{i+1})$$

Missing values of  $x_1^{i+1}, x_2^{i+1}$

Etc.

• **Example: medical records**

• **We still want to estimate parameters of**  $P(\mathbf{X})$

---

CS 2750 Machine Learning

## Density estimation

**Goal: Find the set of parameters**  $\hat{\Theta}$

**Estimation criteria:**

– **ML**  $\max_{\Theta} p(D | \Theta, \xi)$

– **Bayesian**  $p(\Theta | D, \xi)$

**Optimization methods for ML:** gradient-ascent, conjugate gradient, Newton-Rhapon, etc.

**Problem:** No or very small advantage from the structure of the corresponding belief network when unobserved variable values

**Expectation-maximization (EM) method**

– An alternative optimization method

– Suitable when there are missing or hidden values

– **Takes advantage of the structure of the belief network**

---

CS 2750 Machine Learning

## General EM

**The key idea of a method:**

**Compute the parameter estimates** iteratively by performing the following two steps:

**Two steps of the EM:**

1. **Expectation step.** Complete all hidden and missing variables with expectations for the current set of parameters  $\Theta'$
2. **Maximization step.** Compute the new estimates of  $\Theta$  for the completed data

**Stop when no improvement possible**

---

CS 2750 Machine Learning

## EM

Let  $H$  – be a set of hidden or missing values

**Derivation**

$$P(H, D | \Theta, \xi) = P(H | D, \Theta, \xi)P(D | \Theta, \xi)$$

$$\log P(H, D | \Theta, \xi) = \log P(H | D, \Theta, \xi) + \log P(D | \Theta, \xi)$$

$$\log P(D | \Theta, \xi) = \log P(H, D | \Theta, \xi) - \log P(H | D, \Theta, \xi)$$

 **Log-likelihood of data**

**Average both sides** with  $P(H | D, \Theta', \xi)$  **for some**  $\Theta'$

$$E_{H|D, \Theta'} \log P(D | \Theta, \xi) = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi) - E_{H|D, \Theta'} \log P(H | \Theta, \xi)$$

$$\underbrace{\log P(D | \Theta, \xi)}_{\text{Log-likelihood of data}} = Q(\Theta | \Theta') + H(\Theta | \Theta')$$

**Log-likelihood of data**

---

CS 2750 Machine Learning

## EM algorithm

**Algorithm** (general formulation)

Initialize parameters  $\Theta$

Repeat

Set  $\Theta' = \Theta$

1. **Expectation step**

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

2. **Maximization step**

$$\Theta = \arg \max_{\Theta} Q(\Theta | \Theta')$$

until no or small improvement in  $\Theta$  ( $\Theta = \Theta'$ )

**Questions:** Why this leads to the ML estimate ?

What is the advantage of the algorithm?

---

CS 2750 Machine Learning

## EM algorithm

- Why is the EM algorithm correct?
- **Claim: maximizing Q improves the log-likelihood**

$$l(\Theta) = Q(\Theta | \Theta') + H(\Theta | \Theta')$$

**Difference in log-likelihoods (current and next step)**

$$l(\Theta) - l(\Theta') = Q(\Theta | \Theta') - Q(\Theta' | \Theta') + H(\Theta | \Theta') - H(\Theta' | \Theta')$$

**Subexpression**  $H(\Theta | \Theta') - H(\Theta' | \Theta') \geq 0$

**Kullback-Leibler (KL) divergence** (distance between 2 distributions)

$$KL(P | R) = \sum_i P_i \log \frac{P_i}{R_i} \geq 0 \quad \text{Is always positive !!!}$$

$$H(\Theta | \Theta') = -E_{H|D, \Theta'} \log P(H | \Theta, D, \xi) = -\sum_i p(H | D, \Theta') \log P(H | \Theta, D, \xi)$$

$$H(\Theta | \Theta') - H(\Theta' | \Theta') = \sum_i P(H | D, \Theta') \log \frac{P(H | \Theta', D, \xi)}{P(H | \Theta, D, \xi)} \geq 0$$

---

CS 2750 Machine Learning

## EM algorithm

### Difference in log-likelihoods

$$l(\Theta) - l(\Theta') = Q(\Theta | \Theta') - Q(\Theta' | \Theta') + H(\Theta | \Theta') - H(\Theta' | \Theta')$$

$$l(\Theta) - l(\Theta') \geq Q(\Theta | \Theta') - Q(\Theta' | \Theta')$$

Thus

by **maximizing Q** we maximize the log-likelihood

$$l(\Theta) = Q(\Theta | \Theta') + H(\Theta | \Theta')$$

EM is a first-order optimization procedure

- **Climbs the gradient**
- **Automatic learning rate**

**No need to adjust the learning rate !!!!**

---

CS 2750 Machine Learning

## EM advantages

### Key advantages:

- In many problems (e.g. Bayesian belief networks)

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

- has a nice form and the maximization of Q can be carried out in the closed form
- No need to compute Q before maximizing
- We directly optimize
  - using quantities corresponding to expected counts

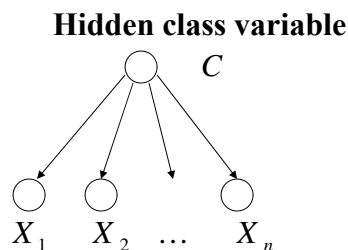
---

CS 2750 Machine Learning

## Naïve Bayes with a hidden class and missing values

### Assume:

- $P(\mathbf{X})$  is modeled using a Naïve Bayes model with hidden class variable
- Missing entries (values) for attributes in the dataset  $D$



Attributes are independent given the class

CS 2750 Machine Learning

## EM for the Naïve Bayes

- We can use EM to learn the parameters

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

- **Parameters:**

$\pi_j$  prior on class  $j$

$\theta_{ijk}$  probability of an attribute  $i$  having value  $k$  given class  $j$

- **Indicator variables:**

$\delta_j^l$  for example  $l$ , the class is  $j$ ; if true (=1) else false (=0)

$\delta_{ijk}^l$  for example  $l$ , the class is  $j$  and the value of attrib  $i$  is  $k$

- because the class is hidden and some attributes are missing, the values (0,1) of indicator variables are not known; they are hidden

$H$  – a collection of all indicator variables

CS 2750 Machine Learning

## EM for the Naïve Bayes model

- We can use EM to do the learning of parameters

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

$$\begin{aligned} \log P(H, D | \Theta, \xi) &= \log \prod_{l=1}^N \prod_j \pi_j^{\delta_j^l} \prod_i \prod_k \theta_{ijk}^{\delta_{ijk}^l} \\ &= \sum_{l=1}^N \sum_j (\delta_j^l \log \pi_j + \sum_i \sum_k \delta_{ijk}^l \log \theta_{ijk}) \end{aligned}$$

$$E_{H|D, \Theta'} \log P(H, D | \Theta, \xi) = \sum_{l=1}^N \sum_j (E_{H|D, \Theta'}(\delta_j^l) \log \pi_j + \sum_i \sum_k E_{H|D, \Theta'}(\delta_{ijk}^l) \log \theta_{ijk})$$

$$\begin{aligned} E_{H|D, \Theta'}(\delta_j^l) &= p(C_l = j | D_l, \Theta') && \text{Substitutes 0,1} \\ E_{H|D, \Theta'}(\delta_{ijk}^l) &= p(X_{il} = k, C_l = j | D_l, \Theta') && \text{with expected value} \end{aligned}$$

CS 2750 Machine Learning

## EM for the Naïve Bayes model

- Computing derivatives of  $Q$  for parameters and setting it to 0 we get:

$$\pi_j = \frac{\tilde{N}_j}{N} \qquad \theta_{ijk} = \frac{\tilde{N}_{ijk}}{\sum_{k=1}^{r_i} \tilde{N}_{ijk}}$$

$$\tilde{N}_j = \sum_{l=1}^N E_{H|D, \Theta'}(\delta_j^l) = \sum_{l=1}^N p(C_l = j | D_l, \Theta')$$

$$\tilde{N}_{ijk} = \sum_{l=1}^N E_{H|D, \Theta'}(\delta_{ijk}^l) = \sum_{l=1}^N p(X_{il} = k, C_l = j | D_l, \Theta')$$

- Important:**
  - Use expected counts instead of counts !!!
  - Re-estimate the parameters using expected counts

CS 2750 Machine Learning



## EM for BBNs

- The same result applies to learning of parameters of **any Bayesian belief network** with discrete-valued variables

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

$$\theta_{ijk} = \frac{\tilde{N}_{ijk}}{\sum_{k=1} r_i} \leftarrow \text{Parameter value maximizing } Q$$

$$\tilde{N}_{ijk} = \sum_{l=1}^N p(x_i^l = k, pa_i^l = j | D^l, \Theta')$$

may require inference

- **Again:**
  - Use expected counts instead of counts