CS 2750 Machine Learning
Lecture 16

Learning Bayesian belief networks

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Learning of BBN

Learning.

- **Learning of parameters of conditional probabilities**
- **Learning of the network structure**

Variables:

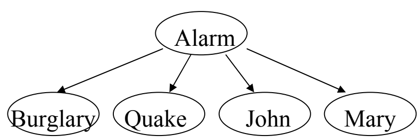
- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

Already Covered: Learning of parameters of BBN

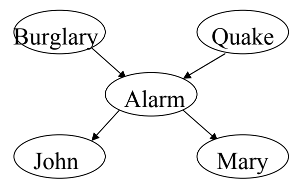
1. With observable variables
2. Hidden variables and missing values

Model selection

- **BBN has two components**:
  - **Structure of the network** (models conditional independences)
  - **A set of parameters** (conditional child-parent distributions)
- How to learn the structure?



?



Learning the structure

Criteria we can choose to score the structure S

- **Marginal likelihood**

  maximize $P(D \mid S, \xi)$

  ξ - represents the prior knowledge

- **Maximum posterior probability**

  maximize $P(S \mid D, \xi)$

$$P(S \mid D, \xi) = \frac{P(D \mid S, \xi) P(S \mid \xi)}{P(D \mid \xi)}$$

How to compute marginal likelihood $P(D \mid S, \xi)$?

Learning of BBNs

- **Notation**:
  - i ranges over all possible variables $i=1,..,n$
  - $j=1,..,q$ ranges over all possible parent combinations
  - $k=1,..,r$ ranges over all possible variable values
  - $\boldsymbol{\Theta}$ - parameters of the BBN

  Θ_{ij} is a vector of Θ_{ijk} representing parameters of the conditional probability distribution; such that $\sum_{k=1}^{r} \Theta_{ijk} = 1$

  N_{ijk} - a number of instances in the dataset where parents of variable X_i take on values j and X_i has value k

  $N_{ij} = \sum_{k=1}^{r} N_{ijk}$

  α_{ijk} - prior counts (parameters of Beta and Dirichlet priors)

  $\alpha_{ij} = \sum_{k=1}^{r} \alpha_{ijk}$

Marginal likelihood

- Integrate over all possible parameter settings

$$P(D \mid S,\xi) = \int_{\Theta} P(D \mid S,\Theta,\xi)\, p(\Theta \mid S,\xi)\, d\Theta$$

- Using the assumption of parameter and sample independence

$$P(D \mid S,\xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- We can use **log-likelihood score** instead

$$\log P(D \mid S,\xi) = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right\}$$

Score is decomposable along variables !!!

Appendix: marginal likelihood

- **Marginal likelihood**

$$P(D \mid S,\xi) = \int_{\Theta} P(D \mid S,\Theta,\xi)\, p(\Theta \mid S,\xi)\, d\Theta$$

- **From the iid assumption:**

$$P(\boldsymbol{D} \mid S,\boldsymbol{\Theta}) = \prod_{h=1}^{N} \prod_{i=1}^{n} P(x_i^h \mid parents_i^h, \boldsymbol{\Theta})$$

- Let r_i = number of values that attribute x_i can take
  q_i= number of possible parent combinations
  N_{ijk}= number of cases in D where x_i has value k and parents with values j.

$$= \prod_{i}^{n} \prod_{j}^{q_i} \prod_{k}^{r_i} P(x_i = k \mid parents_i = j, \boldsymbol{\Theta})^{N_{ijk}}$$

$$= \prod_{i}^{n} \prod_{j}^{q_i} \prod_{k}^{r_i} \theta_{ijk}^{N_{ijk}}$$

Appendix: the marginal likelihood

$$P(D \mid S,\xi) = \int_{\Theta} P(D \mid S,\Theta,\xi)\, p(\Theta \mid S,\xi)\, d\Theta$$

- **From parameter independence**

$$p(\boldsymbol{\Theta} \mid S,\xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\Theta_{ij} \mid S,\xi)$$

- **Priors for** $p(\Theta_{ij} \mid S,\xi)$
  - $\Theta_{ij} = (\Theta_{ij1},\ldots,\Theta_{ijr_i})$ is a vector of parameters;
  - we use a Dirichlet distribution with parameters $\boldsymbol{\alpha}$

$$P(\Theta_{ij} \mid S,\xi) = P(\Theta_{ij1},\ldots,\Theta_{ijr_i} \mid S,\xi) = Dir(\Theta_{ij1},\ldots,\Theta_{ijr_i} \mid \alpha)$$

$$= \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \Theta_{ijk}^{\alpha_{ijk}-1}$$

Appendix: marginal likelihood

- **Combine things together:**

$$P(D \mid S_i) = \int_{\Theta} P(D \mid S_i, \Theta) P(\Theta \mid S_i) d\Theta$$

$$= \int \prod_i^n \prod_j^{q_i} \prod_k^{r_i} \Theta_{ijk}^{N_{ijk}} \cdot \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \Theta_{ijk}^{\alpha_{ijk}-1} d\Theta$$

$$= \prod_i^n \prod_j^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \int \prod_{k=1}^{r_i} \Theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1} d\Theta$$

$$= \prod_i^n \prod_j^{q_i} \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \cdot \frac{\prod_{k=1}^{r_i} \Gamma(a_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij} + N_{ij})}$$

A trick to compute the marginal likelihood

- Integrate over all possible parameter settings

$$P(D \mid S, \xi) = \int_{\Theta} P(D \mid S, \Theta, \xi) p(\Theta \mid S, \xi) d\Theta$$

- Posterior of parameters, given data and the structure

$$p(\mathbf{\Theta} \mid D, S, \xi) = \frac{P(D \mid \mathbf{\Theta}, S, \xi) p(\mathbf{\Theta} \mid S, \xi)}{P(D \mid S, \xi)}$$

Trick

$$P(D \mid S, \xi) = \frac{P(D \mid \mathbf{\Theta}, S, \xi) p(\mathbf{\Theta} \mid S, \xi)}{p(\mathbf{\Theta} \mid D, S, \xi)}$$

- Gives the solution

$$P(D \mid S, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

Learning the structure

- **Likelihood of data for the BBN** (structure and parameters)

$$P(D \mid S, \Theta, \xi)$$

measures the goodness of fit of the BBN to data

- **Marginal likelihood** (for the structure only)

$$P(D \mid S, \xi)$$

- **Does not measure only a goodness of fit. It is:**
  - different for structures of different complexity
  - Incorporates preferences towards simpler structures, **implements Occam's razor** !!!!

Occam's Razor

- Why there is a preference towards simpler structures ?

Rewrite marginal likelihood as

$$P(D \mid S, \xi) = \frac{\int_{\Theta} P(D \mid S, \Theta, \xi)\, p(\Theta \mid S, \xi)\, d\Theta}{\int_{\Theta} p(\Theta \mid S, \xi)\, d\Theta}$$

We know that $\int_{\Theta} p(\Theta \mid S, \xi)\, d\Theta = 1$

Interpretation: in more complex structures there are more ways parameters can be set badly

- **The numerator:** count of good assignments
- **The denominator:** count of all assignments

Approximations of probabilistic scores

Approximations of the marginal likelihood and posterior scores

- **Information based measures**
  - Akaike criterion
  - Bayesian information criterion (BIC)
  - Minimum description length (MDL)
- Reflect the tradeoff between the fit to data and preference towards simpler structures

Example: **Akaike criterion.**

Maximize: $score(S) = \log P(D \mid S, \Theta_{ML}, \xi) - \text{compl(S)}$

Bayesian information criterion (BIC)

Maximize:

$$score(S) = \log P(D \mid S, \Theta_{ML}, \xi) - \frac{1}{2}\text{compl(S) logN}$$

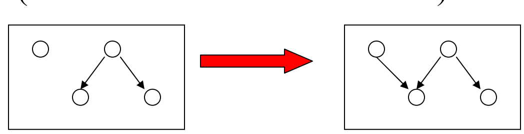
Optimizing the structure

Finding the best structure is a **combinatorial optimization** problem

- A good feature: the score **is decomposable along variables**:

$$\log P(D \mid S, \xi) = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right\}$$

Algorithm idea: Search the space of structures using local changes (additions and deletions of a link)



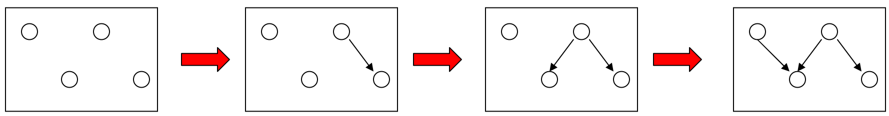
Advantage:

– we do not have to compute the whole score from scratch

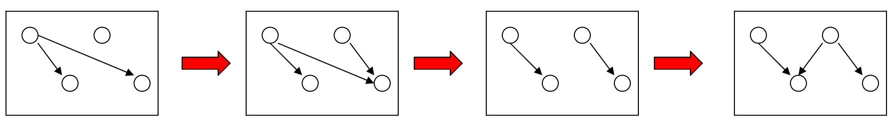
– Recompute the partial score for the affected variable

Optimizing the structure. Algorithms

- **Greedy search**
  - Start from the structure with no links
  - Add a link that yields the best score improvement



- **Metropolis algorithm (with simulated annealing)**
  - Local additions and deletions
  - Avoids being trapped in “local” optimal



Dimensionality reduction
Feature selection

Dimensionality reduction. Motivation.

- **Is there a lower dimensional representation of the data that captures well its characteristics?**
- **Assume:**
  - We have an data $\{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N\}$ such that
    $\mathbf{x}_i = (x_i^1, x_i^2, .., x_i^d)$
  - Assume the dimension d of the data point $\boldsymbol{x}$ is very large
  - We want to analyze $\boldsymbol{x}$
- **Methods of analysis are sensitive to the dimensionality *d***
- **Our goal:**
  - **Find a lower dimensional representation of data d'**

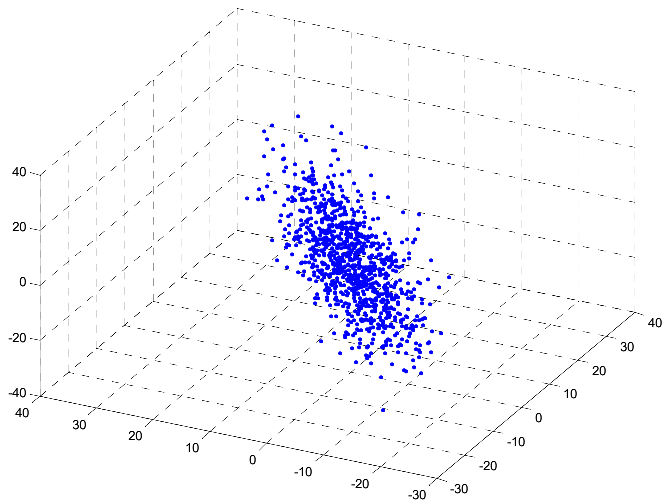
Principal component analysis (PCA)

- **Objective:** We want to replace a high dimensional input with a small set of features (obtained by combining inputs)

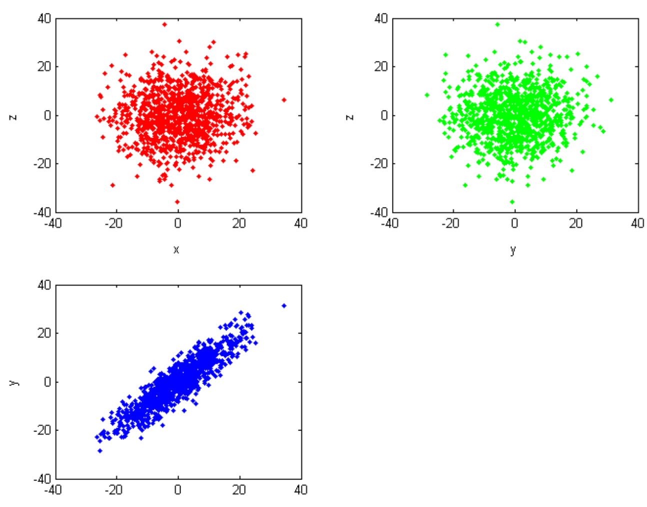
- **PCA:**
  - A linear transformation of d dimensional input x to M dimensional feature vector z such that $M < d$ under which the retained variance is maximal.

  - Equivalently it is the linear projection for which the sum of squares reconstruction cost is minimized.

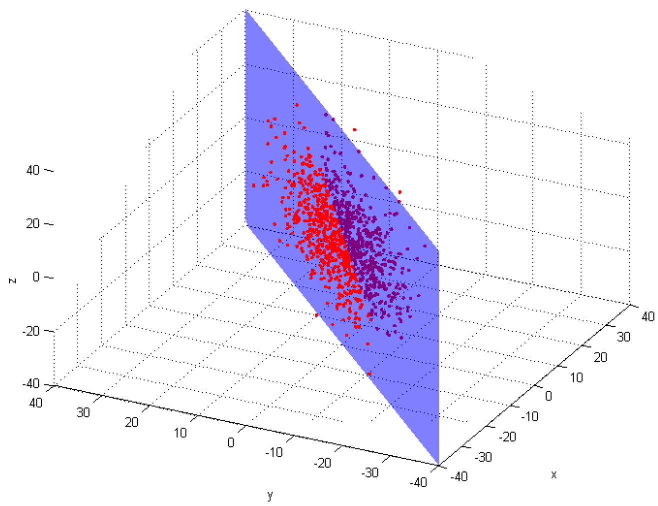
PCA



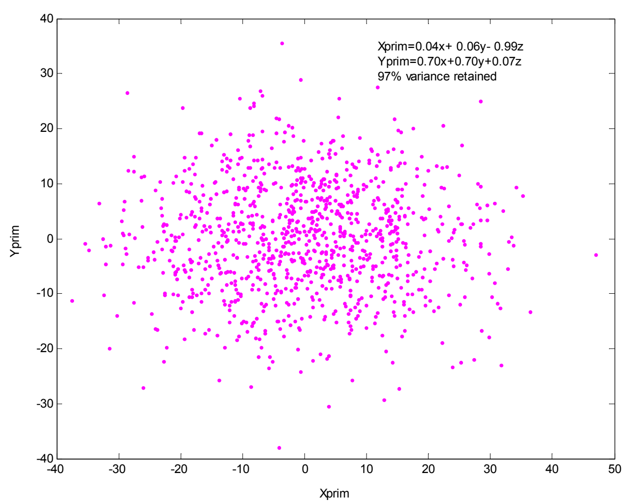
PCA



PCA


z
y
x

PCA


Xprim=0.04x+ 0.06y- 0.99z
Yprim=0.70x+0.70y+0.07z
97% variance retained
Yprim
Xprim

Principal component analysis (PCA)

- **PCA:**
  - linear transformation of d dimensional input x to M dimensional feature vector z such that $M < d$ under which the retained variance is maximal.
  - Task independent
- **Fact:**
  - A vector x can be represented using a set of orthonormal basis vectors u

$$\mathbf{x} = \sum_{i=1}^{d} z_i \mathbf{u}_i$$

  - Leads to transformation of coordinates (from x to z using u's)

$$z_i = \mathbf{u}_i^T \mathbf{x}$$

PCA

- **Idea:** replace d coordinates with M of z_i coordinates to represent x. We want to find the subset M of basis vectors.

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} z_i \mathbf{u}_i + \sum_{i=M+1}^{d} b_i \mathbf{u}_i$$

b_i - constant and fixed

- **How to choose the best set of basis vectors?**
  - We want the subset that gives the best approximation of data x in the dataset on average (we use least squares fit)

Error for data entry $\mathbf{x}^n$ $\quad \mathbf{x}^n - \tilde{\mathbf{x}}^n = \sum_{i=M+1}^{d} (z_i^n - b_i)\mathbf{u}_i$

$$E_M = \frac{1}{2}\sum_{n=1}^{N} \left\| \mathbf{x}^n - \tilde{\mathbf{x}}^n \right\| = \frac{1}{2}\sum_{n=1}^{N} \sum_{i=M+1}^{d} (z_i^n - b_i)^2$$

PCA

- **Differentiate the error function** with regard to all b_i and set equal to 0 we get:

$$b_i = \frac{1}{N}\sum_{n=1}^{N} z_i^n = \mathbf{u}_i^T \overline{\mathbf{x}} \qquad \overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^n$$

- Then we can rewrite:

$$E_M = \frac{1}{2}\sum_{i=M+1}^{d} \mathbf{u}_i^T \boldsymbol{\Sigma} \mathbf{u}_i \qquad \boldsymbol{\Sigma} = \sum_{n=1}^{N} (\mathbf{x}^n - \overline{\mathbf{x}})(\mathbf{x}^n - \overline{\mathbf{x}})^T$$

- The error function is optimized when basis vectors satisfy:

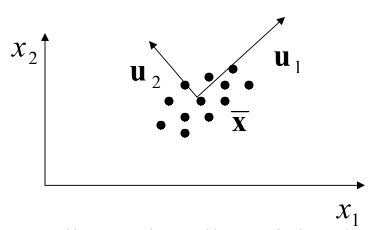
$$\boldsymbol{\Sigma}\mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad E_M = \frac{1}{2}\sum_{i=M+1}^{d} \lambda_i$$

The best *M* basis vectors: discard vectors with *d-M* smallest eigenvalues (or keep vectors with M largest eigenvalues)

Eigenvector $\mathbf{u}_i$ – is called a **principal component**

PCA

- Once eigenvectors $\mathbf{u}_i$ with largest eigenvalues are identified, they are used to transform the original *d*-dimensional data to *M* dimensions



- To find the "true" dimensionality of the data *d'* we can just look at eigenvalues that contribute the most (small eigenvalues are disregarded)
- **Problem:** PCA is a linear method. The "true" dimensionality can be overestimated. There can be non-linear correlations.