Introduction to Machine Learning NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká
hladka@ufal.mff.cuni.cz

Martin Holub
holub@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Lecture #12

Outline

- Principal Component Analysis
- Naïve Bayes algorithm
- Bayesian networks

Principal Component Analysis

PCA is

- a tool to analyze the data
- a tool to do dimensionality reduction

Basic concepts needed

- data analysis
  measures of center and spread, covariance and correlation
- linear algebra
  eigenvectors, eigenvalues, matrices, dot product, basis

Data analysis

How two variables are related

Both covariance and correlation indicate how closely two variables relationship follows a straight line.

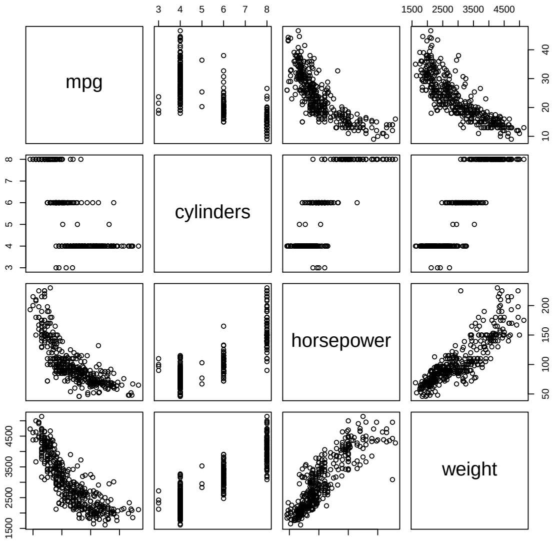
Covariance $\mathrm{cov}(X, Y)$ is a measure of the joint variability of two random variables X and Y

$$\mathrm{cov}(X, Y) = E[(X - EX)(Y - EY)]$$

The magnitude of the covariance is not easy to interpret because it is not normalized and hence depends on the magnitudes of the variables.

- > 0 both variables increase or decrease together
- < 0 while one variable increases the other decreases
- $= 0$ variables are linearly independent of each other

Auto data set



Data analysis

Covariance matrix of features $\mathrm{A}_1, \ldots, \mathrm{A}_m$

$$\mathbf{C}(\mathrm{A}_1, \ldots, \mathrm{A}_m) = \begin{pmatrix} \mathrm{var}(\mathrm{A}_1) & \mathrm{cov}(\mathrm{A}_1, \mathrm{A}_2) & \ldots & \mathrm{cov}(\mathrm{A}_1, \mathrm{A}_m) \\ \mathrm{cov}(\mathrm{A}_2, \mathrm{A}_1) & \mathrm{var}(\mathrm{A}_2) & \ldots & \mathrm{cov}(\mathrm{A}_2, \mathrm{A}_m) \\ \ldots & \ldots & \ldots & \ldots \\ \mathrm{cov}(\mathrm{A}_m, \mathrm{A}_1) & \mathrm{cov}(\mathrm{A}_m, \mathrm{A}_2) & \ldots & \mathrm{var}(\mathrm{A}_m) \end{pmatrix}$$

- diagonal - variance of the features $\mathrm{var}(\mathrm{A}_i)$
- symmetrical about the diagonal $\mathrm{cov}(\mathrm{A}_i, \mathrm{A}_j) = \mathrm{cov}(\mathrm{A}_j, \mathrm{A}_i)$

Data analysis
Auto data set

```
> cov(Auto[c("mpg", "cylinders", "horsepower", "weight")])

#                  mpg   cylinders  horsepower     weight
# mpg          60.91814  -10.352928  -233.85793  -5517.441
# cylinders   -10.35293    2.909696    55.34824   1300.424
# horsepower -233.85793   55.348244  1481.56939  28265.620
# weight    -5517.44070 1300.424363 28265.62023 721484.709

> cor(Auto[c("mpg", "cylinders", "horsepower", "weight")])

#                  mpg  cylinders horsepower     weight
# mpg          1.0000000 -0.7776175 -0.7784268 -0.8322442
# cylinders   -0.7776175  1.0000000  0.8429834  0.8975273
# horsepower -0.7784268  0.8429834  1.0000000  0.8645377
# weight      -0.8322442  0.8975273  0.8645377  1.0000000
```

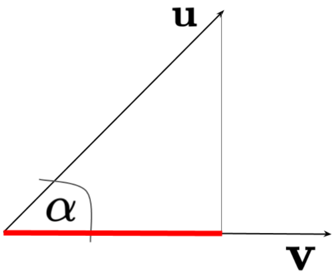
Linear algebra

1. $\mathbf{A}$ is a linear transformation. **Eigenvector** of $\mathbf{A}$ is a vector $\mathbf{u}$ for which exists **eigenvalue** λ so that $\mathbf{A} \cdot \mathbf{u} = \lambda \mathbf{u}$
   - eigenvector $\mathbf{u}$ does not change its direction under the transformation $\mathbf{A}$
   - $\lambda \mathbf{u}$ scales a vector $\mathbf{u}$ by λ; it changes its length, not its direction
2. The covariance matrix of $\mathbf{X}$ is an $m \times m$ symmetric matrix $\mathbf{C}(\mathbf{X}) = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\top$
3. Any symmetric matrix $m \times m$ $\mathbf{A}$ has a set of orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ associated with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$
   - for any i, $\mathbf{A} \cdot \mathbf{v}_i = \lambda_i \mathbf{v}_i$
   - $||\mathbf{v}_i|| = 1$
   - $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ if $i \neq j$
4. $\mathbf{A}$ is a symmetric $m \times m$ matrix and $\mathbf{E}$ is an $m \times m$ matrix whose i-th column is the i-th eigenvector of $\mathbf{A}$. The eigenvectors are ordered in terms of decreasing values of their associated eigenvalues. Then there is a diagonal matrix $\mathbf{D}$ such that $\mathbf{A} = \mathbf{E} \cdot \mathbf{D} \cdot \mathbf{E}^\top$
5. If the rows of $\mathbf{E}$ are orthogonal, then $\mathbf{E}^{-1} = \mathbf{E}^\top$

Linear algebra

Dot product

- $\mathbf{u} = \langle u_1, \ldots, u_m \rangle$, $\mathbf{v} = \langle v_1, \ldots, v_m \rangle$
- algebraic definition $\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + \cdots + u_m v_m$
- geometric definition $\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}|| \cdot ||\mathbf{v}|| \cdot \cos\alpha$
- $\mathbf{u}$ and $\mathbf{v}$ are orthogonal iff $\mathbf{u} \cdot \mathbf{v} = 0$



$$||\mathbf{v}|| = 1 \rightarrow \mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}|| \cdot \cos\alpha$$

Linear algebra

- A set of vectors $\mathbf{x}_i \in \mathcal{R}^m$ is linearly independent if no vector is a linear combination of other vectors.

Basis of $\mathcal{R}^m$ is a set vectors $\mathbf{u}_1, \dots, \mathbf{u}_m$

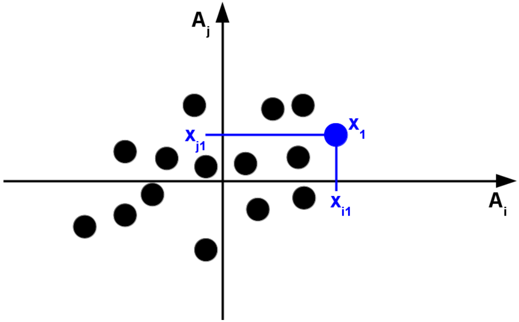
- linearly independent
- $\mathbf{u}_i \cdot \mathbf{u}_j = 0$, $i, j = 1, \dots, m$, $i \neq j$
- any $\mathbf{u} \in \mathcal{R}^m$: $\mathbf{u} = c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m$
- for example, the standard basis of the *3*-dimensional Euclidean space $\mathcal{R}^3$ consists of $\mathbf{x} = \langle 1, 0, 0\rangle, \mathbf{y} = \langle 0, 1, 0\rangle, \mathbf{z} = \langle 0, 0, 1\rangle$. It is an example of orthonormal basis, so called *naive* basis $\mathbf{I}$

Principal Component Analysis

$Data = \{\mathbf{x}_i, \mathbf{x}_i = \langle x_{1i}, \ldots, x_{mi} \rangle\}$, $|Data| = n$

$$\mathbf{X} = \begin{pmatrix} x_{11} & \ldots & x_{1n} \\ x_{21} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots \\ x_{m1} & \ldots & x_{mn} \end{pmatrix}$$

(i.e. examples in columns)



PCA

Which features to keep?

- features that change a lot, i.e. high variance
- features that do not depend on others, i.e. low covariance

Which features to ignore?

- features with some noise, i.e. low variance

$\mathbf{C}(\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_\mathrm{m})$

- on the diagonal, large values correspond to interesting structure
- off the diagonal, large values correspond to high redundancy

- high correlation $\sim$ high redundancy
- the most important feature has the largest variance

PCA

- **Question**

  Is there any other representation of $\mathbf{X}$ to extract the most important features?

- **Answer**

  Use another basis

$$\mathbf{P}^\top \cdot \mathbf{X} = \mathbf{Z}$$

  where $\mathbf{P}$ transforms $\mathbf{X}$ into $\mathbf{Z}$; $\mathbf{Z}$ is a new representation of $\mathbf{X}$

PCA
Heading for P

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_{11} & \cdots & \cdots & \mathbf{p}_{1m} \\ \mathbf{p}_{21} & \cdots & \cdots & \mathbf{p}_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{p}_{m1} & \cdots & \cdots & \mathbf{p}_{mm} \end{pmatrix}$$

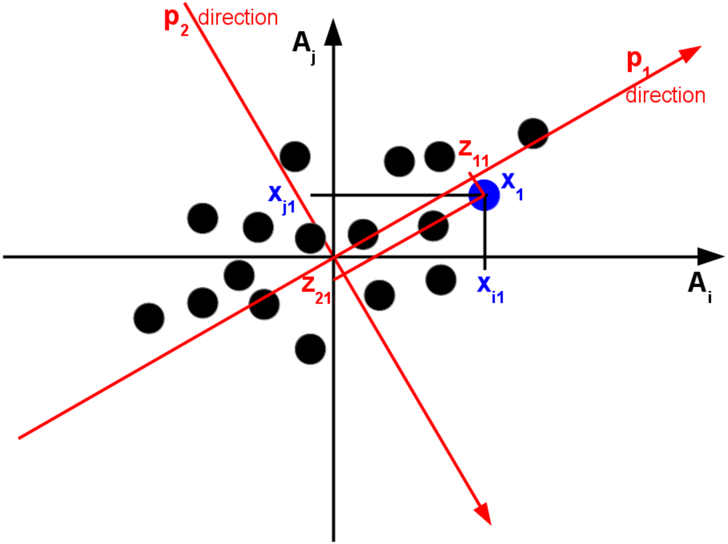
- **principal components** of $\mathbf{X}$ are the vectors $\mathbf{p}_i = \langle p_{1i}, \ldots, p_{mi} \rangle$
- **principal component loadings** of $\mathbf{p}_i$ are the elements $p_{i1}, \ldots, p_{im}$

PCA
Heading for P

$$\mathbf{Z} = \begin{pmatrix} \mathbf{p}_1 \cdot \mathbf{x}_1 & \dots & \dots & \mathbf{p}_1 \cdot \mathbf{x}_n \\ \mathbf{p}_2 \cdot \mathbf{x}_1 & \dots & \dots & \mathbf{p}_2 \cdot \mathbf{x}_n \\ \dots & \dots & \dots & \dots \\ \mathbf{p}_m \cdot \mathbf{x}_1 & \dots & \dots & \mathbf{p}_m \cdot \mathbf{x}_n \end{pmatrix}$$

i-**principal component scores** of *n* instances are $\mathbf{p}_i \cdot \mathbf{x}_1, \mathbf{p}_i \cdot \mathbf{x}_2, \dots, \mathbf{p}_i \cdot \mathbf{x}_n$

PCA
Heading for P



PCA
Heading for P

- What is a good choice of **P**?
- What features we would like **Z** to exhibit?

Goal: Find a set of directions on which to project the data such that

- the variance of each projection is maximized
- the projections are uncorrelated (random variables X, Y are said to be uncorrelated if their $\mathrm{cov}(X, Y) = 0$)

PCA
Heading for P

Let's compute the variance of a random variable obtained by projecting **X** onto a direction represented by the vector **p** ($\mu = E[\mathbf{X}]$):

$$\sigma^2 = E[(\mathbf{p}^\top \mathbf{X} - E[\mathbf{p}^\top \mathbf{X}])^2] = \mathbf{p}^\top E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^\top]\mathbf{p} = \mathbf{p}^\top \mathbf{C}(\mathbf{X})\mathbf{p}$$

We use the method of Lagrange multipliers:

Maximize

$$\mathbf{p}^\top \mathbf{C}(\mathbf{X})\mathbf{p}$$

subject to

$$\mathbf{p}^\top \mathbf{p} = 1$$

PCA
Heading for P

Lagrangian function

$$\mathcal{L}(\mathbf{p}, \lambda) = \mathbf{p}^\top \mathbf{C}(\mathbf{X})\mathbf{p} - \lambda \mathbf{p}^\top \mathbf{p}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}} = 0 \Rightarrow \mathbf{C}(\mathbf{X})\mathbf{p} = \lambda \mathbf{p}$$

Our problem comes down to seeking eigenvalues and eigenvectors of $\mathbf{C}(\mathbf{X})$.
In the general case, $\mathbf{C}(\mathbf{X})$ has *m* distinct eigenvectors and eigenvalues.

Which one is the solution we seek?

$$\sigma^2 = \mathbf{p}^\top \mathbf{C}(\mathbf{X})\mathbf{p} = \mathbf{p}^\top \lambda \mathbf{p} = \lambda \mathbf{p}^\top \mathbf{p} = \lambda$$

The variance is maximized if we choose the unit eigenvector that corresponds to the largest eigenvalue of $\mathbf{C}(\mathbf{X})$. Denote these as $\mathbf{p}_1$, λ_1.

Usually we cannot represent the data sufficiently good with just one projection. Thus, we need to find the procedure for computing the next projection directions $\mathbf{p}_2$, λ_2, $\mathbf{p}_3$, λ_3, $\ldots$

PCA
Heading for P

- principal components are new basis vectors to represent $\mathbf{x}_j$, $j = 1, \ldots, n$
- $\mathbf{p}_i \cdot \mathbf{x}_j$ is a projection of $\mathbf{x}_j$ on $\mathbf{p}_i$
- changing the basis does not change data, it changes their representation

Derivation of PCA

1. preprocessing *Data*
   mean normalization to get centered data $\rightarrow \mathbf{X}$
2. $\mathbf{C}(\mathbf{X}) = \mathbf{A} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\top$
3. Compute eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ and eigenvalues $\lambda_1, \dots, \lambda_m$ of $\mathbf{A}$
4. Take the eigenvectors, order them by eigenvalues, i.e. by significance, highest to lowest: $\mathbf{p}_1, \dots, \mathbf{p}_m, \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$
5. The eigenvectors $\mathbf{p}_1, \dots, \mathbf{p}_m$ become columns of $\mathbf{P}$

$$\boldsymbol{p}_i = \begin{pmatrix} p_{1i} \\ \dots \\ p_{mi} \end{pmatrix}$$

Properties of PCA

$$\mathbf{P}^\top \cdot \mathbf{X} = \mathbf{Z}$$

$$\mathbf{Z} = \begin{pmatrix} \mathbf{p}_1 \cdot \mathbf{x}_1 & \dots & \dots & \mathbf{p}_1 \cdot \mathbf{x}_n \\ \mathbf{p}_2 \cdot \mathbf{x}_1 & \dots & \dots & \mathbf{p}_2 \cdot \mathbf{x}_n \\ \dots & \dots & \dots & \dots \\ \mathbf{p}_m \cdot \mathbf{x}_1 & \dots & \dots & \mathbf{p}_m \cdot \mathbf{x}_n \end{pmatrix}$$

- The i-th diagonal value of $\mathbf{C}(\mathbf{Z})$ is the variance of $\mathbf{X}$ along $\mathbf{p_i}$.
- We calculate a rotation of the original coordinate system such that all non-diagonal elements of the new covariance matrix become zero.
- The principal components define the basis of the new coordinate axes and the eigenvalues correspond to the diagonal elements of the new covariance matrix.
- So the eigenvalues, by definition, define the variance along the corresponding principal components.

Properties of PCA

$$\mathbf{C}(\mathbf{P}^\top \cdot \mathbf{X}) \overset{\text{see } \underline{\text{p.29.4}}}{=} \frac{1}{n-1}(\mathbf{P}^\top \cdot \mathbf{X}) \cdot (\mathbf{P}^\top \cdot \mathbf{X})^\top =$$

$$\frac{1}{n-1}\mathbf{P}^\top \cdot \mathbf{X} \cdot \mathbf{X}^\top \cdot \mathbf{P} \overset{\text{let } \mathbf{A}=\mathbf{X}\cdot\mathbf{X}^\top}{=} \frac{1}{n-1}\mathbf{P}^\top \cdot \mathbf{A} \cdot \mathbf{P} =$$

$$\overset{\text{see } \underline{\text{p.29.4}}}{=} \frac{1}{n-1}\mathbf{P}^\top \cdot (\mathbf{P} \cdot \mathbf{D} \cdot \mathbf{P}^\top) \cdot \mathbf{P} \overset{\text{see } \underline{\text{p.29.5}}}{=} \frac{1}{n-1}\mathbf{P}^\top \cdot (\mathbf{P}^\top)^{-1} \mathbf{D} \cdot \mathbf{P}^\top \cdot (\mathbf{P}^\top)^{-1} = \frac{1}{n-1}\mathbf{D}$$

Properties of PCA

A geometric interpretation for the first principal component $\mathbf{p}_1$

It defines a direction in feature space along which the data vary the most. If we project the n instances $\mathbf{x}_1, \dots, \mathbf{x}_n$ onto this direction, the projected values are the principal component scores $z_{11}, \dots, z_{n1}$ themselves.

Proportion of Variance Explained (PVE)

How much of the information in a given data set is lost by projecting the instances onto the first few principal components?

In other words, how much of the variance in the data is not contained in the first few principal components?

- total variance in **X**: $\sum_{j=1}^{m} \mathrm{var}(A_j) = \sum_{i=1}^{m} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$ (assuming feature normalization)
- variance expressed by $\mathbf{p}_k$: $\frac{1}{n} \sum_{i=1}^{n} z_{ki}^2$
- $\mathrm{PVE}(\mathbf{p}_k) = \frac{\sum_{i=1}^{n} z_{ki}^2}{\sum_{i=1}^{m} \sum_{i=1}^{n} x_{ij}^2}$
- $\mathrm{PVE}(\mathbf{p}_1, \ldots, \mathbf{p}_M) = \sum_{i=1}^{M} \mathrm{PVE}(\mathbf{p}_i)$, $M \leq m$

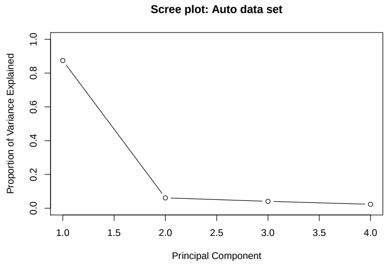
PCA
Auto data set

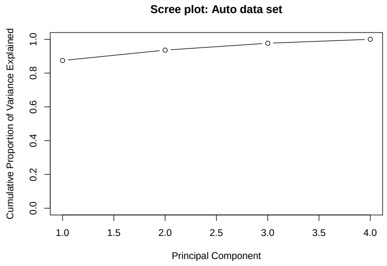
```
> a <- Auto[c("mpg", "cylinders", "horsepower", "weight")]
> pca.a <- prcomp(a, scale = TRUE)
> summary(pca.a)

# Importance of components:
#                          Comp.1    Comp.2   Comp.3   Comp.4
Standard deviation     1.8704 0.49540 0.40390 0.30518
Proportion of Variance 0.8746 0.06135 0.04078 0.02328
Cumulative Proportion  0.8746 0.93593 0.97672 1.00000
```

PCA
Auto data set

Scree plot





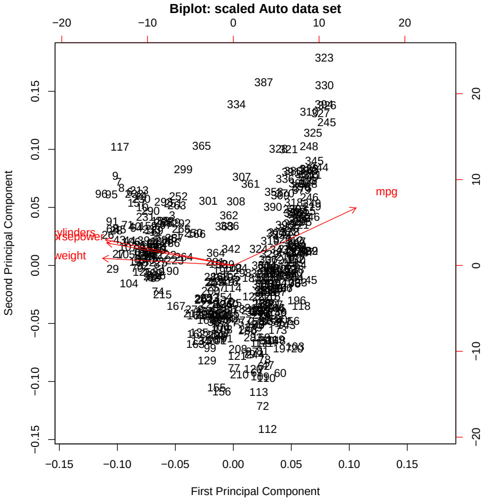
PCA
Auto data set

```
> pca.a$rotation
                  PC1       PC2         PC3        PC4
mpg         0.4833271 0.8550485 -0.02994982  0.1854453
cylinders  -0.5033993 0.3818233 -0.55748381 -0.5385276
horsepower -0.4984381 0.3346173  0.79129092 -0.1159714
weight     -0.5143380 0.1055192 -0.24934614  0.8137252
```

- PC1 places approximately equal weight on `cylinders`, `horsepower`, `weight` with much higher weight on `mpg`.
- PC2 places most of its weight on `mpg` and less weight on the other three features.

Biplot for the Auto data set is showing

A biplot displays both the PC scores and the PC loadings.



The biplot for the Auto data set is showing

- the scores of each example (i.e., cars) on the first two principal components with axes on the top and right
  - see the id cars in black
- the loading of each feature (i.e., `mpg`, `weight`, `cylinders`, `horsepower`) on the first two principal components with axes on the bottom and left
  - see the red arrows
  - their length corresponds to the variabliaty of the original features

```
> a <- Auto[c("mpg", "cylinders", "horsepower", "weight")]
> apply(a, 2, var)
         mpg    cylinders   horsepower       weight
  6.091814e+01 2.909696e+00 1.481569e+03 7.214847e+05
```

PCA

In general, a $m \times n$ matrix $\mathbf{X}$ has $\min(n-1, m)$ distinct principal components.

- **Question**
  How many principal components are needed?
- **Answer**
  There is no single answer to this question. Study scree plots.

Probability vs. likelihood

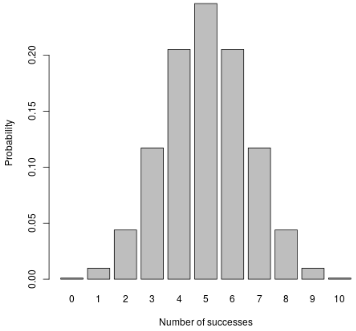
Task: Predict the outcome of each of 10 coin tosses

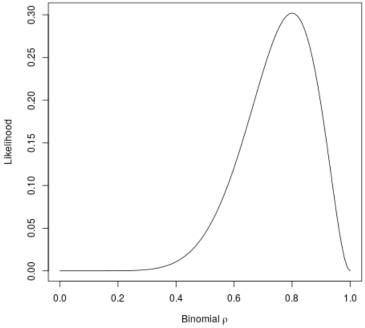
probability

$$\Pr(X = k | n = 10, p = 0.8)$$
$$\Pr(\mathrm{data}|\theta)$$

likelihood

$$\mathcal{L}(p | X = 8)$$
$$\mathcal{L}(\theta|\mathrm{data})$$





Bayes theorem

Probabilistic approach to classification $Y = \{y_{1,2}, \dots, y_K\}$

$$y^\star = \mathrm{argmax}_{y_k \in Y} \Pr(y_k | x_1, \dots, x_m) \tag{1}$$

Bayes theorem

$$\text{posterior probability} = \frac{\text{prior probability} \times \text{likelihood}}{\text{marginal likelihood}} \tag{2}$$

$$\Pr(Y \mid A_1, \dots, A_m) = \frac{\Pr(Y) \times \Pr(A_1, \dots, A_m \mid Y)}{\Pr(A_1, \dots, A_m)}$$

Then

$$y^\star = \mathrm{argmax}_{y_k \in Y} \frac{\Pr(y_k) \times \Pr(x_1, \dots, x_m | y_k)}{\Pr(x_1, \dots, x_m)} \tag{3}$$

Conditional independence

Let X, Y and Z be three descrete random variables. We say that X is *conditionally independent* of Y given Z if

$\forall x_i, y_j, z_k, x_i \in Values(X), y_j \in Values(Y), z_k \in Values(Z):$

$$\Pr(X = x_i | Y = y_j, Z = z_k) = \Pr(X = x_i | Z = z_k) \tag{4}$$

I.e., $\mathrm{P}(X|Y, Z) = \mathrm{P}(X|Z)$.

Conditional independence

Do we enjoy our favorite water sport on this day? (Credit: T. Mitchel, 1997)

Sky	AirTemp	Humidity	Wind	EnjoySport
sunny	warm	normal	strong	No
sunny	warm	high	strong	Yes
rainy	cold	high	strong	No
sunny	warm	high	strong	Yes

Conditional independence of features given *EnjoySport*: presence of one particular feature value does not affect the other features' values given *EnjoySport*, e.g., if the temperature is hot, it does not necessarily mean that the humidity is high and the features have an equal effect on the outcome

Conditional independence

If we work with two features A_1, A_2 and we assume that they are conditionally independent given the target class Y, then

$$\Pr(A_1, A_2|Y) \overset{\text{product rule}}{=} \Pr(A_1|A_2, Y) * \Pr(A_2|Y) \overset{\text{c. i. assumption}}{=} \Pr(A_1|Y) * \Pr(A_2|Y)$$

Note: Product rule (a.k.a. Chain rule)

$$\Pr(A_m, \dots, A_1) = \Pr(A_m|A_{m-1}, \dots, A_1) \cdot \Pr(A_{m-1}, \dots, A_1)$$

Naïve Bayes classifier

$$y^\star = \mathrm{argmax}_{y_k \in Y} \Pr(y_k|x_1, \ldots, x_m) = \mathrm{argmax}_{y_k \in Y} \frac{\Pr(y_k) \Pr(x_1, \ldots, x_m|y_k)}{\Pr(x_1, \ldots, x_m)}$$

– Assume conditional independence of features $A_1, \ldots, A_m$ given Y. Then

$$\Pr(x_1, x_2, \ldots, x_m|y_k) \overset{\text{product rule}}{=} \prod_{j=1}^m \Pr(x_j|x_1, x_2, \ldots, x_{j-1}, y_k) \overset{\text{c. i. a.}}{=}$$

$$= \prod_{j=1}^m \Pr(x_j|y_k)$$

– $\Pr(x_1, \ldots, x_m)$ is constant. Then

$$y^\star = \mathrm{argmax}_{y_k \in Y} \Pr(y_k) \prod_{j=1}^{m} \Pr(x_j|y_k) \tag{5}$$

Discriminative vs. generative classifiers

Computing $\Pr(y|\mathbf{x})$

- **discriminative classifier** does not care about how the data was generated. It directly discriminates the value of y for any $\mathbf{x}$.
- **generative classifier** models how the data was generated in order to classify an example.

Discriminative vs. generative classifiers

- Logistic regression classifier is a **discriminative classifier**

$$f(\mathbf{x}; \Theta) = p(y = 1|\mathbf{x}, \Theta)$$

- Naïve Bayes classifier is a **generative classifier**

1. Learn $\Pr(\mathbf{x}|y)$ and $\Pr(y)$
2. Apply Bayes rule to get

$$\Pr(y|\mathbf{x}) = \frac{\Pr(\mathbf{x}|y)\Pr(y)}{\Pr(\mathbf{x})} \sim \Pr(\mathbf{x}|y)\Pr(y)$$

3. Classify $\mathbf{x}$

$$y^\star = \mathrm{argmax}_y \Pr(y|\mathbf{x}) = \mathrm{argmax}_y \Pr(\mathbf{x}|y)\Pr(y)$$

Naïve Bayes classifier

Naive assumption of feature conditional independence given a target class is rarely true in real world applications (high bias). Nevertheless, Naïve Bayes classifier surprisingly often shows good performance in classification (low variance).

Naïve Bayes Classifier is a linear classifier

NB classifier gives a method for predicting rather than for building an explicit classifier.

Let us focus on **binary classification** $Y = \{0, 1\}$ with binary features $A_1, \dots, A_m$.

We predict 1 iff

$$\frac{\Pr(y=1)\prod_{j=1}^{m}\Pr(x_j|y=1)}{\Pr(y=0)\prod_{j=1}^{m}\Pr(x_j|y=0)} > 1$$

Naïve Bayes Classifier is a linear classifier

Denote $p_j = \Pr(x_j = 1|y = 1)$, $q_j = \Pr(x_j = 1|y = 0)$

Then

$$\frac{\Pr(y=1)\prod_{j=1}^{m} p_j^{x_j}(1-p_j)^{1-x_j}}{\Pr(y=0)\prod_{j=1}^{m} q_j^{x_j}(1-q_j)^{1-x_j}} > 1$$

$$\frac{\Pr(y=1)\prod_{j=1}^{m}(1-p_j)(\frac{p_j}{1-p_j})^{x_j}}{\Pr(y=0)\prod_{j=1}^{m}(1-q_j)(\frac{q_j}{1-q_j})^{x_j}} > 1$$

Naïve Bayes Classifier is a linear classifier

Take logarithm

$$\log \frac{\Pr(y=1)}{\Pr(y=0)} + \sum_{j=1}^{m} \log \frac{1-p_j}{1-q_j} + \sum_{j=1}^{m} (\log \frac{p_j}{1-p_j} - \log \frac{q_j}{1-q_j}) x_j > 0$$

NB classifier as a linear classifier where

$$\theta_0 = \log \frac{\Pr(y=1)}{\Pr(y=0)} + \sum_{j=1}^{m} \log \frac{1-p_j}{1-q_j}$$

$$\theta_j = \log \frac{p_j}{1-p_j} - \log \frac{q_j}{1-q_j}, \quad j = 1, \dots, m$$

Bayesian belief networks (BBN)

- Naïve Bayes classifier assumes that ALL features are conditionally independent given a target attribute.
- A Bayesian network is a probabilistic graphical model that encodes probabilistic relationships among attributes of interest.
- BBNs allow stating conditional independence assumptions that apply to subsets of the attributes.
- Dependencies are modeled as graph where nodes correspond to attributes and edges to dependency between attributes.

Bayesian belief networks
Settings

Consider an arbitrary set of random variables $X_1, X_2, ..., X_m$. Each variable X_i can take on the set of possible values $Values(X_i)$.

We define the **joint space** of the variables $X_1, X_2, ..., X_m$ to be the cross product $Values(X_1) \times Values(X_2) \times Values(X_3) \times ... \times Values(X_m)$.

The probability distribution over the joint space is called the **joint probability distribution** $\Pr(x_1, x_2, ..., x_m)$ where
$x_1 \in Values(X_1), x_2 \in Values(X_2), ..., x_n \in Values(X_m)$.

BBN describes the joint probability distribution for a set of variables by specifying a set of conditional independence assumptions together with sets of local conditional probabilities.

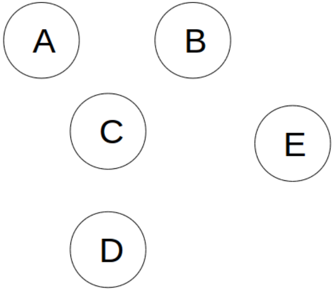
Bayesian belief networks

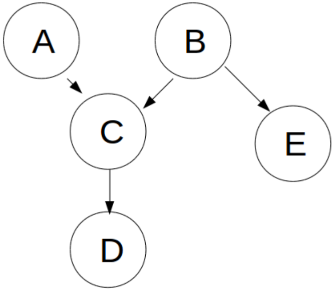
Representation

1. A directed acyclic graph $G = (V, E)$
   - nodes are random variables
   - arcs between nodes represent probabilistic dependencies
   - Y is a *descendant* of X if there is a directed path from X to Y
2. The network arcs represent the assertion that the variable X is conditionally independent of its nondescendants given its immediate predecessors $Parents(X)$; $\Pr(X|Parents(X))$
3. A set of tables for each node in the graph - a conditional probability table is given for each variable; it describes the probability distribution for that variable given the values of its immediate predecessors.

Building a Bayes net

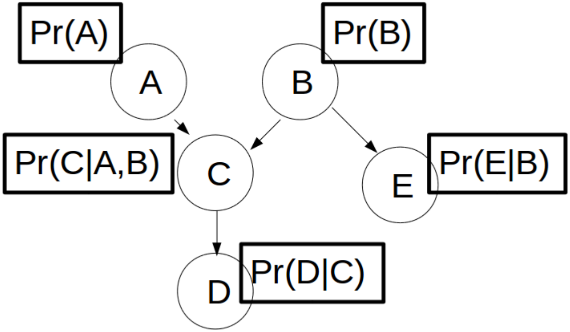
1. Choose the variables to be included in the net: A, B, C, D, E
2. Add the links





Building a Bayes net

3. Add a probability table for each root node $\Pr(X)$ and nonroot node $\Pr(X|\mathit{Parents}(X))$



Once the net is built ...

The join probability of any assignment of values $x_1, x_2, ..., x_m$ to the tuple of network variables $X_1, X_2, ..., X_m$ can be computed by the formula

$$\Pr(x_1, x_2, ..., x_m) = \Pr(X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_m = x_m) = \prod_{i=1}^{m} \Pr(x_i | Parents(X_i)) \tag{6}$$

Bayesian belief networks

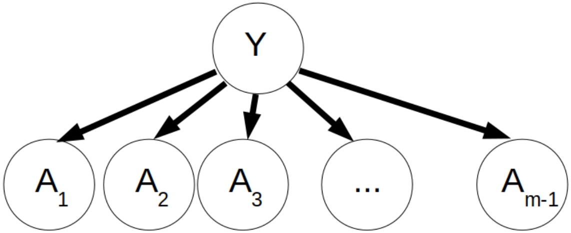
Two components

1. A function for evaluating a given network based on the data.
2. A method for searching through the space of possible networks.

Learning the network structure

- searching through the space of possible sets of edges
- estimating the conditional probability tables for each set
- computing the quality of the network

Bayesian belief networks
Naïve Bayes Classifier



K2 algorithm

This 'search and score' algorithm heuristically searches for the most probable belief-network structure given a training data.

It starts by assuming that a node has no parents, after which, in every step it adds incrementally the parent whose addition mostly increase the probability of the resulting structure. K2 stops adding parents to the nodes when the addition of a single parent cannot increase the probability of the network given the data.

Summary of Examination Requirements

- Discriminative and generative classifiers
- Naïve Bayes Classifier
  conditional independence, linear decision boundary
- Bayesian networks
  structure, conditional probabilities
- Principal Component Analysis
  data analysis, derivation, scree plot, biplot