Introduction to Bayesian Learning

Davide Bacciu

Dipartimento di Informatica
Università di Pisa
bacciu@di.unipi.it

Apprendimento Automatico: Fondamenti - A.A. 2016/2017
Lecture 1 - Introduction
   - Probabilistic reasoning and learning
   - (Bayesian Networks)

Lecture 2 - Parameters Learning (I)
   - Learning fully observed Bayesian models

Lecture 3 - Parameters Learning (II)
   - Learning with hidden variables

If we have time, we will cover also some application examples of Bayesian learning and Bayesian Networks
Reference Webpage:

http://pages.di.unipi.it/bacchiu/teaching/apprendimento-automatico-fondamenti/

Here you can find

- Course information
- Lecture slides
- Articles and course materials

Office Hours

Where? Dipartimento di Informatica - 2nd Floor - Room 367

When? Monday 17-19

When? Basically anytime, if you send me an email beforehand.
Lecture Outline

1. Introduction

2. Probability Theory
   - Probabilities and Random Variables
   - Bayes Theorem and Independence

3. Learning as Bayesian Inference
   - Hypothesis Selection
   - Bayesian Classifiers

4. Bayesian Networks
   - Representation
   - Learning and Inference
Bayesian Learning

- **Why Bayesian?** Easy, because of frequent use of Bayes theorem...
- **Bayesian Inference** A powerful approach to probabilistic reasoning
- **Bayesian Networks** An expressive model for describing probabilistic relationships

- **Why bothering?**
  - Real-world is uncertain
    - Data (noisy measurements and partial knowledge)
    - Beliefs (concepts and their relationships)
  - Probability as a measure of our beliefs
    - Conceptual framework for describing uncertainty in world representation
    - Learning and reasoning become matters of probabilistic inference
    - Probabilistic weighting of the hypothesis
Part I

Probability and Learning
Random Variables

- A **Random Variable (RV)** is a function describing the outcome of a **random process** by assigning unique values to all possible outcomes of the experiment.

  \[
  \text{Random Process} \quad \implies \quad \text{Coin Toss}
  \]

  \[
  \text{Discrete RV} \quad \implies \quad X = \begin{cases} 0 & \text{if heads} \\ 1 & \text{if tails} \end{cases}
  \]

- The **sample space** \( S \) of a random process is the set of all possible outcomes, e.g. \( S = \{\text{heads, tails}\} \).

- An **event** \( e \) is a subset \( e \in S \), i.e. a set of outcomes, that may occur or not as a result of the experiment.

**Random variables are the building blocks for representing our world.**
A probability function \( P(X = x) \in [0, 1] \) (\( P(x) \) in short) measures the probability of a RV \( X \) attaining the value \( x \), i.e. the probability of event \( x \) occurring.

If the random process is described by a set of RVs \( X_1, \ldots, X_N \), then the joint conditional probability writes

\[
P(X_1 = x_1, \ldots, X_N = x_n) = P(x_1 \land \cdots \land x_n)
\]

**Definition (Sum Rule)**

Probabilities of all the events must sum to 1

\[
\sum_x P(X = x) = 1
\]
Definition (Product Rule a.k.a. Chain Rule)

\[ P(x_1, \ldots, x_i, \ldots, x_n | y) = \prod_{i=1}^{N} P(x_i | x_1, \ldots, x_{i-1}, y) \]

- \( P(x|y) \) is the **conditional probability** of \( x \) given \( y \)
- Reflects the fact that the realization of an event \( y \) may affect the occurrence of \( x \)
- **Marginalization**: sum and product rules together yield the complete probability equation

\[
P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2) = \sum_{x_2} P(X_1 = x_1 | X_2 = x_2) P(X_2 = x_2)
\]
Bayes Rule

Given hypothesis \( h_i \in H \) and observations \( d \)

\[
P(h_i|d) = \frac{P(d|h_i)P(h_i)}{P(d)} = \frac{P(d|h_i)P(h_i)}{\sum_j P(d|h_j)P(h_j)}
\]

- \( P(h_i) \) is the prior probability of \( h_i \)
- \( P(d|h_i) \) is the conditional probability of observing \( d \) given that hypothesis \( h_i \) is true (likelihood).
- \( P(d) \) is the marginal probability of \( d \)
- \( P(h_i|d) \) is the posterior probability that hypothesis is true given the data and the previous belief about the hypothesis.
So, Is This All About Bayes???

Frequentists Vs Bayesians

(YET ANOTHER) HISTORY OF LIFE AS WE KNOW IT...

HOMO APRIORIUS  HOMO PRAGMATICUS  HOMO FREQUENTISTUS  HOMO SAPIENS  HOMO BAYESIANIS

Figure: Credit goes to Mike West @ Duke University
Independence and Conditional Independence

- Two RV \( X \) and \( Y \) are **independent** if knowledge about \( X \) does not change the uncertainty about \( Y \) and vice versa

\[
I(X, Y) \iff P(X, Y) = P(X|Y)P(Y) = P(Y|X)P(X) = P(X)P(Y)
\]

- Two RV \( X \) and \( Y \) are **conditionally independent** given \( Z \) if the realization of \( X \) and \( Y \) is an independent event of their conditional probability distribution given \( Z \)

\[
I(X, Y|Z) \iff P(X, Y|Z) = P(X|Y, Z)P(Y|Z) = P(Y|X, Z)P(X|Z) = P(X|Z)P(Y|Z)
\]
### Joint Probability Distribution Table

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$\ldots$</th>
<th>$X_i$</th>
<th>$\ldots$</th>
<th>$X_n$</th>
<th>$P(X_1, \ldots, X_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$\ldots$</td>
<td>$x_i$</td>
<td>$\ldots$</td>
<td>$x_n$</td>
<td>$P(x_1, \ldots, x_n)$</td>
</tr>
<tr>
<td>$x_1^l$</td>
<td>$\ldots$</td>
<td>$x_i^l$</td>
<td>$\ldots$</td>
<td>$x_n^l$</td>
<td>$P(x_1^l, \ldots, x_n^l)$</td>
</tr>
</tbody>
</table>

Describes $P(X_1, \ldots, X_n)$ for all the RV instantiations $x_1, \ldots, x_n$

In general, any probability of interest can be obtained starting from the **Joint Probability Distribution** $P(X_1, \ldots, X_n)$
Wrapping Up....

- We know how to represent the world and the observations
  - Random Variables $\rightarrow X_1, \ldots, X_N$
  - Joint Probability Distribution $\rightarrow P(X_1 = x_1, \ldots, X_N = x_n)$
- We have rules for manipulating the probabilistic knowledge
  - Sum-Product
  - Marginalization
  - Bayes
  - Conditional Independence
- It is about time that we do some learning
  - ...in other words, lets discover the values for $P(X_1 = x_1, \ldots, X_N = x_n)$
Statistical learning approaches calculate the probability of each hypothesis $h_i$ given the data $D$, and selects hypotheses/makes predictions on that basis.

**Bayesian learning** makes predictions using all hypotheses weighted by their probabilities:

$$P(X|D = d) = \sum_i P(X|d, h_i)P(h_i|d)$$

$$= \sum_i P(X|h_i) \cdot P(h_i|d)$$

New prediction  Hypothesis prediction  Posterior weighting
Computational and Analytical Tractability Issue

Bayesian Learning requires a (possibly infinite) summation over the whole hypothesis space.

- **Maximum a-Posteriori** (MAP) predicts $P(X|h_{MAP})$ using the most likely hypothesis $h_{MAP}$ given the training data.

  $$h_{MAP} = \arg \max_{h \in H} P(h|d) = \arg \max_{h \in H} \frac{P(d|h)P(h)}{P(d)}$$

- Assuming uniform priors $P(h_i) = P(h_j)$, yields the **Maximum Likelihood** (ML) estimate $P(X|h_{ML})$.

  $$h_{ML} = \arg \max_{h \in H} P(d|h)$$
Let’s go to the Cinema!!!

- How do I choose the next movie (prediction)?
- I might ask my friends for their favorite choice given their personal taste (hypothesis)
- Select the movie
  - Bayesian advice? Make a voting from all the friends’ suggestions weighted by their attendance to cinema and taste
  - MAP advice? From the friend who goes often to the cinema and whose taste I trust
  - ML advice? From the friend who goes more often to the cinema
A candy manufacturer produces 5 types of candy boxes (hypothesis) that are indistinguishable in the darkness of the cinema:

- $h_1$: 100% cherry flavor
- $h_2$: 75% cherry and 25% lime flavor
- $h_3$: 50% cherry and 50% lime flavor
- $h_4$: 25% cherry and 75% lime flavor
- $h_5$: 100% lime flavor

Given a sequence of candies $d = d_1, \ldots, d_N$ extracted and reinserted in a box (observations), what is the most likely flavor for the next candy (prediction)?
First, we need to compute the posterior for each hypothesis (Bayes)

$$P(h_i|d) = \alpha P(d|h_i)P(h_i)$$

The manufacturer is kind enough to provide us with the production shares (prior) for the 5 boxes

$$P(h_1), P(h_2), P(h_3), P(h_4), P(h_5) = (0.1, 0.2, 0.4, 0.2, 0.1)$$

Data likelihood can be computed under the assumption that observations are independently and identically distributed (i.i.d.)

$$P(d|h_i) = \prod_{j=1}^{N} P(d_j|h_i)$$
Suppose that the bag is a $h_5$ and consider a sequence of 10 observed lime candies.

$$P(h_i | d) = \alpha P(h_i) P(d = l|h_i)^N$$

<table>
<thead>
<tr>
<th>Hyp</th>
<th>$d_0$</th>
<th>$d_1$</th>
<th>$d_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$h_2$</td>
<td>0.2</td>
<td>0.1</td>
<td>0.03</td>
</tr>
<tr>
<td>$h_3$</td>
<td>0.4</td>
<td>0.4</td>
<td>0.30</td>
</tr>
<tr>
<td>$h_4$</td>
<td>0.2</td>
<td>0.3</td>
<td>0.35</td>
</tr>
<tr>
<td>$h_5$</td>
<td>0.1</td>
<td>0.2</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Posterior probability of $h_i$ eventually vanish

Posters of false hypothesis eventually vanish
Bayesian learning seeks

\[ P(d_{11} = l | d_1 = l, \ldots, d_{10} = l) = \sum_{i=1}^{5} P(d_{11} = l | h_i) P(h_i | d) \]
Both ML and MAP are point estimates since they only make predictions based on the most likely hypothesis.

MAP predictions are approximately Bayesian if $P(X|d) \sim P(X|h_{MAP})$

MAP and Bayesian predictions become closer as more data gets available.

ML is a good approximation to MAP if dataset is large and there are no a-priori preferences on the hypotheses:
- ML is fully data-driven
- For large data sets, the influence of prior becomes irrelevant
- ML has problems with small datasets
- $P(h)$ introduces **preference** across hypotheses
- **Penalize** complexity
  - Complex hypotheses have a lower prior probability
  - Hypothesis prior embodies trade-off between complexity and degree of fit
- MAP hypothesis $h_{MAP}$

$$\max_h P(d|h)P(h) \equiv \min_h -\log_2(P(d|h)) - \log_2 P(h)$$

Number of bits required to specify $h$

- MAP $\iff$ choosing the hypothesis that provides **maximum compression**
- MAP is a regularization of the ML estimation
Suppose we want to classify a new observation as one of the classes $c_j \in C$

The most probable Bayesian classification is

$$c_{OPT}^* = \arg \max_{c_j \in C} P(c_j | d) = \arg \max_{c_j \in C} \sum_i P(c_j | h_i) P(h_i | d)$$

Bayesian Optimal Classifier

No other classification method (using same hypothesis space and prior knowledge) can outperform a Bayesian Optimal Classifier in average.
Bayes Optimal Classifier provides the best results, but can be extremely expensive.

Use approximated methods $\Rightarrow$ the Gibbs Algorithm

1. Choose a hypothesis $h_G$ from $H$ at random, by drawing it from the current posterior probability distribution $P(h_i|d)$

2. Use $h_G$ to predict the classification of the next instance

Under certain conditions, the expected number of misclassifications is at most twice the expected value of a Bayesian Optimal classifier.
Naive Bayes Classifier

One of the simplest, yet popular, tools based on a strong probabilistic assumption

Consider the setting

- **Target classification function** $f : X \rightarrow C$
- Each instance $x \in X$ is described by a set of attributes
  \[ x = \langle a_1, \ldots, a_i, \ldots, a_L \rangle \]
- Seek the MAP classification
  \[ c_{NB} = \arg \max_{c_j \in C} P(c_j | a_1, \ldots, a_L) \]
Naive Bayes Assumption

The MAP classification rewrites

\[ c_{NB} = \arg \max_{c_j \in C} P(c_j | a_1, \ldots, a_L) \]

\[ = \arg \max_{c_j \in C} P(a_1, \ldots, a_L | c_j) P(c_j) \]

**Naive Bayes**: Assume conditional independence between the attributes \( a_l \) given classification \( c_j \)

\[ P(a_1, \ldots, a_L | c_j) = \prod_{l=1}^{L} P(a_l | c_j) \]

Naive Bayes Classification

\[ c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{l=1}^{L} P(a_l | c_j) \]
Use it when
- Dealing with **moderate or large** training sets
- Attributes that describe instances are **seemingly conditionally independent**

In reality, conditional independence is often violated...

...but Naive Bayes does not seem to know it. It often works surprisingly well!!
- Text classification
- Medical Diagnosis
- Recommendation systems

Next class we’ll see how to **train a Naive Bayes classifier**
Part II

Bayesian Networks
Bayesian Networks
Summary

Representing Conditional Independence

- **Bayes Optimal Classifier (BOC)**
  - No independence information between RV
  - Computationally expensive

- **Naive Bayes (NB)**
  - Full independence given the class
  - Extremely restrictive assumption

![Diagram of a Bayesian Network]

- **Bayesian Network (BNs)** describe conditional independence between subsets of RV by a graphical model

\[ I(X, Y|Z) \iff P(X, Y|Z) = P(X|Z)P(Y|Z) \]

- Combine a-priori information concerning RV dependencies with observed data
Bayesian Network - Directed Representation

- **Directed Acyclic Graph (DAG)**
- **Nodes represent random variables**
  - Shaded $\Rightarrow$ observed
  - Empty $\Rightarrow$ un-observed
- **Edges describe the conditional independence relationships**
- Every variable is conditionally independent w.r.t. its non-descendant, given its parents

**Conditional Probability Tables (CPT) local to each node** describe the probability distribution given its parents

\[
P(Y_1, \ldots, Y_N) = \prod_{i=1}^{N} P(Y_i | pa(Y_i))
\]
Naive Bayes classifier can be represented as a Bayesian Network

A more compact representation $\implies$ Plate Notation

Allows specifying more (Bayesian) details of the model
Plate Notation

- Boxes denote **replication** for a number of times denoted by the letter in the corner.
- Shaded nodes are **observed** variables.
- Empty nodes denote un-observed **latent** variables.
- Black seeds identify **constant terms** (e.g., the prior distribution \( \pi \) over the classes).
Inference - How can one determine the distribution of the values of one/several network variables, given the observed values of others?

- Bayesian Networks contain all the needed information (joint probability)
- NP-hard problem in general
- Solves easily for a single unobserved variable...
- ...otherwise
  - Exact inference on small DAGs
  - Approximated inference (Variational methods)
  - Simulate the network (Monte Carlo)
# Learning with Bayesian Networks

## Structure

<table>
<thead>
<tr>
<th>Data</th>
<th>Fixed Structure</th>
<th>Fixed Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete</td>
<td>$P(Y</td>
<td>X)$</td>
</tr>
<tr>
<td>Incomplete</td>
<td>$X$ $\rightarrow$ $Y$</td>
<td>$X$ $\rightarrow$ $Y$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>Naive Bayes</th>
<th>Discover dependencies from the data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>Calculate Frequencies (ML)</td>
<td>Structure Search</td>
</tr>
<tr>
<td></td>
<td>Latent variables</td>
<td>Independence tests</td>
</tr>
<tr>
<td></td>
<td>EM Algorithm (ML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MCMC, VBEM (Bayesian)</td>
<td>Difficult Problem</td>
</tr>
<tr>
<td></td>
<td>Parameter Learning</td>
<td>Structural EM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structure Learning</td>
</tr>
</tbody>
</table>
Bayesian techniques formulate learning as probabilistic inference.

ML learning selects the hypothesis that maximizes data likelihood.

MAP learning selects the most likely hypothesis given the data (ML Regularization).

Conditional independence relations can be expressed by a Bayesian network.

- Parameters Learning (Next Lecture)
- Structure Learning