Foundations of Bayesian Learning

Davide Bacciu

Computational Intelligence & Machine Learning Group
Dipartimento di Informatica
Università di Pisa
bacciu@di.unipi.it

Introduzione all'Intelligenza Artificiale - A.A. 2012/2013

IN SUPREMÆ DIGNITATIS
1343

Lecture Outline

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

$$\text{Random Process} \Longrightarrow \quad \text{Coin Toss}$$

$$\text{Discrete RV} \Longrightarrow \quad X = \begin{cases} 0 & \text{if heads} \\ 1 & \text{if tails} \end{cases}$$

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

Probability Functions

- 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

$$P(X_1 = x_1, \ldots, X_N = x_n) = P(x_1 \wedge \cdots \wedge x_n)$$

Definition (Sum Rule)

Probabilities of all the events must sum to 1

$$\sum_x P(X = x) = 1$$

Product Rule and Conditional Probabilities

Definition (Product Rule a.k.a. Chain Rule)

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

- $P(x|y)$ 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

$$\begin{aligned} P(X_1 = x_1) &= \sum_{x_2} P(X_1 = x_1, X_2 = x_2) \\ &= \sum_{x_2} P(X_1 = x_1 | X_2 = x_2) P(X_2 = x_2) \end{aligned}$$

Bayes Rule

Given hypothesis $h_i \in H$ and observations $\mathbf{d}$

$$P(h_i|\mathbf{d}) = \frac{P(\mathbf{d}|h_i)P(h_i)}{P(\mathbf{d})} = \frac{P(\mathbf{d}|h_i)P(h_i)}{\sum_j P(\mathbf{d}|h_j)P(h_j)}$$

- $P(h_i)$ is the prior probability of h_i
- $P(\mathbf{d}|h_i)$ is the conditional probability of observing $\mathbf{d}$ given that hypothesis h_i is true (likelihood).
- $P(\mathbf{d})$ is the marginal probability of $\mathbf{d}$
- $P(h_i|\mathbf{d})$ is the posterior probability that hypothesis is true given the data and the previous belief about the hypothesis.

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

$$
\begin{aligned}
I(X,Y) \Leftrightarrow P(X,Y) &= P(X|Y)P(Y) \\
&= P(Y|X)P(X) = P(X)P(Y)
\end{aligned}
$$

- 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

$$
\begin{aligned}
I(X,Y|Z) \Leftrightarrow P(X,Y|Z) &= P(X|Y,Z)P(Y|Z) \\
&= P(Y|X,Z)P(X|Z) = P(X|Z)P(Y|Z)
\end{aligned}
$$

Representing Probabilities with Discrete Data

Joint Probability Distribution Table

X_1	$\dots$	X_i	$\dots$	X_n	$P(X_1, \dots, X_n)$
$x_1^{'}$	$\dots$	$x_i^{'}$	$\dots$	$x_n^{'}$	$P(x_1^{'}, \dots, x_n^{'})$
x_1^{l}	$\dots$	x_i^{l}	$\dots$	x_n^{l}	$P(x_1^{l}, \dots, x_n^{l})$

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

In general, any probability of interest can be obtained starting from the Joint Probability Distribution $P(X_1, \dots, X_n)$

Wrapping Up....

- We know how to represent the world and the observations
  - Random Variables $\Longrightarrow X_1, \dots, X_N$
  - Joint Probability Distribution $\Longrightarrow P(X_1 = x_1, \dots, X_N = x_n)$
- We have rules for manipulating the probabilistic knowledge
  - Sum-Product
  - Marginalization
  - Bayes
  - Conditional Independence
- It is about time that we do some...
  - Inference - Reasoning and making predictions from a Bayesian perspective
  - Learning - Discover the values for $P(X_1 = x_1, \dots, X_N = x_n)$

Part II

Inference

Bayesian Learning and Inference

- 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

$$\boxed{P(X|\mathbf{D}=\mathbf{d})} = \sum_i P(X|\mathbf{d}, h_i)P(h_i|\mathbf{d})$$

$$= \sum_i \boxed{P(X|h_i)} \cdot \boxed{P(h_i|\mathbf{d})}$$

New prediction   Hypothesis prediction   Posterior weighting

Single Hypothesis Approximation

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

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

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

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

All Too Abstract?

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 judgement
  - 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

The Candy Box Problem

- 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 $\mathbf{d} = d_1, \ldots, d_N$ extracted and reinserted in a box (observations), what is the most likely flavor for the next candy (prediction)?

Candy Box Problem

Hypothesis Posterior

- First, we need to compute the posterior for each hypothesis (Bayes)

$$P(h_i|\mathbf{d}) = \alpha P(\mathbf{d}|h_i)P(h_i)$$

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

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

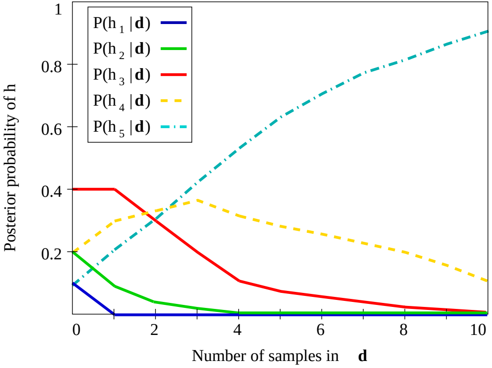
- Data likelihood can be computed under the assumption that observations are independently and identically distributed (i.i.d.)

$$P(\mathbf{d}|h_i) = \prod_{j=1}^{N} P(d_j|h_i)$$

Candy Box Problem

Hypothesis Posterior Computation

Suppose that the bag is a h_5 and consider a sequence of 10 observed lime candies



Hyp	d_0	d_1	d_2
h_1	0.1	0	0
h_2	0.2	0.1	0.03
h_3	0.4	0.4	0.30
h_4	0.2	0.3	0.35
h_5	0.1	0.2	0.31

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

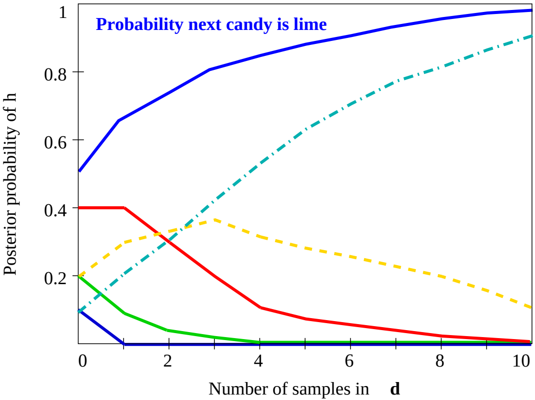
Most likely MAP hypothesis is re-evaluated as more data comes in

Candy Box Problem

Comparing Predictions

Bayesian learning seeks

$$P(d_{11} = l | d_1 = l, \ldots, d_{10} = l) = \sum_{i=1}^{5} P(d_{11} = l | h_i) P(h_i | \mathbf{d})$$



Observations

- Both ML and MAP are point estimates since they only make predictions based on the most likely hypothesis
- MAP predictions are approximately Bayesian if $P(X|\mathbf{d}) \sim P(X|h_{MAP})$
- 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

Part III

Learning

Parameter Learning in Bayesian Models

- Find numerical parameters for a probabilistic model
- Determine the best hypothesis h_θ regulated by a (set of) parameter θ

  h_θ : the expected proportion of coin tosses returning heads is θ
- Define the usual Bayesian probabilities

  Prior $P(h_\theta) = P(\theta)$

  Likelihood $P(\mathbf{d}|h_\theta) = P(\mathbf{d}|\theta)$

  Posterior $P(h_\theta|\mathbf{d}) = P(\theta|\mathbf{d})$
- If hypotheses are equiprobable it is reasonable to try Maximum Likelihood Estimation

Biased Coin

Estimate the probability of a coin toss returning head

Maximum Likelihood Find the unknown probability of heads θ

$$\theta_{ML} = \frac{nheads}{nheads + ntails}$$

Maximum a Posteriori Learn the distribution for the expected proportion of heads θ given the data

$$P(\theta|\mathbf{d}) = P(\mathbf{d}|\theta)P(\theta) \sim Beta(\alpha_H + nheads, \alpha_T + ntails)$$

where α_H and α_T can be thought of as imaginary counts of our prior experience

Naive Bayes Classifier

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

Consider the setting

- Target classification function $f : X \longrightarrow C$
- Each instance $x \in X$ is described by a set of attributes

$$x =< a_1, \ldots, a_l, \ldots, a_L >$$

- 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

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{l=1}^{L} P(a_l | c_j)$$

Learning Naive Bayes with Discrete Data (I)

- Given N observed training pairs $\mathbf{d} = \{(x_j, c_j)\}$ s.t. $x_j =< a_1, \ldots, a_L >$
- Find the maximum likelihood estimate of the model parameters θ

$$\max_{\theta} P(\mathbf{d}|\theta)$$

- The Naive Bayes parameters θ include
  - The attribute-class distribution $P(a_l = s|c = k)$ s.t. $1 \leq s \leq S$ and $1 \leq k \leq K$
  - The class prior $P(c = k)$ s.t. $1 \leq k \leq K$

Notice

Learning is performed by ML, while classification is performed by selecting the MAP hypothesis

Learning Naive Bayes with Discrete Data (II)

Class prior update

$$P(c=k) = \frac{\sum_{j=1}^{D} z_{jk}}{D} = \frac{N(k)}{D}$$

Attribute-class distribution update

$$P(a_l = s|c=k) = \frac{\sum_{j=1}^{D} z_{jk} t_j^{ls}}{\sum_{j=1}^{D} z_{jk} L} = \frac{N_{ls}(k)}{L \cdot S \cdot N(k)}$$

Maximum likelihood estimates are computed by counting the realizations of an event to obtain frequencies

Naive Bayes Classification

- Learning essentially amounts to counting frequencies
- Naive Bayes classification works surprisingly well when...
  - Attributes are close to be independent
  - Noisy data
  - High dimensional problems (scalability)
  - Large datasets (point estimates)
- However, what happens if a class c_k has no occurrences of an attribute $a_l = s$

$$P(a_l = s|c = k) = \frac{N_{ls}(k)}{L \cdot S \cdot N(k)} = 0 \Longrightarrow P(c = k|x) = 0 \; \forall x$$

Need to be careful when applying NB to sparse data

Smoothed Naive Bayes

- Smoothing $\Rightarrow$ dealing with the zero events problem
- Add a constant term α in both the numerator and the denominator to smooth the estimation

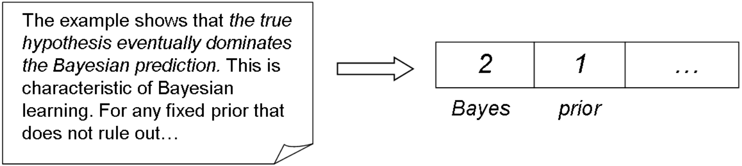
$$P(a_l = s | c = k) = \frac{N_{ls}(k) + \alpha}{L \cdot S \cdot N(k) + L \cdot S \cdot \alpha}$$

- $\alpha = 1 \Rightarrow$ Laplace smoothing
- Can improve NB performance up to 20%..
  - ..or it can cause interference in learning
  - Giving too much probability to unfrequent events
- α is a priori estimate of the attribute-class probability

Text Classification

- Loads of useful applications
  - Learn to classify web-pages by topic
  - Determine if an incoming email contains spam
  - ...
- Problem characterized by
  - Large sample size (i.e. large document collections)
  - High dimensional data (i.e. the vocabulary)
- Well-fit for Naive Bayes classifiers
  - One of the most effective models used in the field (e.g. DSPAM, SpamAssassin, SpamBayes, Bogofilter)
  - Need to count events (document representation)

Bag of Words Document Representation



- Count the occurrences of each dictionary word in your document
- Represent a document d as a vector x_d of word counts
- Easy to compute frequencies from word counts

Definition (Bag of Words Assumption)

Word order is not relevant for determining document semantics

Learning Naive Bayes for Text Classification

- Given a set of N training documents represented as vector of word counts $x_j = [w_1, \ldots, w_l, \ldots, w_L]$ (vocabulary size L)
- **for each** document classification $k = 1$ to K
  - $doc(k) \leftarrow$ set of training documents in class k
  - $P(c = k) \leftarrow \dfrac{|doc(k)|}{N}$
  - $text(k) \leftarrow$ concatenation of all docs in $doc(k)$
  - $N_j \leftarrow |text(k)|$ including duplicates
  - **for each** word w_l in the vocabulary
    - $n_l \leftarrow$ no occurrences of w_l in $text(k)$
    - $P(w_l|c = k) \leftarrow \dfrac{n_l + 1}{N_j + L}$

$$\text{Predict } c_{NB} = \arg\max_k P(c = k) \prod_{l=1}^{L} P(w_l|k)$$

20 Newsgroups Case Study

- Collection of approximately 20K newsgroup documents partitioned evenly across 20 different newsgroups

  Training set 10K documents
  Test set 7K documents
  Vocabulary 100 words

- Learning to classify an incoming newsgroup message into one of the top 4 high level newsgroup classes

comp.	**rec.**	**sci.**	**talk.**
comp.graphics	rec.motorcycles	sci.crypt	talk.politics.misc
comp.os.ms-windows.misc	rec.sport.baseball	sci.electronics	talk.politics.guns
comp.sys.ibm.pc.hardware	rec.sport.hockey	sci.med	talk.politics.mideast
comp.sys.mac.hardware	rec.autos	sci.space	talk.religion.misc
comp.windows.x			

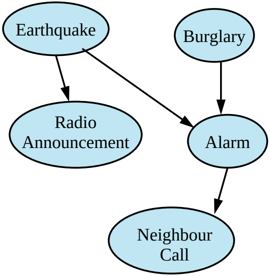
Credit goes to Mark Girolami @ Glasgow University

Part IV

Bayesian Networks

Representing Conditional Independence

- Naive Bayes (NB)
  - Full independence given the class
  - Extremely restrictive assumption
- Bayes Optimal Classifier (BOC)
  - No independence information between RV
  - Computationally expensive

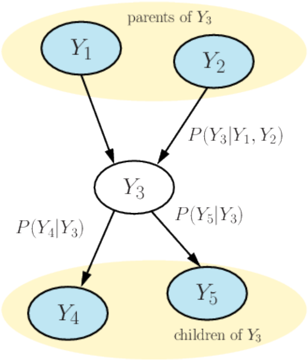


- Bayesian Network (BNs) describe conditional independence between subsets of RV by a graphical model

$$I(X, Y|Z) \Leftrightarrow P(X, Y|Z) = P(X|Z)P(Y|Z)$$

- Combine a-priori information concerning RV dependencies with observed data

Bayesian Network - Directed Representation

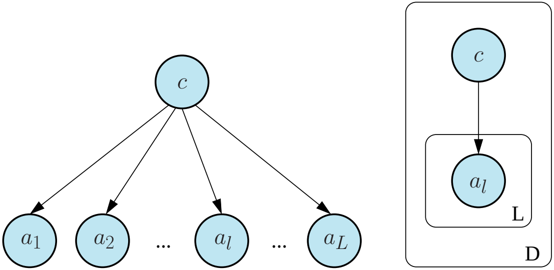


- Directed Acyclic Graph (DAG)
- Nodes represent random variables
  - Shaded $\Rightarrow$ observed
  - Empty $\Rightarrow$ un-observed
- Edges describe the conditional independence relationships
- Every variable is conditionally independent w.r.t. its non-descendant, given its parents

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

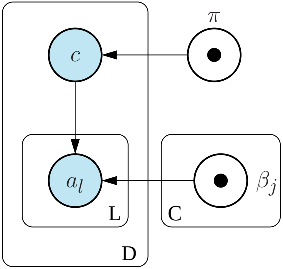
$$P(Y_1, \ldots, Y_N) = \prod_{i=1}^{N} P(Y_i|pa(Y_i))$$

Naive Bayes as a Bayesian Network



- Naive Bayes classifier can be represented as a Bayesian Network
- A more compact representation $\Longrightarrow$ Plate Notation
- Allows specifying more (Bayesian) details of the model

Plate Notation



Bayesian Naive Bayes

- 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 π over the classes)

Learning with Bayesian Networks

		Structure	
		Fixed Structure $X \xrightarrow{P(Y \mid X)} Y$	Fixed Variables $X \quad P(X, Y) \quad Y$
Data	Complete	Naive Bayes Calculate Frequencies (ML)	Discover dependencies from the data Structure Search Independence tests
	Incomplete	Latent variables EM Algorithm (ML) MCMC, VBEM (Bayesian)	Difficult Problem Structural EM
		Parameter Learning	Structure Learning

Take Home Messages

- Bayesian learning is a powerful all-in-one model for
  - Modeling your knowledge about the world (Bayesian Networks)
  - Inference - probabilistic approach to reasoning and prediction
  - Learning - discovering the parameters of given probability distributions
- Bayesian representation of the world
  - Random variables as building blocks
  - Conditional independence relations among RV expressed graphically by a Bayesian network
- ML learning selects the hypothesis that maximizes data likelihood
- MAP learning selects the most likely hypothesis given the data