6 Bayesian Learning

- Bayes Theorem
- MAP, ML hypotheses
- MAP learners
- Bayes optimal classifier
- Naive Bayes learner
- Example: Learning over text data
- Bayesian belief networks
- Expectation Maximization algorithm
Two Roles for Bayesian Methods

▶ Provides practical learning algorithms:
  ▶ Naive Bayes learning
  ▶ Bayesian belief network learning
  ▶ Combine prior knowledge (prior probabilities) with observed data
  ▶ Requires prior probabilities
  ▶ Often significant computation cost

▶ Provides useful conceptual framework;
  ▶ Provides “gold standard” for evaluating other learning algorithms
  ▶ Additional insight into Occam’s razor
Bayes Theorem

Let \( H \) be a hypothesis space and \( D \) a set of observed training data.

What is the best hypothesis from \( H \)?

Bayes Theorem

\[
P(h|D) = \frac{P(D|h)P(h)}{P(D)}
\]

where

\( P(h) \) = prior probability of hypothesis \( h \)
\( P(D) \) = prior probability of training data \( D \)
\( P(h|D) \) = probability of \( h \) given \( D \)
\( P(D|h) \) = probability of \( D \) given \( h \)
Choosing Hypotheses

▶ What is the most probable hypothesis $h \in H$ given the training data $D$?

▶ Maximum a posteriori hypothesis $h_{MAP}$:

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

▶ If we assume $P(h_i) = P(h_j)$ then

$$h_{ML} = \arg\max_{h_i \in H} P(D|h_i)$$

is called the Maximum likelihood ($h_{ML}$) hypothesis.

▶ Here:

▷ $D$ is a set of training examples of some target function.
▷ $H$ is the space of candidate target functions.
An Example

- Does patient have cancer or not?

- A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

- $P(cancer) = .008 \quad P(\neg cancer) = .992$
  $P(\oplus|cancer) = .98 \quad P(\ominus|cancer) = .02$
  $P(\oplus|\neg cancer) = .03 \quad P(\ominus|\neg cancer) = .97$

- $P(\oplus|cancer)P(cancer) = .98 \times .008 = .0078$
  $P(\oplus|\neg cancer)P(\neg cancer) = .03 \times .992 = .0298$

- $h_{MAP} = \neg cancer$. 
Basic Formulas for Probabilities

▶ Product Rule: probability $P(A \land B)$ of a conjunction of two events $A$ and $B$:

$$P(A \land B) = P(A | B)P(B) = P(B | A)P(A).$$

▶ Sum Rule: probability of a disjunction of two events $A$ and $B$:

$$P(A \lor B) = P(A) + P(B) - P(A \land B).$$

▶ Bayes Theorem:

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}.$$

▶ Theorem of total probability:

if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

$$P(B) = \sum_{i=1}^{n} P(B | A_i) P(A_i).$$
Brute Force MAP Hypothesis Learner

▶ For each hypothesis $h$ in $H$, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}.$$ 

▶ Output the hypothesis $h_{MAP}$ with the highest posterior probability

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

▷ Significant computing costs.

▷ Provides a standard.
Consider our usual concept learning task:

- Instance space $X$, hypothesis space $H$, training examples $D$.
- Consider the FINDS learning algorithm (outputs most specific hypothesis from the version space $V_{S_{H,D}}$)

What would Bayes rule produce as the MAP hypothesis?

Does FINDS output a MAP hypothesis?
Relation to Concept Learning

- Assume fixed set of instances $\langle x_1, \ldots, x_m \rangle$.
- Assume $D$ is the set of classifications $D = \langle c(x_1), \ldots, c(x_m) \rangle$.
- Assumptions:
  - $D$ is noise free, i.e., $d_i = c(x_i)$.
  - Target concept $c$ is in $H$.
  - No a priori reason to believe that any hypothesis is more probable than other.
- Choose
  - $P(D|h) = \begin{cases} 1 & \text{if } h \text{ consistent with } D \\ 0 & \text{otherwise} \end{cases}$
  - $P(h) = \frac{1}{|H|}$ for all $h \in H$
- Then,
  
  $$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$
Evolution of Posterior Probabilities

$P(h)$  $P(h|D_1)$  $P(h|D_1, D_2)$

hypotheses $(a)$  hypotheses $(b)$  hypotheses $(c)$
Characterizing Learning Algorithms by MAP Learners

Inductive system

Training examples D

Hypothesis space H

Candidate Elimination Algorithm

Output hypotheses

Equivalent Bayesian inference system

Training examples D

Hypothesis space H

P(h) uniform
P(D|h) = 0 if inconsistent,
= 1 if consistent

Brute force MAP learner

Output hypotheses

Prior assumptions made explicit
Consider any real-valued target function $f$.

Training examples $\langle x_i, d_i \rangle$, $1 \leq i \leq m$, where $d_i$ is noisy training value

- $d_i = f(x_i) + e_i$
- $e_i$ is random variable (noise) drawn independently for each $x_i$ according to some Gaussian distribution with mean 0.

Then the maximum likelihood hypothesis $h_{ML}$ is the one that minimizes the sum of squared errors:

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$
Most Probable Classification of New Instances

- So far we’ve sought the most probable hypothesis given the data $D$ (i.e., $h_{MAP}$).
- Given new instance $x$, what is its most probable classification?
- $h_{MAP}(x)$ is not the most probable classification!
- For example, consider:
  - Three possible hypotheses:
    $$P(h_1|D) = .4, \ P(h_2|D) = .3, \ P(h_3|D) = .3.$$
  - Given new instance $x$:
    $$h_1(x) = \oplus, \ h_2(x) = \ominus, \ h_3(x) = \ominus.$$
  - $h_{MAP} = h_1$, thus $x$ is classified $\oplus$ by $h_{MAP}$.
  - But most probable classification of $x$ is $\ominus$. 
Bayes Optimal Classifier

▶ Assume that $x$ can take any value $v_j \in V$.

▶ $P(v_j | D) = \sum_{h_i \in H} P(v_j | h_i) P(h_i | D)$.

▶ Bayes optimal classification:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j | h_i) P(h_i | D)$$

▶ Example:

$P(h_1 | D) = .4$  $P(\emptyset | h_1) = 0$  $P(\oplus | h_1) = 1$
$P(h_2 | D) = .3$  $P(\emptyset | h_2) = 1$  $P(\oplus | h_2) = 0$
$P(h_3 | D) = .3$  $P(\emptyset | h_3) = 1$  $P(\oplus | h_3) = 0$

▶ $\sum_{h_i \in H} P(\oplus | h_i) P(h_i | D) = .4$
$\sum_{h_i \in H} P(\emptyset | h_i) P(h_i | D) = .6$

▶ $\arg\max_{v_j \in \{\oplus, \emptyset\}} \sum_{h_i \in H} P(v_j | h_i) P(h_i | D) = \emptyset$. 

6 Bayesian Learning 14
Gibbs Classifier

- Bayes optimal classifier provides best result,
- but can be expensive if many hypotheses.

- Gibbs algorithm:
  - Choose one hypothesis at random, according to $P(h|D)$.
  - Use this to classify new instance.

- Surprising fact:
  $$E[error_{\text{Gibbs}}] \leq 2 \times E[error_{\text{BayesOptimal}}]$$

- Suppose correct, uniform prior distribution over $H$, then
  - Pick any hypothesis from VS, with uniform probability.
  - Its expected error is no worse than twice the expected error of the optimal Bayes classifier.
Naive Bayes Classifier – Introduction

- Along with decision trees, neural networks, nearest neighbour, one of the most practical learning methods.

- When to use:
  - Moderate or large training set available.
  - Attributes describing instances are conditionally independent given classification.

- Successful applications:
  - Diagnosis.
  - Classifying text documents.
Naive Bayes Classifier

▶ Assume target function $f : X \to V$, where each instance $x$ is described by attributes $\langle a_1, \ldots, a_n \rangle$.

▶ Most probable value of $f(x)$ is:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, \ldots, a_n)$$

$$= \arg\max_{v_j \in V} \frac{P(a_1, \ldots, a_n | v_j) P(v_j)}{P(a_1, \ldots, a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, \ldots, a_n | v_j) P(v_j).$$

▶ Naive Bayes assumption:

$$P(a_1, \ldots, a_n | v_j) = \prod_i P(a_i | v_j),$$

▶ which gives the Naive Bayes classifier ($v_{NB}$):

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j).$$
Naive Bayes Algorithm

Haye_Bayes_Learn(examples)

▷ For each target value \( v_j \)
  • \( \hat{P}(v_j) \leftarrow \) estimate \( P(v_j) \)
  • For each attribute value \( a_i \) of each attribute \( a \)
    ○ \( \hat{P}(a_i|v_j) \leftarrow \) estimate \( P(a_i|v_j) \)

Haye_New_Instance(\( x \))

\[
v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)
\]
Consider PlayTennis again, and new instance

\[ \langle \text{Outlk} = \text{sun}, \text{Temp} = \text{cool}, \text{Humid} = \text{high}, \text{Wind} = \text{strong} \rangle. \]

Want to compute:

\[ v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j). \]

\[ \hat{P}(y) = \frac{9}{14} = .64 \]
\[ \hat{P}(n) = \frac{5}{14} = .36 \]
\[ \hat{P}(\text{strong}|y) = \frac{3}{9} = .33 \]
\[ \hat{P}(\text{strong}|n) = \frac{3}{5} = .60 \]

etc.

\[ \hat{P}(y) \times \hat{P}(\text{sun}|y) \times \hat{P}(\text{cool}|y) \times \hat{P}(\text{high}|y) \times \hat{P}(\text{strong}|y) = .0053 \]
\[ \hat{P}(n) \times \hat{P}(\text{sun}|n) \times \hat{P}(\text{cool}|n) \times \hat{P}(\text{high}|n) \times \hat{P}(\text{strong}|n) = .0206 \]

\[ v_{NB} = n \text{ with conditional probability } \frac{.0206}{.0206 \times .0053} = .795. \]
Naive Bayes: Subtleties (1)

- Conditional independence assumption

\[
P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)
\]

is often violated.

- But it works surprisingly well anyway.
- Don’t need estimated posteriors \( \hat{P}(v_j | x) \) to be correct; need only that

\[
\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i | v_j) = \arg\max_{v_j \in V} P(v_j) P(a_1 \ldots, a_n | v_j)
\]

- see [Domingos & Pazzani, 1996] for analysis.
- Naive Bayes posteriors often unrealistically close to 1 or 0.
What if none of the training instances with target value $v_j$ have attribute value $a_i$?

\[ \hat{P}(a_i|v_j) = 0 \]
\[ \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0 \]

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

\[ \hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m} \]

where

- $n$ is number of training examples for which $v = v_j$,
- $n_c$ number of examples for which $v = v_j$ and $a = a_i$
- $p$ is prior estimate for $\hat{P}(a_i|v_j)$
- $m$ is weight given to prior (i.e. number of “virtual” examples)
Learning to Classify Text

- Learn which news articles are of interest.
- Learn to classify web pages by topic.
- Naive Bayes is among most effective algorithms.
- What attributes shall we use to represent text documents?
Learning to Classify Text

- Target concept *Interesting?* : Doc → {+, −}.
- Represent each document by vector of words.
- One attribute per word position in document; value is word at that position.
- Use training examples to estimate:
  \[ P(+), P(−), P(Doc|+), P(Doc|−). \]
- Naive Bayes conditional independence assumption
  \[
  P(Doc|v_j) = \prod_{i=1}^{\text{length}(Doc)} P(a_i = w_k|v_j)
  \]
  where \( P(a_i = w_k|v_j) \) is the probability that word in position \( i \) is \( w_k \), given \( v_j \).
- Additional assumption: \( (\forall i, m) P(a_i = w_k|v_j) = P(a_m = w_k|v_j) \).
**Learn\_naive\_Bayes\_text**\((Examples, V)\)

- Collect all words and other tokens that occur in *Examples*:
  - \( Vocabulary \leftarrow \) all distinct words and other tokens in *Examples*.

- Calculate the required \( P(v_j) \) and \( P(w_k|v_j) \) probability terms:
  - For each target value \( v_j \) in \( V \) do:
    - \( docs_j \leftarrow \) subset of *Examples* for which the target value is \( v_j \).
    - \( P(v_j) \leftarrow \frac{|docs_j|}{|Examples|} \).
    - \( Text_j \leftarrow \) document created by concatenating all members of \( docs_j \).
    - \( n \leftarrow \) total number of word occurrences in \( Text_j \).
    - For each word \( w_k \) in \( Vocabulary \) do:
      - \( n_k \leftarrow \) number of times word \( w_k \) occurs in \( Text_j \).
      - \( P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|} \).
Classify_naive_Bayes_text(Doc)

Return the estimated target value for document Doc:

- \( \textit{positions} \leftarrow \text{all word positions in } Doc \text{ that contain tokens found in Vocabulary.} \)
- Return \( v_{NB} \), where

\[
v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in \text{positions}} P(a_i | v_j)
\]

and \( a_i \) denotes the word found in the ith position within \( Doc \).
Twenty NewsGroups

- Given 1000 training documents from each group.
- Learn to classify new documents according to which newsgroup it came from:
  - comp.graphics    misc.forsale
  - comp.os.ms-windows.misc  rec.autos
  - comp.sys.ibm.pc.hardware  rec.motorcycles
  - comp.sys.mac.hardware  rec.sport.baseball
  - comp.windows.x  rec.sport.hockey
  - alt.atheism    sci.space
  - soc.religion.christian  sci.crypt
  - talk.religion.misc  sci.electronics
  - talk.politics.mideast  sci.med
  - talk.politics.misc
  - talk.politics.guns
- Naive Bayes: 89% classification accuracy.
I can only comment on the Kings, but the most obvious candidate for pleasant surprise is Alex Zhitnik. He came highly touted as a defensive defenseman, but he’s clearly much more than that. Great skater and hard shot (though wish he were more accurate). In fact, he pretty much allowed the Kings to trade away that huge defensive liability Paul Coffey. Kelly Hrudey is only the biggest disappointment if you thought he was any good to begin with. But, at best, he’s only a mediocre goaltender. A better choice would be Tomas Sandstrom, though not through any fault of his own, but because some thugs in Toronto decided
Learning Curve for 20 Newsgroups

Accuracy vs. Training set size (1/3 withheld for test)
Bayesian Belief Networks

- Naive Bayes assumption of conditional independence too restrictive.
- But it’s intractable without some such assumptions.
- Bayesian Belief networks describe conditional independence among *subsets* of variables.
- Allows combining prior knowledge about (in)dependencies among variables with observed training data.
Conditional Independence

> **Definition:** $X$ is conditionally independent of $Y$ given $Z$ if

$$(\forall x_i, y_j, z_k) \ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k).$$

More compactly, we write

$$P(X|Y, Z) = P(X|Z).$$

> **Example:** *Thunder* is conditionally independent of *Rain*, given *Lightning*:

$$P(\text{Thunder}|\text{Rain, Lightning}) = P(\text{Thunder}|\text{Lightning}).$$

> Naive Bayes uses conditional independence to justify

$$P(X, Y|Z) = P(X|Y, Z)P(Y|Z)$$

$$= P(X|Z)P(Y|Z)$$
Network represents a set of conditional independence assertions:

- Each node is asserted to be conditionally independent of its nondescendants, given its immediate predecessors.
- Directed acyclic graph.
Example Bayesian Belief Network (2)

Represents joint probability distribution over all variables.

- e.g., \( P(\text{Storm}, \text{BusTourGroup}, \ldots, \text{ForestFire}) \)
- in general,

\[
P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | \text{Parents}(Y_i))
\]

where \( \text{Parents}(Y_i) \) denotes immediate predecessors of \( Y_i \) in graph.
- Hence, joint distribution is fully defined by graph, plus the \( P(y_i | \text{Parents}(Y_i)) \).
Inference in Bayesian Networks

- How can one infer the (probabilities of) values of one or more network variables, given observed values of others?
- Bayesian network contains all information needed for this inference.
- If only one variable with unknown value, easy to infer it.
- In general case, problem is NP-hard.
- In practice, can succeed in many cases.
- Exact inference methods work well for some network structures.
- Monte Carlo methods “simulate” the network randomly to calculate approximate solutions.
Learning of Bayesian Networks (1)

- Several variants of this learning task:
  - Network structure might be *known* or *unknown*.
  - Training examples might provide values of *all* network variables, or just *some*.

- If structure known and all variables observable,
  - then it’s easy as training a Naive Bayes classifier.
Learning of Bayesian Networks (2)

- Suppose structure known, variables partially observable.
- E.g., observe *ForestFire, Storm, BusTourGroup, Thunder*, but not *Lightning, Campfire*.
- Similar to training neural network with hidden units.
- We can learn network conditional probability tables using gradient ascent.
- Converge to network $h$ that (locally) maximizes $P(D|h)$.
Let $w_{ijk}$ denote one entry in the conditional probability table for variable $Y_i$ in the network:

$$w_{ijk} = P(Y_i = y_{ij} | Parents(Y_i) = \text{the list } u_{ik} \text{ of values}).$$

E.g., if $Y_i = \text{Campfire}$, then $u_{ik}$ might be $\langle \text{Storm} = T, \text{BusTourGroup} = F \rangle$

Perform gradient ascent by repeatedly:

- update all $w_{ijk}$ using training data $D$:

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P(y_{ij}, u_{ik} | d)}{w_{ijk}}$$

- then, renormalize the $w_{ijk}$ to assure:

  - $\sum_j w_{ijk} = 1$,
  - $0 \leq w_{ijk} \leq 1$. 

6 Bayesian Learning
EM algorithm can also be used. Repeatedly:

- Calculate probabilities of unobserved variables, assuming $h$.
- Calculate new $w_{ijk}$ to maximize $E[\ln P(D|h)]$,
  - where $D$ includes both observed and (calculated probabilities of) unobserved variables.

When structure unknown:

- Algorithms use greedy search to add/subtract edges and nodes.
- Active research topic.
Summary: Bayesian Belief Networks

- Combine prior knowledge with observed data.
- Impact of prior knowledge (when correct!) is to lower the sample complexity.
- Active research area:
  - Extend from boolean to real-valued variables.
  - Parameterized distributions instead of tables.
  - Extend to first-order instead of propositional systems.
  - More effective inference methods.
  - etc.