



MACHINE LEARNING

Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Bioinformatics and Computational Biology Program
Center for Computational Intelligence, Learning, & Discovery
Iowa State University
honavar@cs.iastate.edu
www.cs.iastate.edu/~honavar/
www.cild.iastate.edu/

Learning as Bayesian Inference

Probability is the logic of Science (Jaynes)

- Bayesian (subjective) probability provides a basis for updating beliefs based on evidence
- By updating beliefs about hypotheses based on data, we can learn about the world.
- Bayesian framework provides a sound probabilistic basis for understanding many learning algorithms and designing new algorithms
- Bayesian framework provides several practical reasoning and learning algorithms

Classification using Bayesian Decision Theory

Consider the problem of classifying an instance X into one of two mutually exclusive classes ω_1 or ω_2

$P(\omega_1|X)$ = probability of class ω_1 given the evidence X

$P(\omega_2|X)$ = probability of class ω_2 given the evidence X

What is the probability of error?

$$\begin{aligned} P(\text{error} | X) &= P(\omega_1|X) \text{ if we choose } \omega_2 \\ &= P(\omega_2|X) \text{ if we choose } \omega_1 \end{aligned}$$

Minimum Error Classification

To minimize classification error

Choose ω_1 if $P(\omega_1|X) > P(\omega_2|X)$

Choose ω_2 if $P(\omega_2|X) > P(\omega_1|X)$

which yields

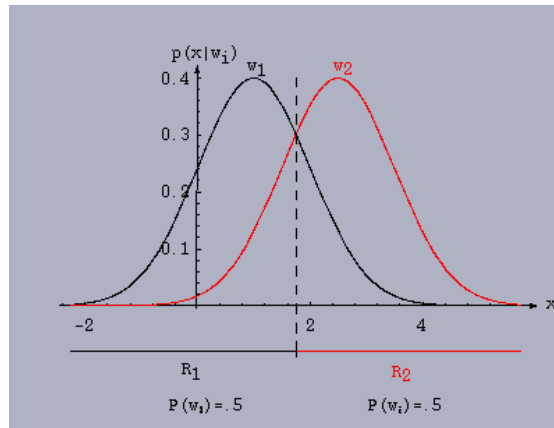
$$P(\text{error} | X) = \min[P(\omega_1|X), P(\omega_2|X)]$$

We have:

$$P(\omega_1|X) = P(X | \omega_1)P(\omega_1);$$

$$P(\omega_2|X) = P(X | \omega_2)P(\omega_2)$$

Classification using Bayesian decision theory



Choose ω_1 if $P(\omega_1|X) > P(\omega_2|X)$ i.e. $X \in R_1$
Choose ω_2 if $P(\omega_2|X) > P(\omega_1|X)$ i.e. $X \in R_2$

Copyright Vasant Honavar, 2006.

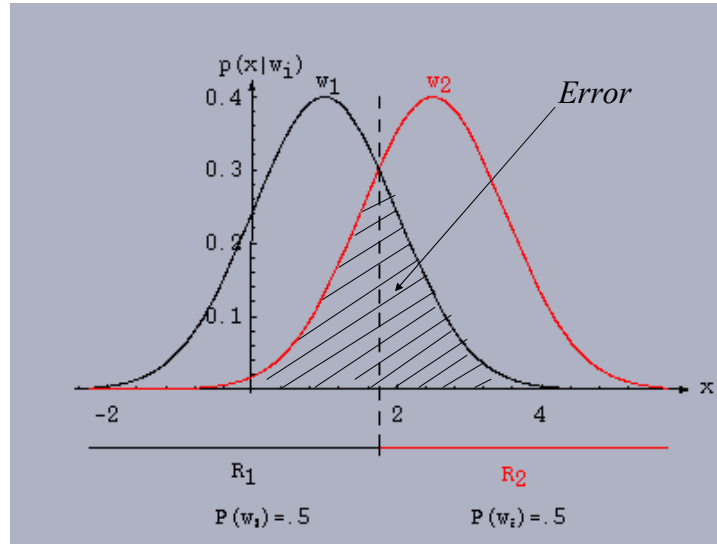
Optimality of Bayesian Decision Rule

We can show that the Bayesian classifier is optimal in that it is guaranteed to minimize the probability of misclassification

- (Proof given in class)

Copyright Vasant Honavar, 2006.

Optimality of Bayes Decision Rule



Copyright Vasant Honavar, 2006.

Optimality of the Bayes Decision Rule

$$\begin{aligned}
 P_e &= P(x \in R_2, x \in \omega_1) + P(x \in R_1, x \in \omega_2) \\
 &= P(x \in R_2 | \omega_1)P(\omega_1) + P(x \in R_1 | \omega_2)P(\omega_2) \\
 &= P(\omega_1) \int_{R_2} p(x | \omega_1) dx + P(\omega_2) \int_{R_1} p(x | \omega_2) dx
 \end{aligned}$$

Applying Bayes Rule :

$$p(x | \omega_i)P(\omega_i) = P(\omega_i | x)p(x) = p(x, \omega_i)$$

$$P_e = \int_{R_2} P(\omega_1 | x)p(x) dx + \int_{R_1} P(\omega_2 | x)p(x) dx$$

Copyright Vasant Honavar, 2006.

Optimality of the Bayes Decision Rule

$$P_e = \int_{R_2} p(\omega_1 | x)p(x)dx + \int_{R_1} p(\omega_2 | x)p(x)dx$$

Because $R_1 \cup R_2$ covers the entire input space,

$$\int_{R_1} P(\omega_1 | x)p(x)dx + \int_{R_2} P(\omega_1 | x)p(x)dx = P(\omega_1)$$

$$P_e = P(\omega_1) - \int_{R_1} (P(\omega_1 | x) - P(\omega_2 | x))p(x)dx$$

P_e is minimized by choosing

R_1 such that $P(\omega_1 | x) > P(\omega_2 | x)$

and

R_2 such that $P(\omega_2 | x) > P(\omega_1 | x)$

Optimality of Bayes Decision Rule

- The proof generalizes to multivariate input spaces
- Similar result can be proved in the case of discrete (as opposed to continuous) input spaces – replace integral over the input space by sum

Bayes Decision Rule yields Minimum Error Classification

To minimize classification error

Choose ω_1 if $P(\omega_1|X) > P(\omega_2|X)$

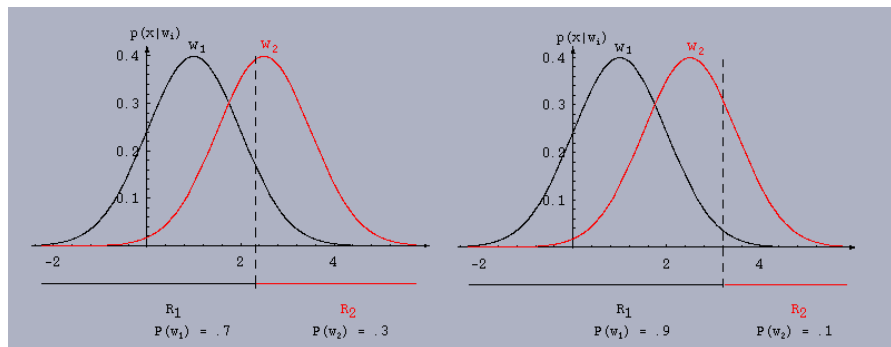
Choose ω_2 if $P(\omega_2|X) > P(\omega_1|X)$

which yields

$$P(\text{error} | X) = \min[P(\omega_1|X), P(\omega_2|X)]$$

Bayes Decision Rule

Behavior of Bayes decision rule as a function of prior probability of classes



Bayes Optimal Classifier

Classification rule that guarantees minimum error :

Choose ω_1 if $P(X | \omega_1)P(\omega_1) > P(X | \omega_2)P(\omega_2)$

Choose ω_2 if $P(X | \omega_2)P(\omega_2) > P(X | \omega_1)P(\omega_1)$

If $P(X | \omega_1) = P(X | \omega_2)$

classification depends entirely on $P(\omega_1)$ and $P(\omega_2)$

If $P(\omega_1) = P(\omega_2)$,

classification depends entirely on $P(X | \omega_1)$ and $P(X | \omega_2)$

Bayes classification rule combines the effect of the two terms optimally - so as to yield minimum error classification.

Generalization to multiple classes $c(X) = \arg \max_{\omega_j} P(\omega_j | X)$

Minimum Risk Classification

Let λ_{ij} = risk or cost associated with assigning an instance to class ω_j when the correct classification is ω_i

$R(\omega_i | X)$ = expected loss incurred in assigning X to class ω_i

$R(\omega_1 | X) = \lambda_{11}P(\omega_1 | X) + \lambda_{21}P(\omega_2 | X)$

$R(\omega_2 | X) = \lambda_{12}P(\omega_1 | X) + \lambda_{22}P(\omega_2 | X)$

Classification rule that guarantees minimum risk :

Choose ω_1 if $R(\omega_1 | X) < R(\omega_2 | X)$

Choose ω_2 if $R(\omega_2 | X) < R(\omega_1 | X)$

Flip a coin otherwise

Minimum Risk Classification

λ_{ij} = risk or cost associated with assigning an instance
to class ω_j when the correct classification is ω_i

Ordinarily $(\lambda_{21} - \lambda_{22})$ and $(\lambda_{12} - \lambda_{11})$ are positive
(cost of being correct is less than the cost of error)

So we choose ω_1 if $\frac{P(X|\omega_1)}{P(X|\omega_2)} > \frac{(\lambda_{21} - \lambda_{22}) P(\omega_2)}{(\lambda_{12} - \lambda_{11}) P(\omega_1)}$

Otherwise choose ω_2

Minimum error classification rule is a special case :

$$\lambda_{ij} = 0 \text{ if } i = j \text{ and } \lambda_{ij} = 1 \text{ if } i \neq j$$

This classification rule can be shown to be optimal in that it is guaranteed to minimize the risk of misclassification

Summary of Bayesian recipe for classification

λ_{ij} = risk or cost associated with assigning an instance
to class ω_j when the correct classification is ω_i

Choose ω_1 if $\frac{P(X|\omega_1)}{P(X|\omega_2)} > \frac{(\lambda_{21} - \lambda_{22}) P(\omega_2)}{(\lambda_{12} - \lambda_{11}) P(\omega_1)}$

Choose ω_2 if $\frac{P(X|\omega_1)}{P(X|\omega_2)} < \frac{(\lambda_{21} - \lambda_{22}) P(\omega_2)}{(\lambda_{12} - \lambda_{11}) P(\omega_1)}$

Minimum error classification rule is a special case :

Choose ω_1 if $\frac{P(X|\omega_1)}{P(X|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)}$ Otherwise choose ω_2

Bayesian recipe for classification

- Chef 1 - Generative (Informative) model

$$\text{Note that } P(\omega_i | \mathbf{x}) = \frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{P(\mathbf{x})}$$

Model $P(\mathbf{x} | \omega_1)$, $P(\mathbf{x} | \omega_2)$, $P(\omega_1)$, and $P(\omega_2)$

Using Bayes rule, choose ω_1 if $P(\mathbf{x} | \omega_1)P(\omega_1) > P(\mathbf{x} | \omega_2)P(\omega_2)$

Otherwise choose ω_2

- Chef 2: Discriminative Model (will return to this later)

Model $P(\omega_1 | \mathbf{x})$, $P(\omega_2 | \mathbf{x})$, or the ratio $\frac{P(\omega_1 | \mathbf{x})}{P(\omega_2 | \mathbf{x})}$ directly

Choose ω_1 if $\frac{P(\omega_1 | \mathbf{x})}{P(\omega_2 | \mathbf{x})} > 1$

Otherwise choose ω_2

Summary of Bayesian recipe for classification

- The Bayesian recipe is simple, optimal, and in principle, straightforward to apply
- To use this recipe in practice, we need to know $P(X|\omega_i)$ – the **generative model for data** for each class and $P(\omega_i)$ – the **prior probabilities of classes**
- **Because these probabilities are unknown, we need to estimate them from data – or learn them!**
- X is typically high-dimensional
- Need to estimate $P(X|\omega_i)$ from *limited* data

Naïve Bayes Classifier

- We can classify X if we know $P(X|\omega_i)$
- How to learn $P(X|\omega_i)$?

One solution: Assume that the random variables in X are conditionally independent given the class.

- **Result: Naïve Bayes classifier which performs optimally under certain assumptions**
- A simple, practical learning algorithm grounded in Probability Theory

When to use

- Attributes that describe instances are likely to be conditionally independent given classification
- The data is insufficient to estimate all the probabilities reliably if we do not assume independence

Naïve Bayes Classifier

Successful applications

- Diagnosis
- Document Classification
- Protein Function Classification
- Prediction of protein-protein interfaces
and many others.....

Conditional Independence

Let Z_1, \dots, Z_n and W be random variables on a given event space.

Z_1, \dots, Z_n are mutually independent given W if

$$P(Z_1, Z_2, \dots, Z_n | W) = \prod_{i=1}^n P(Z_i | W)$$

Note that these represent sets of equations, for all possible value assignments to random variables

Implications of Independence

- Suppose we have 5 Binary attributes and a binary class label
- Without independence, in order to specify the joint distribution, we need to specify a probability for each possible assignment of values to each variable resulting in a table of size $2^6=64$
- Suppose the features are independent given the class label – we only need $5(2 \times 2)=20$ entries
- The reduction in the number of probabilities to be estimated is even more striking when N , the number of attributes is large – from $O(2^N)$ to $O(N)$

Naive Bayes Classifier

Consider a discrete valued target function $f : \mathcal{X} \rightarrow \Omega$
 where an instance $X = (X_1, X_2, \dots, X_n) \in \mathcal{X}$ is described
 in terms of attribute values $X_1 = x_1, X_2 = x_2, \dots, X_n = x_n$
 where $x_i \in \text{Domain}(X_i)$

$$\begin{aligned} \omega_{MAP} &= \arg \max_{\omega_j \in \Omega} P(\omega_j | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \\ &= \arg \max_{\omega_j \in \Omega} \frac{P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n | \omega_j) P(\omega_j)}{P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)} \\ &= \arg \max_{\omega_j \in \Omega} P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n | \omega_j) P(\omega_j) \end{aligned}$$

ω_{MAP} is called the *maximum a posteriori* classification

Naive Bayes Classifier

$$\begin{aligned} \omega_{MAP} &= \arg \max_{\omega_j \in \Omega} P(\omega_j | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \\ &= \arg \max_{\omega_j \in \Omega} P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n | \omega_j) P(\omega_j) \end{aligned}$$

If the attributes are *independent* given the class, we have

$$\begin{aligned} \omega_{MAP} &= \arg \max_{\omega_j \in \Omega} \prod_{i=1}^n P(X_i = x_i | \omega_j) P(\omega_j) \\ &= \omega_{NB} \\ &= \arg \max_{\omega_j \in \Omega} P(\omega_j) \prod_{i=1}^n P(X_i = x_i | \omega_j) \end{aligned}$$

Naive Bayes Learner

For each possible value ω_j of Ω ,

$$\hat{P}(\Omega = \omega_j) \leftarrow \text{Estimate}(P(\Omega = \omega_j), D)$$

For each possible value a_{i_k} of X_i

$$\hat{P}(X_i = a_{i_k} | \omega_j) \leftarrow \text{Estimate}(P(X_i = a_{i_k} | \Omega = \omega_j), D)$$

Classify a new instance $X = (x_1, x_2, \dots, x_N)$

$$c(X) = \underset{\omega_j \in \Omega}{\operatorname{argmax}} P(\omega_j) \prod_{i=1}^n P(X_i = x_i | \omega_j)$$

Estimate is a procedure for estimating the relevant probabilities from set of *training examples*

Estimation of Probabilities from Small Samples

$$\hat{P}(X_i = a_{i_k} | \omega_j) \leftarrow \frac{n_{j i_k} + mp}{n_j + m}$$

n_j is the number of training examples of class ω_j

$n_{j i_k}$ = number of training examples of class ω_j

which have attribute value a_{i_k} for attribute X_i

p is the prior estimate for $\hat{P}(X_i = a_{i_k} | \omega_j)$

m is the weight given to the prior

$$\text{As } n \rightarrow \infty, \hat{P}(X_i = a_{i_k} | \omega_j) \rightarrow \frac{n_{j i_k}}{n_j}$$

This is effectively the same as using Dirichlet priors as we shall see later

Sample Applications of Naïve Bayes Classifier

- Learning dating preferences
- Learn which news articles are of interest.
- Learn to classify web pages by topic.
- Learn to classify SPAM
- Learn to assign proteins to functional families based on amino acid composition

Naive Bayes is among the most useful algorithms

What attributes shall we use to represent text?

Learning Dating Preferences

Instances –

ordered 3-tuples of attribute values corresponding to

Height (tall, short)

Hair (dark, blonde, red)

Eye (blue, brown)

Classes –

+, –

Training Data

Instance	Class label
I_1 (t, d, l)	+
I_2 (s, d, l)	+
I_3 (t, b, l)	–
I_4 (t, r, l)	–
I_5 (s, b, l)	–
I_6 (t, b, w)	+
I_7 (t, d, w)	+
I_8 (s, b, w)	+

Probabilities to estimate

$P(+)=5/8$

$P(-)=3/8$

$P(\text{Height} c)$	t	s
+	3/5	2/5
-	2/3	1/3

$P(\text{Hair} c)$	d	b	r
+	3/5	2/5	0
-	0	2/3	1/3

$P(\text{Eye} c)$	l	w
+	2/5	3/5
-	1	0

Classify ($\text{Height}=t, \text{Hair}=b, \text{eye}=l$)

$$P(X | +) = (3/5)(2/5)(2/5) = (12/125)$$

$$P(X | -) = (2/3)(2/3)(1) = (4/9)$$

Classification = ?

Classify ($\text{Height}=t, \text{Hair}=r, \text{eye}=w$)

Note the problem with zero probabilities

Solution – Use Laplacian estimates

Learning to Classify Text

- Target concept *Interesting?* : Documents $\rightarrow \{+, -\}$
- Learning: Use training examples to estimate $P(+), P(-), P(d|+), P(d|-)$

Alternative generative models for documents:

- Represent each document by sequence of words
 - In the most general case, we need a probability for each word occurrence in each position in the document, for each possible document length
 - Too many probabilities to estimate!
- Represent each document by tuples of word counts

$$P(d | \omega_i) = P(\text{length}(d)) \prod_{i=1}^{\text{length}(d)} P(X_i | \omega_i, \text{length}(d))$$

This would require estimating for each document,

$$|\text{Vocabulary}|^{\text{length}(d)} \times |\Omega|$$

probabilities for each possible document length!

To simplify matters, assume that probability of encountering a specific word in a particular position is independent of the position, and of document length

Treat each document as a bag of words!

Bag of Words Representation

So we estimate one position - independent class - conditional probability $P(w_k | \omega_j)$ for each word instead of the set of probabilities $P(X_1 = w_k | \omega_j) \dots P(X_{\text{length}(d)} = w_k | \omega_j)$

The number of probabilities to be estimated drops to

$$|\text{Vocabulary}| \times |\Omega|$$

The result is a generative model for documents that treats each document as an ordered tuple of word frequencies

More sophisticated models can consider dependencies between adjacent word positions (Markov models – we will come back to these later)

Learning to Classify Text

With the bag of words representation, we have

$$P(d | \omega_j) \text{ is proportional to } \left\{ \frac{\left(\sum_k n_{kd} \right)!}{\prod_k n_{kd}!} \right\} \prod_k \left(P(w_k | \omega_j) \right)^{n_{kd}}$$

where n_{kd} is the number of occurrences of w_k in document d
(ignoring dependence on length of the document)

We can estimate $P(w_k | \omega_j)$ from the labeled bags of words we have.

Naïve Bayes Text Classifier

- Given 1000 training documents from each group, learn to classify new documents according to the newsgroup where it belongs
- Naive Bayes achieves 89% classification accuracy

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	

Naïve Bayes Text Classifier

Representative article from rec.sport.hockey

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uwm.edu
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinion)...
Date: 5 Apr 93 09:53:39 GMT

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 Hrudehy 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 ...

Sequence Classification

Need a generative model for sequences

Simplest alternative – sequence-length independent multinomial (bag of letters) model!

More sophisticated alternatives possible – for example, Markov models that capture dependencies among small windows of neighboring letters

Naïve Bayes Learner – Summary

- Produces minimum error classifier if attributes are conditionally independent given the class

When to use

- Attributes that describe instances are likely to be conditionally independent given classification
- There is not enough data to estimate all the probabilities reliably if we do not assume independence
- Often works well even if when independence assumption is violated (Domigos and Pazzani, 1996)
- Can be used iteratively – Kang et al., 2006