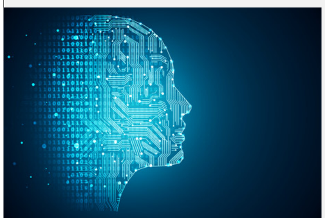
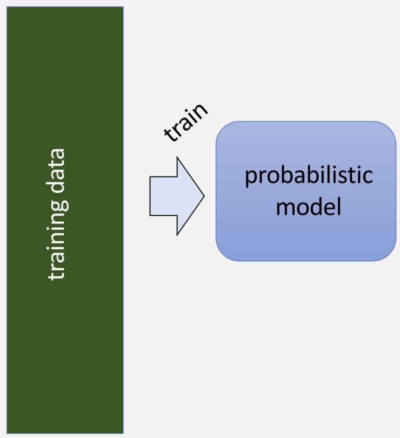


Data Science for Researchers and Scholars

Vasant G. Honavar
Dorothy Foehr Huck and J. Lloyd Huck Chair in Biomedical Data Sciences and Artificial Intelligence
Professor of Data Sciences, Informatics, Computer Science and Engineering, Bioinformatics & Genomics, Public Health Sciences and Neuroscience
Director, Center for Artificial Intelligence Foundations and Scientific Applications
Associate Director, Institute for Computational and Data Sciences
Pennsylvania State University

vhonavar@psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu

Probabilistic Modeling



Model the data with a probabilistic model

specifically, learn p(*features, label*)

p(*features, label*) tells us the probability of co-occurrence of features and the class label

An example: classifying fruit

Training data

Data Samples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana

train

probabilistic model:
p(*features, label*)

Probabilistic models

Probabilistic models define a probability distribution over features and labels:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Probabilistic model vs. classifier

Probabilistic model:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Classifier:

yellow, curved, no leaf, 6oz → probabilistic model: p(*features, label*) → ?

Probabilistic models: classification

Probabilistic models specify a probability distribution over features and labels:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Given an unlabeled example: yellow, curved, no leaf, 6oz predict the label

How do we use a probabilistic model for classification/prediction?

Probabilistic models

Probabilistic models define a probability distribution over features and labels:

yellow, curved, no leaf, 6oz, banana	→	probabilistic model: p(*features, label*)	→	**0.004**
yellow, curved, no leaf, 6oz, apple	→		→	0.00002

- For each label, ask for the probability under the model
- Pick the label with the highest probability

Probabilistic model vs. classifier

Probabilistic model:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Classifier:

yellow, curved, no leaf, 6oz → probabilistic model: p(*features, label*) → banana

Why probabilistic models?

Probabilistic models

Probabilities are nice to work with

- Naturally model uncertainty
- Can be combined in well-understood ways
- Rest on strong mathematical foundations

Probabilistic models: questions

- Which model do we use, i.e. how do we calculate p(*feature, label*)?
- How do train the model, i.e. how to we we estimate the probabilities for the model?
- How do we deal with over-fitting?

Probabilistic models are a special class of ML models

Probabilistic models

- Which model do we use, i.e. how do we obtain p(*feature, label*)?
- How do train the model, i.e. how to we we estimate the probabilities for the model?
- How do we deal with over-fitting?

ML in general

- Which model do we use (linear model, non-parametric)
- How do train the model?
- How do we deal with over-fitting?

Basic steps for probabilistic modeling

Step 1: pick a model

Step 2: figure out how to estimate the probabilities for the model

Step 3: deal with overfitting

Probabilistic models

- Which model do we use, i.e. how do we calculate p(*feature, label*)?
- How do train the model, i.e. how to we we estimate the probabilities for the model?
- How do we deal with over-fitting?

Basic steps for probabilistic modeling

Step 1: pick a model

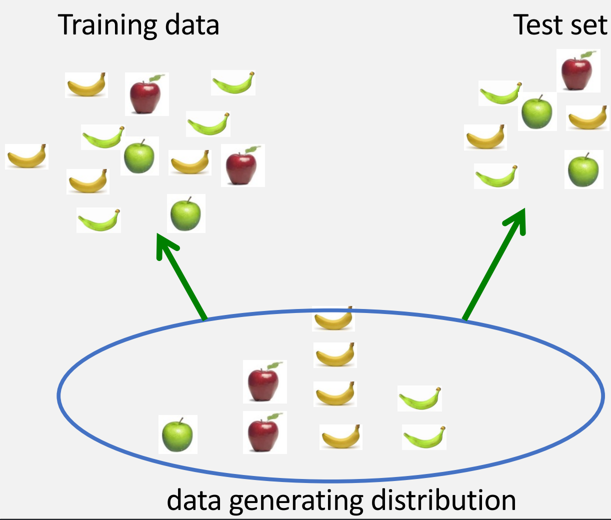
Step 2: figure out how to estimate the probabilities for the model

Step 3 (optional): deal with overfitting

Probabilistic models

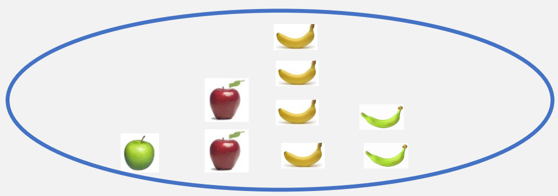
- Which model do we use, i.e. how do we calculate p(*feature, label*)?
- How do train the model, i.e. how to we we estimate the probabilities for the model?
- How do we deal with overfitting?

What was the data generating distribution?



Step 1: picking a model

What we're really trying to do is model the data generating distribution, that is how likely the feature/label combinations are



data generating distribution

Some math

$$p(features, label) = p(x_1, x_2, \ldots, x_m, y)$$

$$= p(y)p(x_1, x_2, \ldots, x_m \mid y)$$

- chain rule!

Some math

$$p(features, label) = p(x_1, x_2, ..., x_m, y)$$

$$= p(y)p(x_1, x_2, ..., x_m \mid y)$$

$$= p(y)p(x_1 \mid y)p(x_2, ..., x_m \mid y, x_1)$$

$$= p(y)p(x_1 \mid y)p(x_2 \mid y, x_1)p(x_3, ..., x_m \mid y, x_1, x_2)$$

$$= p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, ..., x_{i-1})$$

- chain rule!

Decision Theoretic Foundations

- What is an “optimal” classifier?
- How can a classifier assign labels optimally?
- Can we build an optimal classifier?
- Example

Decision theoretic foundations of classification

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?

$$P(error \mid X) = P(\omega_1|X) \text{ if we choose } \omega_2$$
$$= P(\omega_2|X) \text{ if we choose } \omega_1$$

Minimum Error Classification

To minimize classification error

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

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

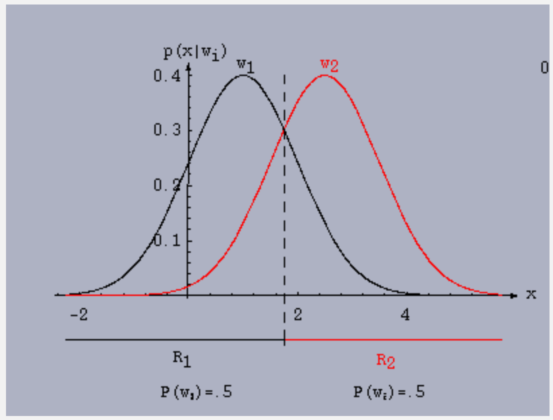
which yields

$$P(error \mid X) = \min\left[P(\omega_1|X), P(\omega_2|X)\right]$$

We have :

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

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



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

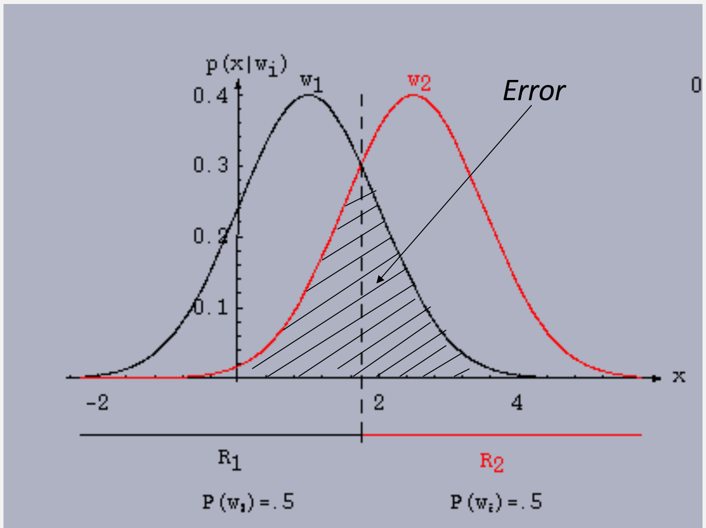
Choose ω_2 if $P(\omega_2|X) > P(\omega_1|X)$ i.e. $X \in R_2$

Optimality of Bayes Decision Rule

We can show that the Bayesian classifier

- is optimal in that it is guaranteed to minimize the probability of misclassification

Optimality of Bayes Decision Rule



Optimality of Bayes Decision Rule

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

Bayes Decision Rule yields Minimum Error Classification

To minimize classification error

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

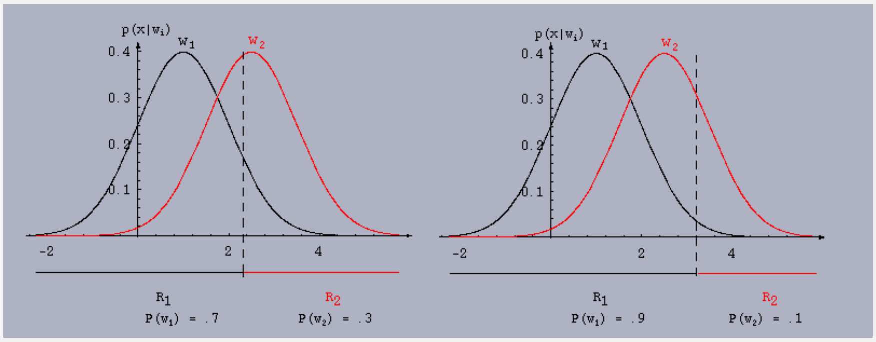
Choose ω_2 if $P(\omega_2|X) > P(\omega_1|X)$

which yields

$$P(error \mid X) = \min\left[P(\omega_1|X), P(\omega_2|X)\right]$$

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 $P(X \mid \omega_1)P(\omega_1) > P(X \mid \omega_2)P(\omega_2)$

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

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

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

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

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

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

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

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

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

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

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

Classification rule that guarantees minimum risk:

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

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

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

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

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

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

Note that $P(\omega_i \mid \mathbf{x}) = \dfrac{P(\mathbf{x}|\omega_i)P(\omega_i)}{P(\mathbf{x})}$

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

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

Otherwise choose ω_2

Multiple classes

Estimate $P(\omega_i|X) = \frac{P(X|\ \omega_i)P(\omega_i)}{P(X)}$

$\omega = argmax\ P(\omega_i|X)$

Assign sample to the most probable class!

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 or may have complex structure
- Need to estimate $P(X|\omega_i)$ from data

Step 1: pick a model

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, ..., x_{i-1})$$

So, far we have made NO assumptions about the data

$$p(x_m \mid y, x_1, x_2, ..., x_{m-1})$$

How many entries would the probability distribution table have if we tried to represent all possible values
Suppose we have 10000 binary features?

Full distribution tables

x_1	x_2	x_3	…	y	p( )
0	0	0	…	0	*
0	0	0	…	1	*
1	0	0	…	0	*
1	0	0	…	1	*
0	1	0	…	0	*
0	1	0	…	1	*
			…		

Problem:

- all possible combination of features
- ~10,000 binary features
- Sample space size: 2^{10000} = ?

Full distribution tables

x_1	x_2	x_3	…	y	p( )
0	0	0	…	0	*
0	0	0	…	1	*
1	0	0	…	0	*
1	0	0	…	1	*
0	1	0	…	0	*
0	1	0	…	1	*
			…		

- Storing a table of that size is impossible
- How are we supposed to learn/estimate each entry in the table?

Step 1: pick a model

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, ..., x_{i-1})$$

So, far we have made NO assumptions about the data

Model selection involves making assumptions about the data

We did this before, e.g. assume the data is linearly separable

These assumptions allow us to represent the data more compactly and to estimate the parameters of the model

ndependence

s are independent if one has nothing to do
er

pendent variables, knowing the value of one
nge the probability distribution of the other
he probability of any individual event)

of the toss of a coin is independent of a roll of a die

of tea in England is independent of the whether or
ass DS Methods course

nt or dependent?

and having a cat-allergy

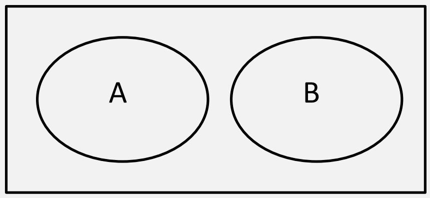
n and driving habits

gevity

ity

Independent variables

How does independence affect our probability equations/properties?

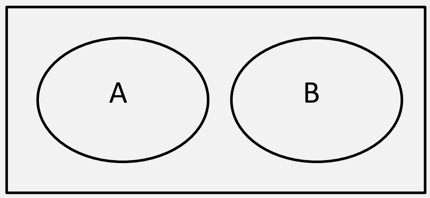


If A and B are independent (written A$\coprod B$)

- $P(A, B) \ = ?$
- $P(A|B) \ = ?$
- $P(B|A) \ = ?$

Independent variables

How does independence affect our probability equations/properties?



If A and B are independent (written …)

- P(A,B) = P(A)P(B)
- P(A|B) = P(A)
- P(B|A) = P(B)

How does independence help us?

Independent variables

If A and B are independent

- P(A,B) = P(A)P(B)
- P(A|B) = P(A)
- P(B|A) = P(B)

- Reduces the storage requirement for the distributions
- Reduces the complexity of the distribution
- Reduces the number of probabilities we need to estimate

ndependence

nts can become independent given certain other

l weight

l weight given genetics

onally independent given C

P(A|C)P(B|C)

P(A|C)

P(B|C)

≠ P(A)P(B)

Conditional Independence

- *X* is conditionally independent of Y given *Z* if the probability distribution governing *X* is independent of the value *of Y* given the value of *Z*:
- $P(X \mid Y, Z) = P(X \mid Z)$ that is,

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

Independence is symmetric

- Assume: $P(X|Y, Z) = P(X|Y)$
- X and Z are independent given Y

$$P(Z \mid X,Y) = \frac{P(X,Y \mid Z)P(Z)}{P(X,Y)}$$ (Bayes' s Rule)

$$= \frac{P(Y \mid Z)P(X \mid Y,Z)P(Z)}{P(X \mid Y)P(Y)}$$ (Chain Rule)

$$= \frac{P(Y \mid Z)P(X \mid Y)P(Z)}{P(X \mid Y)P(Y)}$$ (By Independence Assumption)

$$= \frac{P(Y \mid Z)P(Z)}{P(Y)} = P(Z \mid Y)$$ (Bayes' s Rule)

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{i=1}^{m} p(x_i | y, x_1, \cdots, x_{i-1})$$

$$\forall i \quad p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

What does this mean?

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{i=1}^{m} p(x_i | y, x_1, \cdots, x_{i-1})$$

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

Assumes *i*th feature is independent of the the other features *given the label*

Naïve Bayes assumption

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

- We assume that the ith feature is independent of the the other features *given the label*
- Example: the probability of a word occurring in a review is independent of the other words *given the label*
- For example, the probability of the word “fish” occurring is independent of whether or not “wine” occurs given that the review is about “chardonnay”

Is this assumption true?

Naïve Bayes assumption

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

- For most applications, this is not true!
- For example, the fact that “pinot” occurs will probably make it *more likely* that “noir” occurs (or take a compound phrase like “San Francisco”)
- However, this is often a reasonable approximation:

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) \approx p(x_i \mid y)$$

Naïve Bayes model

$$p(features, label) = p(y)\prod_{i=1}^{m} p(x_i \mid y, x_1, \cdots, x_{i-1})$$

$$= p(y)\prod_{i=1}^{m} p(x_i \mid y)$$ Naïve bayes assumption

$p(x_i|y)$ is the probability of a particular feature value given the label

How do we model this?

- for binary features
- for count features
- for real valued features

Naïve Bayes Classifier

- How to learn $P(X|\omega_i)$?
- Naïve Bayes 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

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 $5(2\times2)=20$ 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 $f : \chi \rightarrow \Omega$
where an instance $X = (X_1, X_2 ... X_n) \in \chi$ is described
in termsof attribute values $X_1 = x_1,\ X_2 = x_2,\ ...\ X_n = x_n$
where $x_i \in Domain(X_i)$

$$\omega_{MAP} = \arg\max_{\omega_j \in \Omega} P(\omega_j \mid X_1 = x_1, X_2 = x_2 ... X_n = x_n)$$

$$= \arg\max_{\omega_j \in \Omega} \frac{P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n \mid \omega_j) P(\omega_j)}{P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n)}$$

$$= \arg\max_{\omega_j \in \Omega} P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n \mid \omega_j) P(\omega_j)$$

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

Naive Bayes Classifier

$$\omega_{MAP} = \arg\max_{\omega_j \in \Omega} P(\omega_j \mid X_1 = x_1, X_2 = x_2 \ldots X_n = x_n)$$

$$= \arg\max_{\omega_j \in \Omega} P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n \mid \omega_j) P(\omega_j)$$

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

$$\omega_{MAP} = \arg\max_{\omega_j \in \Omega} \prod_{i=1}^{n} P\left(X_i = x_i \mid \omega_j\right) P(\omega_j)$$

$$= \omega_{NB}$$

$$= \arg\max_{\omega_j \in \Omega} P(\omega_j) \prod_{i=1}^{n} P\left(X_i = x_i \mid \omega_j\right)$$

Naive Bayes Learner

For each possible value ω_j of Ω,

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

For each possible value a_{i_k} of X_i

$$\hat{P}(X_i = a_{i_k} \mid \omega_j) \leftarrow Estimate\left(P(X_i = a_{i_k} \mid \Omega = \omega_j), D\right)$$

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

$$c(X) = \arg\max_{\omega_j \in \Omega} P(\omega_j) \prod_{i=1}^{n} P(X_i = x_i \mid \omega_j)$$

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

Learning Dating Preferences

Data samples – 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

$P(+) = 5/8$

$P(-) = 3/8$

$P(Height \mid c)$	t	s
+	3/5	2/5
–	2/3	1/3

$P(Hair \mid c)$	d	b	r
+	3/5	2/5	0
–	0	2/3	1/3

$P(Eye \mid c)$	l	w
+	2/5	3/5
–	1	0

Classify (*Height=t, Hair=b, eye=l*)

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

P(X | –) = (2/3)(2/3)(1) = (4/9)

P(+|X) α P(+)P(X|+)=(5/8)(12/125)=0.06

P(-|X) α P(-)P(X|-)=(3/8)(4/9)=0.1667

Classify (*Height=t, Hair=r, eye=w*)

Note the problem with zero probabilities

Solution – Use Laplacian correction

p(x|y)

Binary features:

$$p(x_i \mid y) = \begin{cases} \theta_i & if\ \ x_i = 1 \\ 1-\theta_i & otherwise \end{cases}$$

biased coin toss!

Other features:

Model using an appropriate distribution:

- gaussian (i.e. normal) distribution
- poisson distribution
- multinomial distribution (more on this later)
- …

- for discrete, we could simply do a much larger table, but often that doesn't capture everything we want

Basic steps for probabilistic modeling

Step 1: pick a model

Step 2: figure out how to estimate the probabilities for the model

Step 3 (optional): deal with overfitting

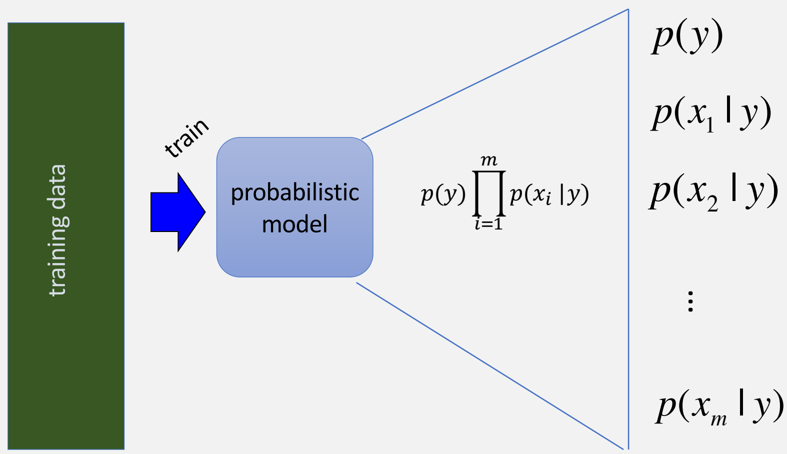
Probabilistic models

Which model do we use, i.e. how do we calculate p(*feature, label*)?

How do train the model, i.e. how to we we estimate the probabilities for the model?

How do we deal with overfitting?

Obtaining probabilities



Estimating probabilities

What is the probability of a Macbook Air review?

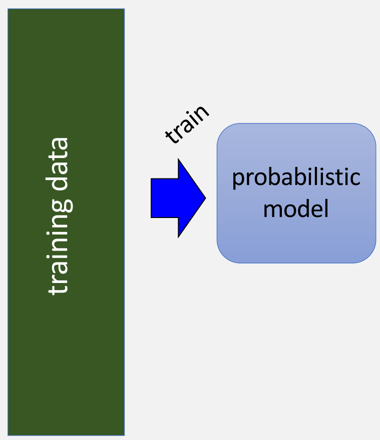
We don't know!

We can ***estimate*** it based on data, though:

$$\frac{\text{number of reviews labeled Macbook Air}}{\text{total number of reviews}}$$

This is called the maximum likelihood estimation. Why?

MLE estimation for NB



$$p(y)\prod_{i=1}^{m} p(x_i \mid y)$$

$$p(y) \qquad\qquad p(x_i \mid y)$$

What are the MLE estimates for these?

Maximum likelihood estimates

$$p(y) = \frac{count(y)}{n}$$

$$\frac{\text{number of examples with label}}{\text{total number of examples}}$$

$$p(x_i \mid y) = \frac{count(x_i, y)}{count(y)}$$

$$\frac{\text{number of examples with the label with feature}}{\text{number of examples with label}}$$

What does training a NB model then involve?
How difficult is this to calculate?

- just involves iterating over the data and aggregating these counts!

Naïve Bayes classification

yellow, curved, no leaf, 6oz, banana → NB Model p(*features, label*) → 0.004

$$p(y)\prod_{i=1}^{m} p(x_i \mid y)$$

Given an unlabeled example:

predict the label

How do we use a probabilistic model for classification/prediction?

Probabilistic models

yellow, curved, no leaf, 6oz, banana

yellow, curved, no leaf, 6oz, apple

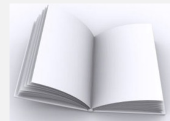
probabilistic model: p(*features*, *label*)

pick largest

$$p(y)\prod_{i=1}^{m} p(x_i \mid y)$$

$$label = \underset{y \in labels}{argmax}\; p(y)\prod_{i=1}^{m} p(x_i \mid y)$$

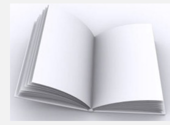
Generative Story



- To classify with a model, we're given an example and we obtain the probability
- We can also ask how a given model would generate a document
- This is the "generative story" for a model
- Looking at the generative story can help understand the model
- We also can use generative stories to help develop a model

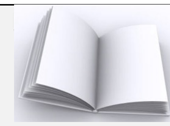
- although we don't generally "generate" a document from a model, it's often useful to look at the generative story of a model (i.e. how the model says a document was generate) to help us understand why the model assigns certain probabilities

NB generative story



$$p(y)\prod_{i=1}^{m} p(x_i \mid y)$$

What is the generative story for the NB model?



NB generative story

$$p(y)\prod_{i=1}^{m} p(x_i\,|y)$$

1. Pick a label according to p(y)
   - roll a biased, m sided die
2. For each binart feature:
   - Flip a biased coin:
     - if heads, include the feature (value 1)
     - if tails, don't include the feature (value 0)

Sample Applications of Naïve Bayes Classifier

Naive Bayes is among the most useful algorithms

- 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

What attributes shall we use to represent text?

Learning to Classify Text

- Target function *Interesting*: Documents $\rightarrow$ {+,-}
- Learning: Use training examples to estimate

  $P(+), P(-), P(d \mid +), P(d \mid -)$

Alternative generative models for documents:

- Represent each document as a 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

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

This would require estimating for each document,

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

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 $P(w_k \mid \boldsymbol{\omega}_j)$ for each word instead of the set of position-specific word occurrence probabilities $P(X_1 = w_k \mid \boldsymbol{\omega}_j) \ldots P(X_{length(d)} = w_k \mid \boldsymbol{\omega}_j)$

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

Learning to Classify Text

With the bag of words representation, we have

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

where n_{kd} is the number of occurences of w_k in document d
(ignoring dependence on length of the document)
We can estimate $P(w_k \mid \omega_j)$ 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
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware comp.windows.x

alt.atheism
soc.religion.christian
talk.religion.misc
talk.politics.mideast
talk.politics.misc
talk.politics.guns

misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey

sci.space
sci.crypt
sci.electronics
sci.med

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 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 to label images

Object → Bag of visual ‘words’

Bags of features for object recognition

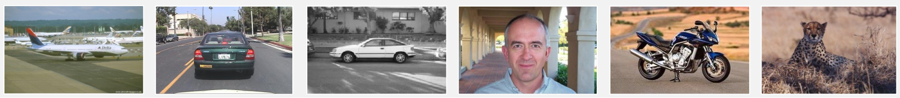
face, flowers, building

- Works pretty well for image-level classification and for recognizing object *instances*

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

Bags of features for object recognition

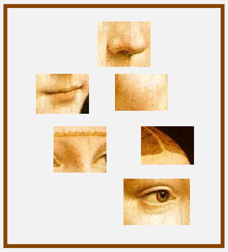
Caltech6 dataset



class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Parts-and-shape model Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Bag of features

1. Extract features



Bag of features

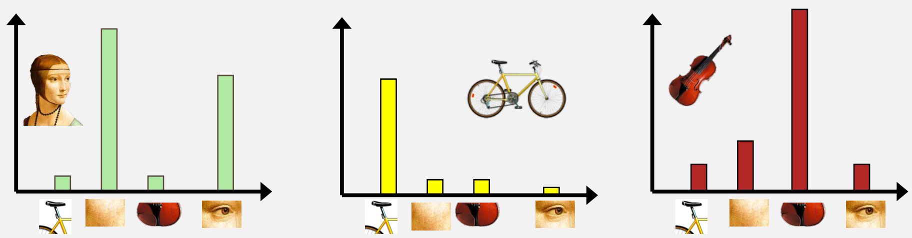
1. Extract features
2. Construct a "visual vocabulary"

Bag of features: outline

1. Extract features
2. Learn "visual vocabulary"
3. Represent images by frequencies of visual words

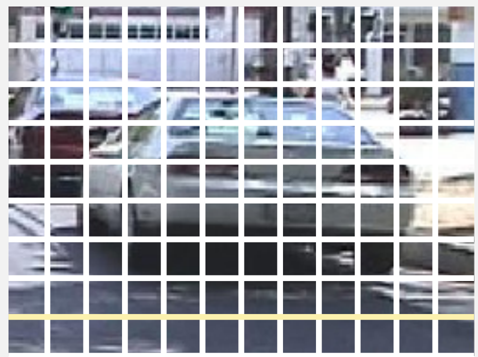
Bag of features

1. Extract features
2. Learn "visual vocabulary"
3. Represent images by frequencies of "visual words" - Note this requires matching image features to visual words



Feature extraction

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005



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

NB decision boundary

$$label = \underset{y \in labels}{argmax} \; p(y) \prod_{i=1}^{m} p(x_i \mid y)$$

What does the decision boundary for NB look like if the features are binary?

Some maths

$$label = \log(\operatorname{argmax}_{y \in labels}\, p(y) \prod_{j=1}^{m} p(x_i \mid y))$$

$$= \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} \log(p(x_i \mid y))$$

$$= \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log(p(x_i \mid y)) + \overline{x}_i \log(1 - p(x_i \mid y))$$

$$\boxed{p(x_i \mid y) = \begin{cases} \theta_i & \text{if } x_i = 1 \\ 1 - \theta_i & \text{otherwise} \end{cases}}$$

Some more maths

$$labels = \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log(p(x_i \mid y)) + \bar{x}_i \log(1 - p(x_i \mid y))$$

$$= \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log(p(x_i \mid y)) + (1 - x_i) \log(1 - p(x_i \mid y))$$

(because x_i are binary)

$$= \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log(p(x_i \mid y)) - x_i \log(1 - p(x_i \mid y) + \log(1 - p(x_i \mid y)$$

$$= \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log\left(\frac{p(x_i \mid y)}{1 - p(x_i \mid y)}\right) + \log(1 - p(x_i \mid y)$$

And…

$$labels = \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log\left(\frac{p(x_i \mid y)}{1 - p(x_i \mid y)}\right) + \log(1 - p(x_i \mid y)$$

$$= \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} \log(1 - p(x_i \mid y)) + \sum_{i=1}^{m} x_i \log\left(\frac{p(x_i \mid y)}{1 - p(x_i \mid y)}\right)$$

What does this look like?

And

$$labels = \operatorname{argmax}_{y \in labels} \log(p(y)) + \sum_{i=1}^{m} x_i \log\left(\frac{p(x_i \mid y)}{1 - p(x_i \mid y)}\right) + \log(1 - p(x_i \mid y))$$

$$= \operatorname{argmax}_{y \in labels} \underbrace{\log(p(y)) + \sum_{i=1}^{m} \log(1 - p(x_i \mid y))}_{b} + \sum_{i=1}^{m} x_i \log\left(\frac{p(x_i \mid y)}{1 - p(x_i \mid y)}\right)$$

b + $x_i * w_i$

$\boxed{\mathbf{w} \,.\, \mathbf{x} + b}$

Linear model !!!

What are the weights?

NB as a linear model

How likely this feature is to be 1 given the label

$$w_i = \log\left(\frac{p(x_i \mid y)}{1 - p(x_i \mid y)}\right)$$

How likely this feature is to be 0 given the label

- low weights indicate there isn't much difference
- larger weights (positive or negative) indicate feature is important

Maximum likelihood estimation

Intuitive

- Sets the probabilities so as to maximize the probability of the training data

Problems?

- Overfitting!
- Amount of data
  - particularly problematic for rare events
- Is our training data representative?

Basic steps for probabilistic modeling

Step 1: pick a model

Step 2: figure out how to estimate the probabilities for the model

Step 3 (optional): deal with overfitting

Probabilistic models

- Which model do we use, i.e. how do we calculate p(*feature, label*)?
- How do train the model, i.e. how to we we estimate the probabilities for the model?
- How do we deal with overfitting?

Priors

Coin1 data: 3 Heads and 1 Tail

Coin2 data: 30 Heads and 10 tails

Coin3 data: 2 Tails

Coin4 data: 497 Heads and 503 tails

If someone asked you what the probability of heads was for each of these coins, what would you say?

Estimation of Probabilities from Small Samples

$$\hat{P}(X_i = a_{i_k} \mid \boldsymbol{\omega}_j) \leftarrow \frac{n_{ji_k} + mp_{ji}}{n_j + m}$$

n_j is the number of training examples of class $\boldsymbol{\omega}_j$

n_{ji_k} = number of training examples of class $\boldsymbol{\omega}_j$ which have attribute value a_{i_k} for attribute X_i

p_{ji} is the prior estimate for $\hat{P}(X_i = a_{i_k} \mid \boldsymbol{\omega}_j)$

m is the weight given to the prior

$$\text{As } n \rightarrow \infty, \hat{P}(X_i = a_{i_k} \mid \boldsymbol{\omega}_j) \rightarrow \frac{n_{ji_k}}{n_j}$$

This is effectively the same as using Dirichlet priors