

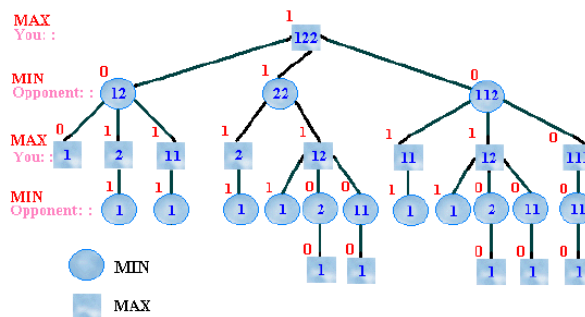
Announcements

- HW 2 due today
- HW 3 assigned; due Monday Oct. 28
 - example playing Nim
- Presentations: About 10 minutes. Can send me slides before class, or put on web page or pen drive.
- Meetings with teams -- schedule

Homework 3

- [Nim demo](#)

Game Tree for (1, 2, 2) NIM



Introduction to Machine Learning

Common types of machine learning tasks

- Classification
 - Output is one of a number of classes (e.g., ‘4’)



- Regression
 - Output is a real value (e.g., ‘\$35/share’)



Classification

- General description of task:
 - Given a *feature vector*, $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$, representing a possible instance of a class, classify x as a positive (1) or negative (0) example.
- General description of learning problem:
 - Given a set of *training examples* $(\mathbf{x}, c(\mathbf{x}))$, where $c(\mathbf{x})$ is the correct classification of \mathbf{x} , construct a hypothesis that will correctly classify the training examples.

Example: Detecting spam

From: =?iso-8859-1?q?james=20ken?= <ja_ken2004@yahoo.fr>
 Subject: URGENT ASSISTANCE
 To: ja_ken2004@yahoo.fr

FROM:JAMES KEN.

ATTN:

Dear Respectful one,

I know this proposal letter may come to you as a surprise considering the fact that we have not had any formal acquaintance before .but all the same I would want you for the sake of God to give this an immediate attention in view of the fact that the security of our live and possession is at stake .

I am Mr JAMES KEN 28 years old from war ravaged SIERRA LEONE but presently domiciled in Abidjan Ivory coast with my sister JANET who is 18 years old .My father Mr KEN who before his untimely assassination by the rebels was the Director of SIERRA LEONE Diamond corporation (SLDC) .He was killed in our government residential house along side two of my other brothers ,two house maids and one government attached security guard fortunately for I, younger sister and mother ,we were on a week end visit to our home town As we got the news of the tragedy .We immediately managed to ran into neighbouring Ivory coast for refuge .But unfortunately .As Fate would have it ,we lost our dear mother (may soulrest in peace) as a result of what the Doctor called cardiac arrest .

As we were coming into this country ,we had some documents of a deposit of \$ 11 700 000 USD (eleven million seven hundred thousand USD) made by my late father in a security and trust company .According to my father, he intended to use this fund for his international business transaction after his tenure in office but was unfortunately murdered .We had located the security company where the money is deposited with the help of an attorney and established ownership .please right now ,with the bitter experiences we had in our country and the war still going on especially in diamond area which incidentally is where we hail from .coupled with the incessant political upheavals and hostilities in this country Ivory coast ,we desire seriously to leave here and live the rest of our life into a more peaceful and politically stable country like yours Hence this proposal and request .We therefore wish you can help us in the following regards :

- 1)To provide us with a good bank account to transfer the money into.
- 2)To help us invest the money into a lucrative business .
- 3)To assist my sister Janet get a college admission to further her education.

Please I know that , this letter may sound strange and incredible to you but the CNN and the BBC African bulletin normally have it as their major news features .Therefore for the sake of God and humanity give an immediate positive consideration and reply to me via our e-mail address. I will willingly agree to any suitable percentage of the money you will propose as your compensation for your assistance with regards to the above .please in view of our sensitive refugee status and as we are still conscious of our father 's enemies .I would like you to give this a highly confidential approach .

Best Regards .
JAMES KEN.

Spamassassin results

```
X-Spam-Report: ---- Start SpamAssassin results
6.70 points, 4 required;
* 0.4 -- BODY: Offers a limited time offer
* 0.1 -- BODY: Free Offer
* 0.4 -- BODY: Stop with the offers, coupons, discounts etc!
* 0.1 -- BODY: HTML font color is red
* 0.1 -- BODY: Image tag with an ID code to identify you
* 2.8 -- BODY: Bayesian classifier says spam probability is 80 to 90%
[score: 0.8204]
* 0.8 -- BODY: HTML font color is green
* 0.3 -- BODY: FONT Size +2 and up or 3 and up
* 0.1 -- BODY: HTML font color not within safe 6x6x6 palette
* 0.1 -- BODY: HTML font color is blue
* 0.3 -- BODY: Message is 70% to 80% HTML
* 1.2 -- Date: is 6 to 12 hours after Received: date
---- End of SpamAssassin results
```

Spamassassin results

```
X-Spam-Report: ---- Start SpamAssassin results
10.70 points, 4 required;
* 0.4 -- BODY: Message is 40% to 50% HTML
* 1.0 -- URI: URL contains username and (optional) password
* 0.8 -- URI: Uses a username in a URL
* 1.2 -- RBL: Received via a relay in dnsbl.njabl.org
[RBL check: found 78.199.241.24.dnsbl.njabl.org.,]
[type: 127.0.0.9]
* 4.3 -- RBL: Received via a relay in list.dsbl.org
[RBL check: found 78.199.241.24.list.dsbl.org.]
* 0.1 -- Message has X-MSMail-Priority, but no X-MimeOLE
* 0.1 -- Message only has text/html MIME parts
* 2.8 -- Forged mail pretending to be from MS Outlook IMO
---- End of SpamAssassin results
```

Spamassassin results

```
X-Spam-Report: ---- Start SpamAssassin results
13.30 points, 4 required;
* 0.7 -- From: ends in numbers
* 1.4 -- Subject is indicative of a Nigerian spam
* 1.5 -- BODY: Nigerian scam key phrase (million dollars)
* 2.7 -- BODY: Contains urgent matter
* 2.8 -- BODY: Bayesian classifier says spam probability is 80 to 90%
[score: 0.8584]
* 0.7 -- Subject is all capitals
* 0.7 -- From: contains an underline and numbers/letters
* 2.8 -- Message body has multiple indications of Nigerian spam
---- End of SpamAssassin results
```

Can we do such classifications with a decision tree?

What features should we use?

- [Spamassasin features](#)

Data sets

- From [UC Irvine Machine Learning Repository](#)

- Inductive learning hypothesis:
 - Any hypothesis that approximates the target concept well over sufficiently large set of training examples will also approximate the concept well over other examples outside of the training set.
- Difference between “induction” and “deduction”?

Key Ingredients for Any Machine Learning Method

- Underlying representation for “hypothesis”, “model”, or “target function”:
 - mathematical expression, bit string, neural network, decision tree, logical description, if-then rules
- Features (or “attributes”)
 - Elements of underlying representation that describe which aspects of problem instances (or training examples) should be used in learning.

Key Ingredients for Any Machine Learning Method

- Space to search:
 - coefficient values, bits, weights, topologies of networks, topologies of trees, possible logic sentences
- Search method:
 - decision-tree learning, gradient descent, genetic algorithm, greedy algorithm, etc.
- Data: Divide into three parts.
 - Training data
 - Used to train the model
 - Validation data
 - Used to select model complexity, to determine when to stop training, or to alter training method
 - Test data
 - Used to evaluate trained model

Types of Machine Learning Methods

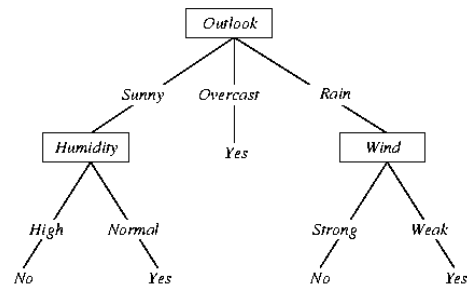
- Supervised
 - provide explicit training examples with correct answers
 - e.g. neural networks with back-propagation
- Unsupervised
 - no feedback information is provided
 - e.g., unsupervised clustering based on similarity

Types of Machine Learning Methods, continued

- “Semi-supervised”
 - feedback information is provided, but is not detailed
- examples:
 - **genetic algorithm**: calculates single-valued “fitness” of individual in population
 - **reinforcement learning**: reinforcement single is single-valued assessment of current state

Decision Trees

<u>Day</u>	<u>Outlook</u>	<u>Temp</u>	<u>Humidity</u>	<u>Wind</u>	<u>PlayGolf</u>
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



- Target concept: “Good days to play golf”
- Example:
<Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong>

Classification?

- Would it be possible to use a “generate-and-test” strategy to find a correct decision tree?
 - I.e., systematically generate all possible decision trees, in order of size, until a correct one is generated.

- Why should we care about finding the simplest (i.e., smallest) correct decision tree?

Decision Tree Induction

- Goal is, given set of training examples, construct decision tree that will classify those training examples correctly (and, hopefully, generalize)
- Original idea of decision trees developed in 1960s by psychologists Hunt, Marin, and Stone, as model of human concept learning. (CLS = “Concept Learning System”)
- In 1970s, AI researcher Ross Quinlan used this idea for AI concept learning:
 - ID3 (“Itemized Dichotomizer 3”), 1979

The Basic Decision Tree Learning Algorithm (ID3)

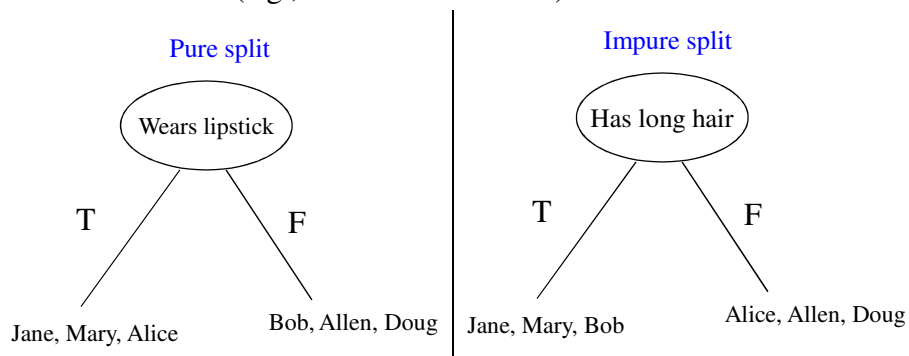
1. Determine which attribute is, by itself, the most useful one for distinguishing the two classes over all the training data. Put it at the root of the tree.
2. Create branches from the root node for each possible value of this attribute. Sort training examples to the appropriate value.
3. At each descendant node, determine which attribute is, by itself, the most useful one for distinguishing the two classes for the corresponding training data. Put that attribute at that node.
4. Go to 2, but for the current node.

Note: This is greedy best-first search with no backtracking

How to determine which attribute is the best classifier for a set of training examples?

“Impurity” of a split

- Perfect (“pure”) split: all instances on a branch belong to same class (e.g., “female” or “male”).



For each node, we want to choose attribute that gives purest split.
But how to measure degree of impurity of a split ?

Entropy

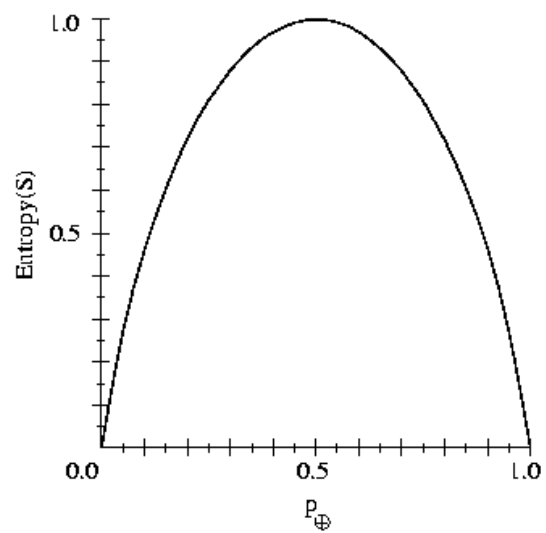
- Let S be a set of training examples.
 p_+ = proportion of positive examples.
 p_- = proportion of negative examples

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- Entropy measures the degree of uniformity or non-uniformity in a collection.
- Roughly measures how predictable collection is on basis of distribution of + and - examples.

Entropy

- What is the entropy of the “play golf” data set?
- When is entropy zero?
- When is entropy maximum, and what is its value?



Information gain

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where $\text{Values}(A)$ is set of possible values for A , and

$$S_v = \{s \in S : A(s) = v\}$$

- What is the attribute with highest information gain in the “play golf” data set?

Demo illustrating operation of ID3
(actually, here “C4.5”)

ID3's Inductive Bias

- Given a set of training examples, there are typically many decision trees consistent with that set.
 - E.g., what would be another decision tree consistent with the example training data?
- Of all these, which one does ID3 construct?
 - First acceptable tree found in greedy search

ID3's Inductive Bias, continued

- Algorithm does two things:
 - Favors shorter trees over longer ones
 - Places attributes with highest information gain closest to root.
- What would be an algorithm that explicitly constructs the shortest possible tree consistent with the training data?
- ID3: Efficient approximation to “find shortest tree” method

Continuous valued attributes

- Original decision trees: Two discrete aspects:
 - Target class (e.g., “*PlayGolf*”) has discrete values
 - Attributes (e.g., “*Humidity*”) have discrete values
- How to incorporate continuous-valued decision attributes?
 - E.g., *Humidity* \in [0,100]

Continuous valued attributes, continued

- Create new attributes, e.g., $Humidity_c$ true if $Humidity < c$, false otherwise.
- How to choose c ?
 - Find c that maximizes information gain.

Continuous valued attributes, continued

- Sort examples according to continuous value of *Humidity*

<i>Humidity:</i>	10	25	55	60	72	85
<i>PlayGolf:</i>	Yes	No	Yes	Yes	No	No
- Find adjacent examples that differ in target classification.
- Choose candidate c as midpoint of the corresponding interval.
 - Can show that optimal c must always lie at such a boundary.
- Then calculate information gain for each candidate c .
- Choose best one.
- Put new attribute $Humidity_c$ in pool of attributes.

Introduction to Bayesian Learning

Bayesianism vs. Frequentism

- Classical probability: **Frequentists**
 - Probability of a particular event is defined relative to its *frequency* in a sample space of events.
 - E.g., probability of “the coin will come up heads on the next trial” is defined relative to the *frequency* of heads in a sample space of coin tosses.
- **Bayesian** probability:
 - Combine measure of “prior” belief you have in a proposition with your subsequent observations of events.
- **Example:** Bayesian can assign probability to statement “There was life on Mars a billion years ago” but frequentist cannot.

In other words:

A **frequentist** believes that a true population mean μ is real, but unknown, and can only be estimated from the data.

Given the distribution of the sample mean, the frequentist constructs a confidence interval (say, 95%) centered at the sample mean.

Problem with this: Is μ in this interval or not?

Can't say that μ is in the interval with probability 0.95, (i.e., μ is in the interval 95% of the time), since μ doesn't have a distribution.

In contrast, a **Bayesian** believes that only the data are real, and the population mean μ is an abstraction.

Some values for μ are more believable than others based on prior beliefs and data.

The Bayesian constructs a probability that μ is in a particular interval about the sample mean, given these prior beliefs and subsequent data. Here "probability" means "degree of believability".

Bayesian Learning

- Let $h \in H$, where H is a hypothesis space
- Question is: given the data D and our prior beliefs, what is the probability that is the correct hypothesis?
- Formally: there are two probability distributions over the hypothesis space:
 - **Prior probability of h :**
 - $P(h)$: Probability that hypothesis h is true given our prior knowledge
 - If no prior knowledge, all $h \in H$ are equally probable
 - **Posterior probability of h :**
 - $P(h|D)$: Probability that hypothesis h is true, given the training set D .

- Also introduces notion of “likelihood” of a particular set of training data D , given that a particular hypothesis h is true:
 - **Likelihood of D :**
 - $P(D | h)$

Example

- Medical diagnosis problem with $H = \{h_1, h_2\}$

h_1 = patient has bone marrow cancer

h_2 = patient does not have bone marrow cancer

- **Prior knowledge:**

0.008 of the population has bone marrow cancer

Thus $P(h_1) = 0.008$, $P(h_2) = 0.992$

- **Likelihood:**

– There is a blood test for bone marrow cancer that has two outcomes, + and -

– If patient has bone marrow cancer, test is positive 98% of the time. If patient does not have bone marrow cancer, test is negative 97% of the time

– Thus $P(+ | h_1) = 0.98$, $P(- | h_1) = 0.02$
 $P(+ | h_2) = 0.03$, $P(- | h_2) = 0.97$

- **Posterior knowledge:**
Blood test is + for this patient.
- **Question:** What is the probability that this patient has bone marrow cancer, given the outcome of the blood test?
I.e., what is $P(h_1 | D)$?
- Bayes theorem tells us how to answer this.

Bayes Theorem

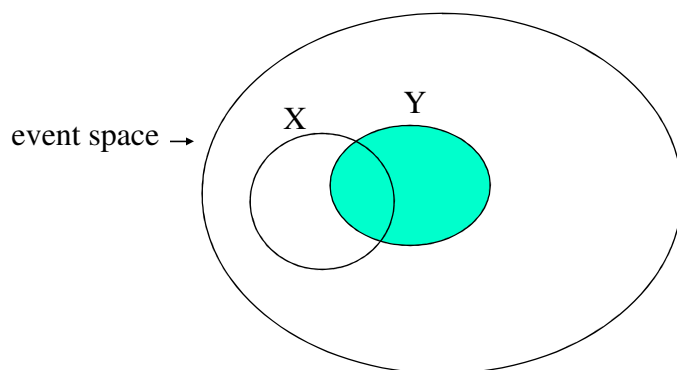
- Theorem:

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

Proof:

Recall the definition of conditional probability:

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}$$



In general:

$$P(X|Y)P(Y) = P(X \cap Y) = P(Y|X)P(X)$$

Thus,

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

MAP Hypothesis

- Back to Bayes theorem:

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

- In machine learning, want to find most probable hypothesis h given D .
- This is called the “maximum a posteriori” (MAP) hypothesis.

Back to our example

- We had:

$$P(h_1) = 0.008, P(h_2) = 0.992$$

$$P(+ | h_1) = 0.98, P(- | h_1) = 0.02$$

$$P(+ | h_2) = 0.03, P(- | h_2) = 0.97$$

- Thus:

$$h_{\text{MAP}} = \operatorname{argmax}_{h \in H} P(D | h)P(h)$$

$$P(+ | \text{cancer})P(\text{cancer}) = (0.98)(0.008) = 0.0078$$

$$P(+ | \neg \text{cancer})P(\neg \text{cancer}) = (0.03)(0.992) = 0.0298$$

$$h_{\text{MAP}} = \neg \text{cancer}$$

- What is $P(\text{cancer} | +)$?

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

So,

$$P(\text{cancer} | +) = \frac{0.0078}{0.0078 + 0.0298} = 0.21$$

$$P(\neg\text{cancer} | +) = \frac{0.0298}{0.0078 + 0.0298} = 0.79$$

These are called the “posterior” probabilities.

Bayes Optimal Classifier

- We’ve answered: “what is the most probable hypothesis given the training data?” (h_{MAP})
- But what we really want to know is, “what is the most probable classification of a new instance x , given the training data?”
- Could just apply h_{MAP} to x .
- But can do better via “ensemble” method (using all $h \in H$)

Example:

Consider $H=\{h_1, h_2, h_3\}$. Assume all have equal *a priori* probability.

Suppose $P(h_1|D) = 0.4$, $P(h_2|D)=0.3$, and $P(h_3|D)=0.3$.

$h_{\text{MAP}} = h_1$. However, ...

Let x be a new instance, with

$$h_1(x) = +$$

$$h_2(x) = -$$

$$h_3(x) = -$$

We have

$$P(x \text{ is a } + \text{ instance} \mid D) = 0.4$$

$$P(x \text{ is a } - \text{ instance} \mid D) = 0.6$$

- In general (assuming binary classes):
 - Let the possible classifications of x be $class \in \{+, -\}$
 - Let $P(class \mid D)$ be the probability that x is classified as $class$ given training data D . Then

$$P(class \mid D) = \sum_{h_i \in H} P(class \mid h_i)P(h_i \mid D)$$

- Optimal classification of x is value $class$ for which $P(class \mid D)$ is maximum:

$$class(x) = \operatorname{argmax}_{class \in \{+, -\}} \sum_{h_i \in H} P(class \mid h_i)P(h_i \mid D)$$

This is called a “Bayes optimal classifier”

- **Back to example:**

$$P(h_1 | D) = 0.4 \quad P(- | h_1) = 0 \quad P(+ | h_1) = 1$$

$$P(h_2 | D) = 0.3 \quad P(- | h_2) = 1 \quad P(+ | h_2) = 0$$

$$P(h_3 | D) = 0.3 \quad P(- | h_3) = 1 \quad P(+ | h_3) = 0$$

Bayes Optimal Classifier, continued

- “No other classification method using the same hypothesis space and same prior knowledge can outperform this method on average.”
- However, method is almost always impractical — why?

Naive Bayes Classifier

Let $f(\mathbf{x})$ be a target function for classification: $f(\mathbf{x}) \in \{+, -\}$.

Let $\mathbf{x} = \langle a_1, a_2, \dots, a_n \rangle$

We want to find the most probable class value, $class_{MAP}$, given the data \mathbf{x} :

$$\begin{aligned} class_{MAP} &= \operatorname{argmax}_{class \in \{+, -\}} P(class | D) \\ &= \operatorname{argmax}_{class \in \{+, -\}} P(class | a_1, a_2, \dots, a_n) \end{aligned}$$

By Bayes Theorem:

$$\begin{aligned} class_{MAP} &= \operatorname{argmax}_{class \in \{+, -\}} \frac{P(a_1, a_2, \dots, a_n | class)P(class)}{P(a_1, a_2, \dots, a_n)} \\ &= \operatorname{argmax}_{class \in \{+, -\}} P(a_1, a_2, \dots, a_n | class)P(class) \end{aligned}$$

$P(class)$ can be estimated from the training data. How?

However, in general, not practical to use training data to estimate $P(a_1, a_2, \dots, a_n | class)$. Why not?

- Naive Bayes classifier: Assume

$$P(a_1, a_2, \dots, a_n | class) = P(a_1 | class)P(a_2 | class) \cdots P(a_n | class)$$

Is this a good assumption?

Given this assumption, here's how to classify an instance

$$\mathbf{x} = \langle a_1, a_2, \dots, a_n \rangle:$$

Naive Bayes classifier:

$$class_{NB} = \operatorname{argmax}_{class \in \{+, -\}} P(class) \prod_i P(a_i | class)$$

Estimate the values of these various probabilities over the training set.

Example

- Use training data from decision tree example to classify the new instance:

$\langle \text{Outlook}=\text{sunny}, \text{Temperature}=\text{cool}, \text{Humidity}=\text{high}, \text{Wind}=\text{strong} \rangle$

Day	Outlook	Temp	Humidity	Wind	PlayGolf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Estimating probabilities

- **Recap:** In previous example, we had a training set and a new example,

<Outlook=sunny, Temperature=cool, Humidity=high,
Wind=strong>

- We needed to estimate

$$P(\text{Wind} = \text{strong} \mid \text{PlayGolf} = \text{yes})$$

$$P(\text{Wind} = \text{strong} \mid \text{PlayGolf} = \text{no})$$

- To do this, we calculated

$$P(\text{Wind} = \text{strong} \mid \text{PlayGolf} = \text{yes}) = \frac{n_c}{n}$$

where n_c is the number of training examples for which **Wind = Strong AND PlayGolf = yes**, and n is the number of training examples for which **PlayGolf = yes**. (Similar for **no** case.)

- But suppose we didn't have training example D6. Then $P(\text{Temp} = \text{cool} \mid \text{PlayGolf} = \text{no}) = 0$. Then

$$P(-) \prod_i P(a_i \mid -) =$$

$$P(-) \times P(\text{outlook} = \text{sunny} \mid -) \times P(\text{outlook} = \text{rain} \mid -) \times \dots \\ \times P(\text{temp} = \text{cool} \mid -) \dots \times P(\text{wind} = \text{weak} \mid -) \\ = 0.$$

This incorrectly gives us zero probability due to sampling error on one attribute.

General problem with this method: If n_c is very small or zero, gives a poor estimate.

Here is one solution:

- Let n be the number of training examples with class $class$. For attribute a with value a_i , let n_c be the number of training examples with $a = a_i$.
- Define m -estimate of probability as:

$$P(a = a_i \mid \text{class} = \text{class}) \approx \frac{n_c + mp}{n + m}$$

where p = prior estimate of $P(a = a_i \mid \text{class} = \text{class})$ and m is the weight given to p .

- Usually, unless you have good prior information, set $p = 1/(\text{number of values of attribute } a)$. The weight m is usually set to 2.

- The idea here is to pretend that we have an extra m training examples with class = $class$, with $p \times m$ of them having $a = a_i$.
- Example:

What is the m -estimate for

$$P(\text{Wind} = \text{strong} \mid \text{PlayGolf} = \text{no})$$

with $m = 2$?

Naive Bayes on Spam data

- Recall Naive Bayes classifier:

Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$, for attributes A_1, A_2, \dots, A_n

Let $C = \{P, N\}$

$$p(C = c \mid \mathbf{x}) \propto P(C = c) \prod_{i=1}^n P(A_i = x_i \mid C = c)$$

- How to deal with continuous-valued attributes?
 - Assume particular probability distribution of classes over values (estimate parameters from training data)
 - Discretize

Simplest discretization method

For each attribute A_i , create k equal-length bins in interval from A_i^{min} to A_i^{max} .

Questions: What should k be? What if some bins have very few instances?

Problem with balance between *discretization bias* and *variance*.

The more bins, the lower the bias, but the higher the variance, due to small sample size.

Alternative simple (but effective) discretization method

(Yang & Webb, 2001)

Let n = number of training examples. For each attribute A_i , create $\approx \sqrt{n}$ bins. Sort values of A_i in ascending order, and put $\approx \sqrt{n}$ of them in each bin.

Don't need m-estimate of probability

This gives good balance between discretization bias and variance.

**Beyond Independence: Conditions for the
Optimality of the Simple Bayesian Classifier
(P. Domingos and M. Pazzani)**

- Recap of naive Bayes classifier:

Let instance $\mathbf{x} = \langle a_1, a_2, \dots, a_n \rangle$. Then:

$$class_{NB} = \operatorname{argmax}_{class \in \{+, -\}} P(class) \prod_i P(a_i | class)$$

Called “naive” because it assumes attributes are independent of one another.

- This paper asks: why does the naive (“simple”) Bayes classifier, SBC, do so well in domains with clearly dependent attributes?

Experiments

- Compare five classification methods on 30 data sets from the UCI ML database.

SBC = Simple Bayesian Classifier

Default = “Choose class with most representatives in data”

C4.5 = Quinlan’s decision tree induction system

PEBLS = An instance-based learning system

CN2 = A rule-induction system

- For SBC, numeric values were discretized into ten equal-length intervals.

Domain	SBC	Default	C4.5	PEBLS	CN2
Audiology	73.9±5.3	21.3±2.5 ⁴	72.5±5.8 ⁶	75.8±5.4 ⁴	71.0±5.1 ²
Annealing	93.5±2.7	76.4±1.8 ¹	91.3±2.3 ³	98.7±0.9 ¹	81.2±5.4 ¹
Breast cancer	68.7±5.4	67.6±7.6 ⁶	70.1±5.6 ⁴	65.8±4.7 ³	67.9±7.1 ⁶
Credit screening	85.2±1.7	57.4±3.8 ¹	85.0±2.0 ⁶	81.3±2.0 ¹	82.0±2.2 ¹
Chess endgames	88.0±1.4	52.0±1.9 ¹	99.2±0.1 ¹	96.9±0.7 ¹	98.1±1.0 ¹
Fima diabetes	74.4±3.0	66.0±2.3 ¹	72.4±2.8 ⁴	71.4±2.4 ¹	73.8±2.7 ⁶
Echocardiogram	66.7±7.4	67.8±6.6 ⁶	65.8±6.2 ⁶	64.1±6.1 ⁵	68.2±7.2 ⁶
Glass	50.4±15.9	31.7±5.5 ¹	66.1±8.4 ¹	65.8±7.3 ¹	63.8±5.5 ¹
Heart disease	83.1±3.2	55.0±3.4 ¹	74.2±4.2 ¹	79.2±3.8 ¹	79.7±2.9 ¹
Hepatitis	81.2±3.7	78.1±3.1 ²	78.7±4.7 ⁴	79.9±6.6 ⁶	80.3±4.2 ⁶
Horse colic	77.8±4.2	63.6±3.9 ¹	83.6±4.1 ¹	76.3±4.4 ⁵	82.5±4.2 ¹
Thyroid disease	97.3±0.7	95.3±0.6 ¹	99.1±0.2 ¹	97.3±0.4 ⁶	98.8±0.4 ¹
Iris	89.0±12.8	26.5±5.2 ¹	93.4±2.4 ⁵	91.7±3.7 ⁶	93.3±3.6 ⁵
Labor neg.	92.6±7.9	65.0±9.5 ¹	79.7±7.1 ¹	91.6±4.3 ⁶	82.1±6.9 ¹
Lung cancer	46.4±14.7	26.8±12.3 ¹	40.9±16.3 ⁶	42.3±17.3 ⁶	38.6±13.5 ⁴
Liver disease	61.8±6.9	58.1±3.4 ³	63.7±4.3 ⁶	60.1±3.6 ⁶	65.0±3.8 ⁴
LED	66.8±5.9	8.0±2.7 ¹	61.2±8.4 ²	55.3±6.1 ¹	58.6±8.1 ¹
Lymphography	81.5±5.6	57.3±5.4 ¹	75.3±4.8 ¹	82.9±5.6 ⁶	78.8±4.9 ³
Post-operative	61.8±9.8	71.2±5.2 ¹	70.2±4.9 ¹	58.8±8.1 ⁶	60.8±8.2 ⁶
Promoters	87.6±6.0	43.1±4.2 ¹	74.3±7.8 ¹	91.7±5.9 ¹	75.9±8.8 ¹
Primary tumor	44.9±5.4	24.6±3.2 ¹	35.9±5.8 ¹	30.9±4.7 ¹	39.8±5.2 ¹
Solar flare	68.0±3.1	25.2±4.4 ¹	70.6±2.9 ¹	67.6±3.5 ⁶	70.4±3.0 ¹
Sonar	24.1±8.7	50.8±7.6 ¹	64.7±7.2 ¹	73.3±7.5 ¹	66.2±7.5 ¹
Soybean	100.0±0.0	30.0±14.3 ¹	95.0±9.0 ³	100.0±0.0 ⁶	96.9±5.9 ³
Splice junctions	95.4±0.6	52.4±1.6 ¹	93.4±0.8 ¹	94.3±0.5 ¹	81.5±5.5 ¹
Voting records	91.2±1.6	60.5±3.1 ¹	96.3±1.3 ¹	94.9±1.2 ¹	95.8±1.6 ¹
Wine	90.9±13.3	36.4±9.9 ¹	91.7±5.6 ⁶	96.9±2.2 ⁴	90.8±4.7 ⁶
Zoology	91.9±3.6	39.4±6.4 ¹	89.6±4.7 ¹	94.6±4.3 ¹	90.6±5.0 ⁵

Table 1: Empirical results: average accuracies and standard deviations. Superscripts denote significance levels for the difference in accuracy between the SBC and the corresponding algorithm, using a one-tailed paired t test: 1 is 0.005, 2 is 0.01, 3 is 0.025, 4 is 0.05, 5 is 0.1, and 6 is above 0.1.

Number of domains in which SBC was more accurate versus less accurate than corresponding classifier

Same as line 1, but significant at 95% confidence

Table 2. Summary of accuracy results.

Measure	SBC	C4.5	PEBLS	CN2
No. wins	-	16-12	15-11	18-10
No. sig. wins	-	12-9	7-9	12-8
Rank	2.32	2.54	2.79	2.68

Average rank over all domains (1 is best in each domain)

Measuring Attribute Dependence

They used a simple, pairwise mutual information measure:

For attributes A_m and A_n dependence is defined as

$$D(A_m, A_n | class) = H(A_m | class) + H(A_n | class) - H(A_m A_n | class)$$

where

$A_m A_n$ is a “derived attribute”, whose values consist of the possible combinations of values of A_m and A_n

$H(A_j | C)$ = “conditional entropy”

$$= \sum_{class \in \{+, -\}} P(class) \sum_k -P(class \wedge A_j = v_k) \log_2 P(class \wedge A_j = v_k)$$

Note: If A_m and A_n are independent, then $D(A_m, A_n | C) = 0$.

Table 3: Empirical measures of attribute dependence.

Domain	Rank	D_{Max}	% Hi.	D_{Avg}
Breast cancer	2	0.548	66.7	0.093
Credit screening	1	0.790	46.7	0.060
Chess endgames	4	0.383	25.0	0.015
Pima diabetes	1	0.483	62.5	0.146
Echocardiogram	3	0.769	85.7	0.360
Glass	4	0.836	100.0	0.363
Heart disease	1	0.388	53.8	0.085
Hepatitis	1	0.589	52.6	0.089
Horse colic	3	0.510	95.5	0.157
Thyroid disease	3	0.516	44.0	0.054
Iris	4	0.731	100.0	0.469
Labor neg.	1	1.189	100.0	0.449
Lung cancer	1	1.226	98.2	0.165
Liver disease	3	0.513	100.0	0.243
LED	1	0.060	0.0	0.025
Lymphography	2	0.410	55.6	0.076
Post-operative	3	0.181	0.0	0.065
Promoters	2	0.394	98.2	0.149
Primary tumor	1	0.098	0.0	0.023
Solar flare	3	0.216	16.7	0.041
Sonar	5	1.471	100.0	0.491
Soybean	1	0.726	31.4	0.016
Splice junctions	1	0.084	0.0	0.017
Voting records	4	0.316	25.0	0.052
Wine	3	0.733	100.0	0.459
Zoology	2	0.150	0.0	0.021

Results:

(1) SBC is more successful than more complex methods, even when there is substantial dependence among attributes.

(2) No correlation between degree of attribute dependence and SBC's rank.

But why????

- The probability calculations are correct only if the independence assumption is correct.
- However, the classification is correct in all cases in which the relative ranking of the two probabilities, as calculated by the SBC, is correct!
- The latter covers a lot more cases than the former.
- Thus, the SBC is effective in many cases in which the independence assumption does not hold.