Naïve Bayes Classifiers

Example: PlayTennis (6.9.1)

Given a new instance, e.g. (Outlook = "sunny", Temperature = "cool", Humidity = "high", Wind = "strong"), we want to compute the most likely hypothesis:

$$v_{NB} = \underset{v_j \varepsilon V}{\arg\max} \left\{ P(v_j) \prod_{i=1}^{n} P(a_i \mid v_j) \right\}$$

P(yes)*P(sunny|yes)*P(cool|yes)*P(high|yes)*P(strong|yes) = 0.005
P(no)*P(sunny|no)*P(cool|no)*P(high|no)*P(strong|no) = 0.021

Therefore v_{NB} = (PlayTennis = "no")

Example: Learning to classify text (6.10)

Why do we want to automatically classify text?

- Rank news articles by order of interest to reader
- Classify web pages by topic
- Filter spam

The naïve Bayes classifier is among the most effective algorithms for this task.

Issue: How do we represent a text document? What attributes should we define? Possibilities include:

- word frequency vector (word : frequency)
- position vector (position : word)

Note that both of these possibilities are, from a linguistic point of view, poor representations. However, "with friends like statistics, who needs linguistics?"

Also, it is difficult to formally define "good" grammar for a language like English. For example, the sentence:

"The horse, raced past the barn, fell."

is technically correct English under a formal grammar, but no one would ever actually say something like that.

Formal definition of text classification parameters

Target concept:	Interesting? : Document → {+, –}
Representation:	Vector of words (one per position)
Learning:	Use training examples to estimate the following probabilities: P(+), P(–), P(document | +), P(document | –)

Naïve Bayes classifier independence assumption

$$P(document \mid v_j) = \prod_{i=1}^{length(document)} P(a_i = w_k \mid v_j)$$

where $P(a_i = w_k \mid v_j)$ is the probability of seeing word k in position i, given hypothesis v_j.

This ignores the possibility of predicting the next word occurrence based on previous words. For example, it is far more likely to see the word “prime” adjacent to the words “minister” or “rib” than it is to see it anywhere else.

We can add one more simplifying assumption: that it does not matter where in a document a given word appears. Formally:

$$\forall i, j, k, m : P(a_i = w_k \mid v_j) = P(a_m = w_k \mid v_j)$$

or, words (attributes) are identically and independently distributed (i.i.d.).

Some classification algorithms will add more tweaks for efficiency, such as ignoring the top 100 most frequent words (“and”, “is”, “the”, etc.) or ignoring words with frequency <3.

Estimating attribute frequencies (6.9.1.1)

The Maximum Likelihood Estimate for word probabilities would be:

$$P(w_k \mid v_j) \approx \frac{n_k}{n}$$

where n_k is the number of occurrences of word *k* in the text, and *n* is the total number of words in the text. However, there are two problems with this estimate. First, it does not account for prior distributions or base rates. Second, if a word does not appear in the training text, this probability will be 0. If this word appears in a test example or a real world example, this probability term will dominate the classifier (everything will be multiplied by 0).

To solve both of these problems, we introduce the m-estimate:

$$P(a_k \mid v_j) = \frac{n_k + mp_k}{n + m}$$

We show the more general form here. For any attribute, we add *m* “virtual” training examples drawn from our prior distribution (frequency p_k for attribute a_k). In the case of our text classification example, we assume uniform priors for every word:

$$p_k = \frac{1}{|Vocabulary|}$$

If we choose $m = |Vocabulary|$, then:

$$P(w_k \mid v_j) = \frac{n_k + 1}{n + |Vocabulary|}$$

We can think of the number of “virtual” examples *m* as a measure of confidence in our prior distribution. In this case, it makes intuitive sense that *m* should be at least the size of the vocabulary.

Notes:
In general, it is very important that your training examples be representative of the general data for the classification problem.

Naïve Bayes algorithms for learning and classifying text (Table 6.2)

LEARN_NAÏVE_BAYES_TEXT(*Examples*, *V*)

Examples is a set of text documents along with their target values.
V is the set of all possible target values.
This function learns the probability terms $P(w_k \mid v_j)$ describing the probability that a randomly drawn word from a document in class v_j will be the English word w_k.
It also learns the class prior probabilities $P(v_j)$.

```
1. Collect all words, punctuation, and other tokens that occur in Examples
      • Vocabulary ← the set of all distinct words and other tokens occurring in any
        text document from Examples
2. Calculate the required $P(v_j)$ and $P(w_k \mid v_j)$ probability terms
      • For each target value $v_j$ in V do
          • $docs_j$ ← the subset of documents from Examples for which the target value
            is $v_j$
          • $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
          • $Text_j$ ← a single document created by concatenating all members of $docs_j$
          • n ← total number of distinct word positions in $Text_j$
          • For each word $w_k$ in Vocabulary do
              • $n_k$ ← number of times word $w_k$ appears in $Text_j$
              • $P(w_k \mid v_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$
```

CLASSIFY_NAÏVE_BAYES_TEXT(*Doc*)

Returns the estimated target value for the document *Doc*.
a_i denotes the word found in the *i*th position within *Doc*.

```
  • positions ← all word positions in Doc that contain tokens found in Vocabulary
  • Return $v_{NB}$, where
```

$$v_{NB} = \arg\max_{v_j \varepsilon V} P(v_j) \prod_{i\, \varepsilon\, positions} P(a_i \mid v_j)$$

Resources on Bayesian noise filtering

http://www.cs.unc.edu/~welch/kalman/
http://www.acfr.usyd.edu.au/technology/bayesianfilter/Bayesian%20Filtering%20Classes.htm

Bayes Nets

Bill Gates said that the competitive advantage of Microsoft was its expertise in Bayes nets. Bayes nets are embedded in a variety of Microsoft products. For more information, go to http://www.gametheory.net/News/Items/063.html.

Basic Definitions

A Bayes net describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions along with a set of conditional probabilities.

Two variables A and B are independent if and only if $P(A \wedge B) = P(A) \times P(B)$. Independence implies irrelevance. If A is independent of B, then B is irrelevant to A.

Exercise: Prove A and B are independent if and only if $P(A \mid B) = P(A)$.

Proof:
By the Product rule, $P(A \wedge B) = P(A \mid B) \times P(B)$.
So $P(A \mid B) = P(A \wedge B) / P(B)$.
$P(A \wedge B) = P(A) \times P(B)$ because A and B are independent.
Therefore, $P(A \mid B) = P(A) \times P(B) / P(B) = P(A)$.

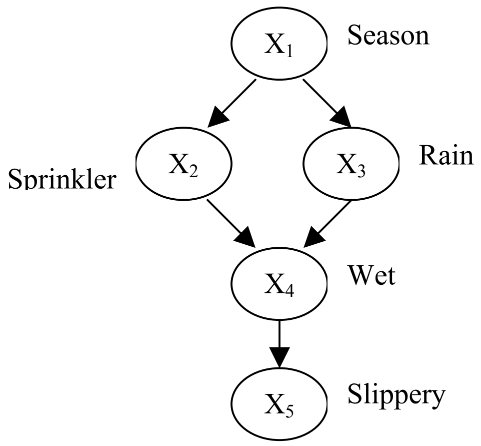
Let X, Y and Z be random variables:

- X is independent of Y if and only if $P(X=x \mid Y=y) = P(X=x)$ for all x and y such that $P(Y=y) > 0$.
- X is independent of Y given Z if and only if $P(X=x \mid Y=y \wedge Z=z) = P(X=x \mid Z=z)$ for all x, y and z such that $P(Y=y \wedge Z=z) > 0$. Notation: $(X \perp\!\!\!\perp Y \mid Z)$
- Intuitively, if X is independent of Y given Z, then Y is irrelevant to X given Z. Once you know the value of Z, the value of Y does not give you any new information about that of X.

Axioms for independence:

- Symmetry: if $(X \perp\!\!\!\perp Y \mid Z)$, then $(Y \perp\!\!\!\perp X \mid Z)$.
- Decomposition: if $(X \perp\!\!\!\perp YW \mid Z)$, then $(X \perp\!\!\!\perp Y \mid Z)$.

Example: Sprinkler



In the graph above, X_5 is independent of X_1 given X_4.

Markovian Parents

In constructing a Bayes net, we look for “direct causes”. A direct cause is a variable that immediately determines the value of another variable. Such direct causes “screen off” other variables that are only indirect causes.

Formally:
Let an ordering of $X_1, X_2, \ldots, X_n$ be given. Consider X_j. Let PA be any subset of $X_1, \ldots, X_{j-1}$. Suppose that $P(X_j \mid PA) = P(X_j \mid X_1, \ldots, X_{j-1})$ and that no subset of PA has this property. Then PA forms the Markovian parents of X_j.

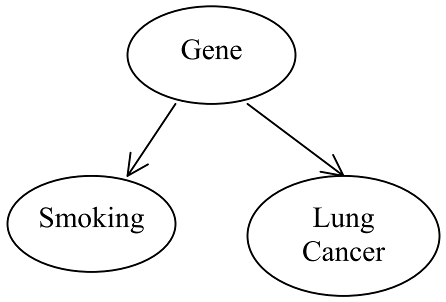
In the sprinkler example, $P(X_4 \mid X_1, X_2, X_3, X_5) = P(X_4 \mid X_2, X_3)$. X_1 is screened off by X_2 and X_3. So X_2 and X_3 are direct causes of X_4, while X_1 is an indirect cause.

Markovian Parents and Bayes Nets

Given an ordering of variables, we can construct a causal graph by drawing arrows between Markovian parents and children. Note that graphs are suitable for drawing the distinction between direct and indirect causes. An arrow directed from variable A to variable B means that A is a direct cause of B.

In a causal graph, given a parent, variable X is independent of all its non-descendents. A graph G is compatible with a probability P if and only if for every node V in G, given Parents(V), V is independent of all its non-descendents.

Example:



In the graph above, Smoking is independent of Lung Cancer given Gene. Without Gene, we cannot tell whether they are independent or there is a causal relation between them. An experiment might help us answer the question.

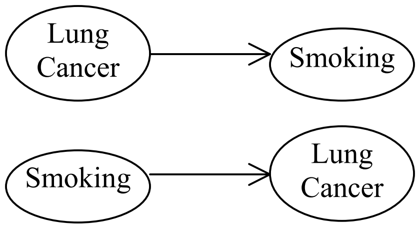
Observational Equivalence

Suppose we can observe the probabilities of various occurrences (e.g. smoking vs. lung cancer). How does probabilities constrain graph?

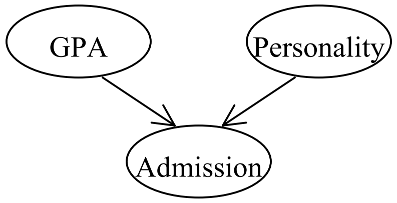
Two causal graphs G_1 and G_2 are compatible with the same probabilities iff:

- G_1 has the same adjacencies as G_2, and
- G_1 has the same v-structures (i.e. colliders) as G_2.

Adjacency means that there is an edge between a pair of variables A and B.
For example, by observing the probabilities of smoking vs. lung cancer, one cannot tell which direction the causal influences goes, so the following two graphs are equivalent.



A v-structure is a system of three variables, say, A, B and C, such that there is an arrow from A to C and there is an arrow from B to C (i.e. $A \rightarrow C \leftarrow B$). Then, C is a collider. For example, in the following causal graph, the variable Admission is a collider such that GPA $\perp\!\!\!\perp$ Personality.



Recall the sprinkler network example, one cannot tell whether X_1 -> X_2 or vice versa. But can tell that X_2 -> X_4 and X_4 -> X_5 (because if you reverse the arrow between X_2 and X_4, you keep the same adjacencies but can tell that the new graph has the different v-structure, and same if you reverse the arrow between X_4 and X_5).

Note: In general, you cannot always tell in machine learning what the correct hypothesis is even if you have all possible data. You need more assumptions or other kinds of data.

Inferring Casual Structure: The IC Algorithm

- Assume a *stable*[1] probability distribution P.
- Find a *minimal*[2] graph for P with as many edges directed as possible.
- General idea: First find variables that are "directly causally related". Connect those. Add arrows as far as possible.

The IC Algorithm

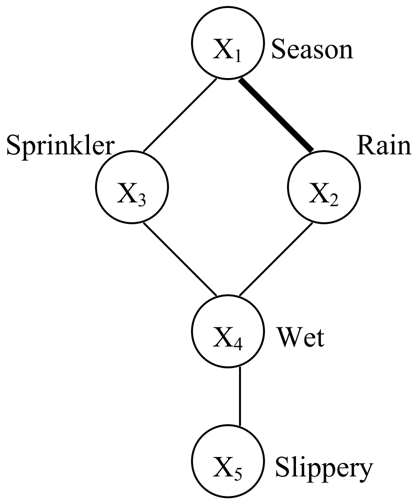
1. For each pair of variables *X* and *Y*, look for a "screen off" set S(X,Y) such that $X \perp\!\!\!\perp Y | S(X,Y)$ holds. If there is no such set, add an undirected edge between X and Y.
2. For each pair X,Y with a common neighbour Z, check if Z is part of a "screening off" set S(X,Y). If not, make Z a common consequence of X,Y.
3. Orient edges without creating cycles or v-structures.

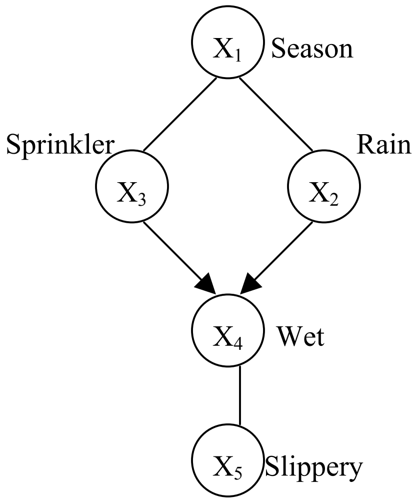
<u>Simple Illustration: The sprinkler example</u>

In the first step of the algorithm, you want to check if there's edge to be added between any two variables. Let's consider X_1 and X_2. If there is no way to screen off the causal effect between them (i.e. there does not exist any $S(X_1, X_2)$ such that $X_1 \perp\!\!\!\perp X_2|S(X_1, X_2)$), then add a direct link there. Follow this manner to add other edges (see the left figure below).

To see how the second step works, let's consider X_2 and X_3, and their common neighbours are nodes X_1 and X_4. So, fix X_1 and X_4 respectively, and check whether each of those is in the "screen off" set $S(X_2, X_3)$. Since we know that $X_3 \perp\!\!\!\perp X_2 | X_1$, but NOT $[X_3 \perp\!\!\!\perp X_2 | X_4]$, we can add the two arrows to make X_3 and X_2 both parents of X_4 (see the right figure below).

Finally, you can orient edges subject to no cycles or new v-structures.





[1] A distribution P is stable iff there is a graph G such that $(X \perp\!\!\!\perp Y |Z)$ in P iff X and Y are d-separated given Z in G. We say that two variables X and Y are d-separated if all the paths between them are blocked given the conditioning set Z (which can be empty).

[2] A graph G is minimal for a probability distribution P iff
- G is compatible with P, and
- no subgraph of G is compatible with P.