Probability Theory Primer plus Naïve Bayes all ways

James G. Shanahan[2]

[1]Church and Duncan Group, [2]iSchool UC Berkeley, CA,

EMAIL: James_DOT_Shanahan_AT_gmail_DOT_com

Lecture 5

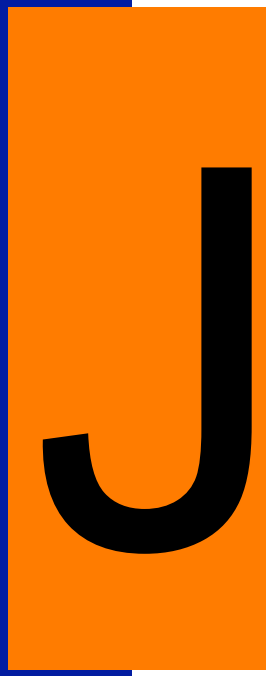
June 14, 2016

References and Resources

- **Manning, Raghavan, Schutz, IRBook**
- **Sebastian Raschka - Naive Bayes & Text Classification**
  - http://sebastianraschka.com/Articles/2014_naive_bayes_1.html

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**
  - Multinomial Naïve Bayes in Hadoop



Probability theory 1/3

- **Probability theory is the branch of mathematics concerned with probability, the analysis of random phenomena.**
- **The central objects of probability theory are random variables, stochastic processes, and events:**
  - mathematical abstractions of non-deterministic events or measured quantities that may either be single occurrences or evolve over time in an apparently random fashion.

Probability theory 2/3

- **It is not possible to predict precisely results of random events.**
- **However, if a sequence of individual events, such as coin flipping or the roll of dice, is influenced by other factors, such as friction, it will exhibit certain patterns, which can be studied and predicted.**
- **Two representative mathematical results describing such patterns are the**
  - (1) law of large numbers (LLN) and the (see next slide)
  - (2) central limit theorem. (central tendencies) (see next slide)

Probability theory 3/3

- **As a mathematical foundation for statistics, probability theory is essential to many human activities that involve quantitative analysis of large sets of data.**
- **Many applications**
  - Machine learning
  - statistical mechanics: Methods of probability theory also apply to descriptions of complex systems given only partial knowledge of their state, as in statistical mechanics.
  - quantum mechanics: A great discovery of twentieth century physics was the probabilistic nature of physical phenomena at atomic scales, described in quantum mechanics.

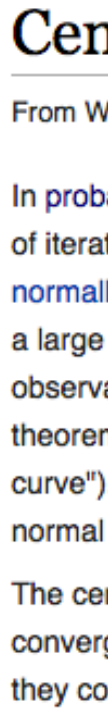
law of large numbers

- In probability theory, the law of large numbers (LLN) is a theorem that describes the result of performing the same experiment a large number of times.
- According to the law, the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.

CLT: mean will be approximately normally distributed

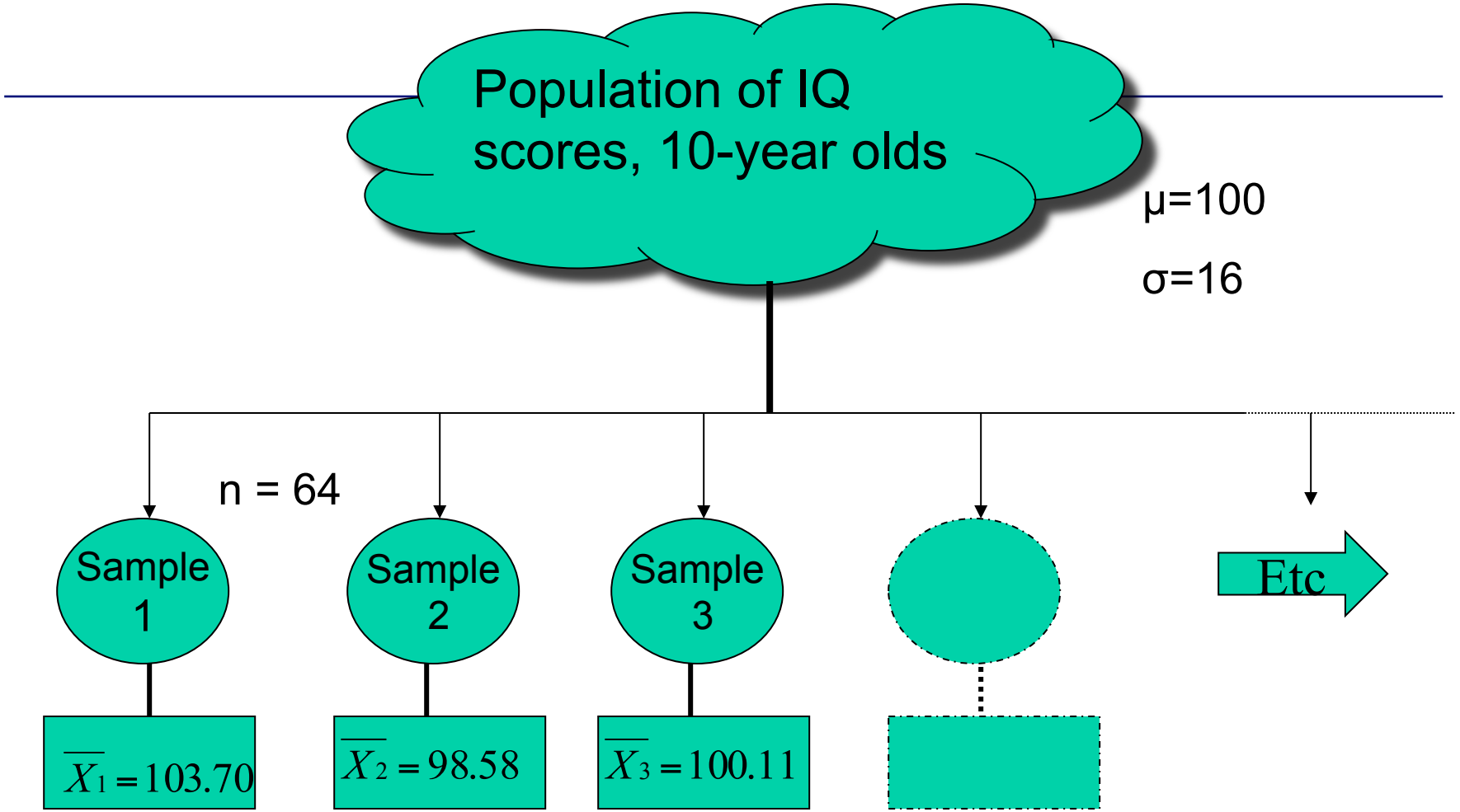
- In probability theory, the central limit theorem (CLT) states that, given certain conditions, the arithmetic mean of a sufficiently large number of iterates of independent random variables, each with a well-defined expected value and well-defined variance, will be approximately normally distributed, regardless of the underlying distribution.[1][2]

- To illustrate what this means, suppose that a sample is obtained containing a large number of observations, each observation being randomly generated in a way that does not depend on the values of the other observations, and that the arithmetic average of the observed values is computed.
- If this procedure is performed many times, the central limit theorem says that the computed values of the average will be distributed according to the normal distribution (commonly known as a "bell curve").
- A simple example of this is that if one flips a coin many times, the probability of getting a given number of heads should follow a norm curve, with mean equal to half the total number of flips.


Cen
From W
In prob
of iterat
normall
a large
observa
theorer
curve")
normal
The cer
converg
they co

Standard Error

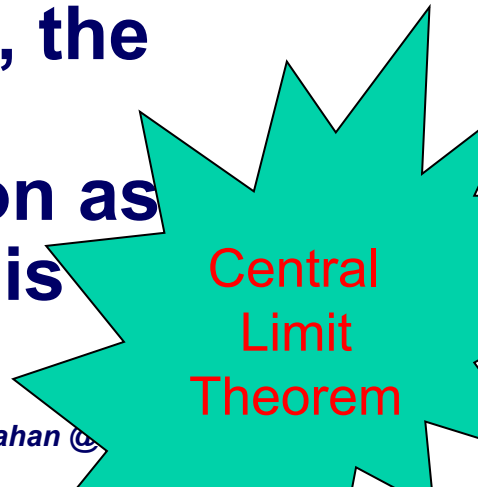
- The standard error is the standard deviation of the sampling distribution of a statistic.[1]
- The term may also be used to refer to an estimate of that standard deviation, derived from a particular sample used to compute the estimate.
- For example, the sample mean is the usual estimator of a population mean. However, different samples drawn from that same population would in general have different values of the sample mean. The standard error of the mean (i.e., of using the sample mean as a method of estimating the population mean) is the standard deviation of those sample means over all possible samples (of a given size) drawn from the population. Secondly, the standard error of the mean can refer to an estimate of that standard deviation, computed from the sample of data being analyzed at the time.



Is sample 2 a likely representation of our population?

Distribution of Sample Means

1. The mean of a sampling distribution is identical to mean of raw scores in the population (μ)
2. If the population is Normal, the distribution of sample means is also Normal
3. If the population is not Normal, the distribution of sample means approaches Normal distribution as the size of sample on which it is based gets larger

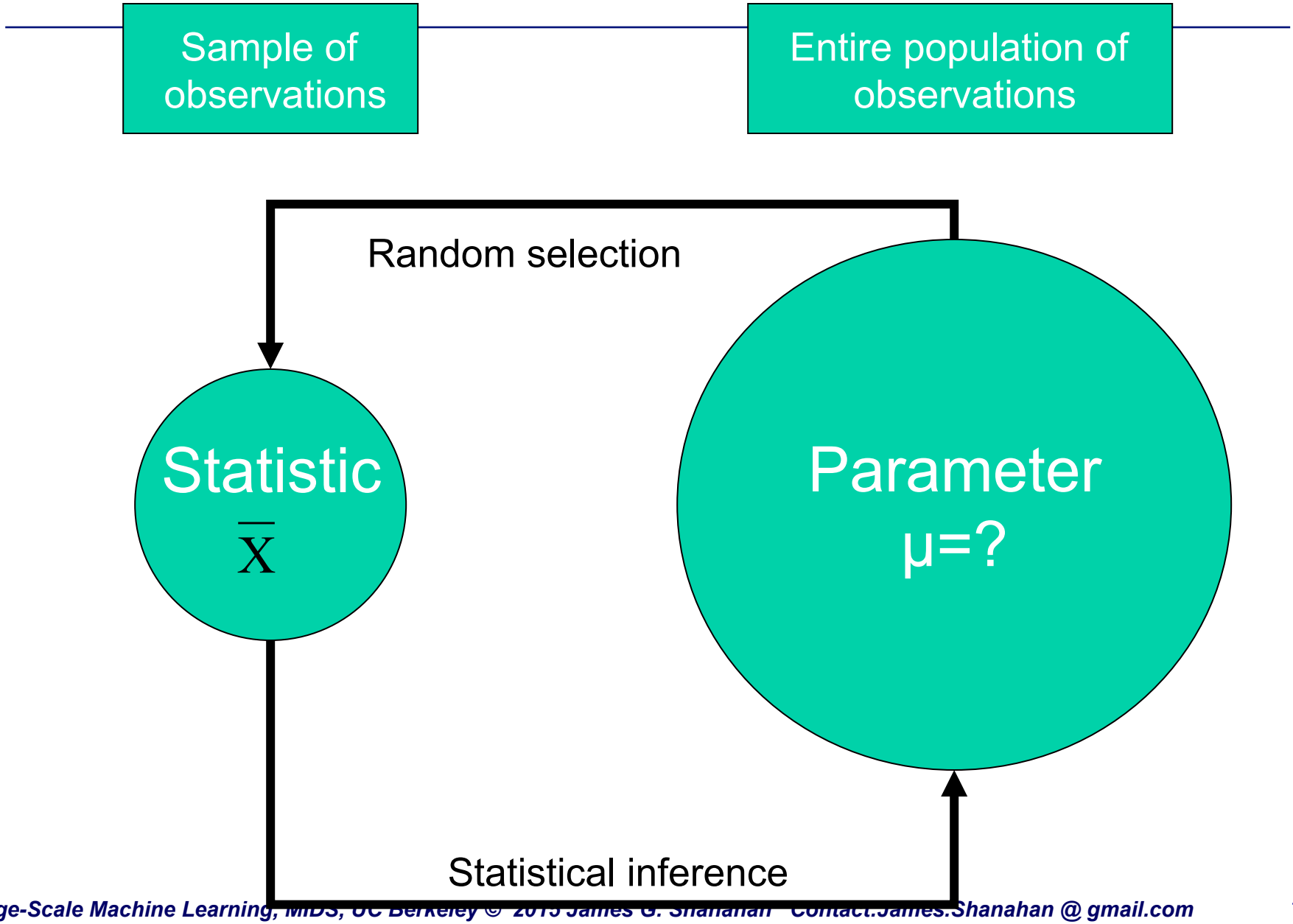


Standard Error of the Mean

- The standard deviation of means in a sampling distribution is known as the standard error of the mean.
- It can be calculated from the standard deviation of observations

$$S_{\bar{x}} = \frac{s}{\sqrt{n}} \qquad \text{Standard Error of } \bar{X} : \sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

3. The larger our sample size, the smaller our standard error


Sample of observations
Entire population of observations
Random selection
Statistic
$\bar{x}$
Parameter
μ=?
Statistical inference

Estimation Procedures

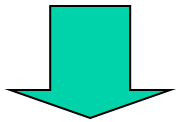
- **Point estimates**
  - For example mean of a sample of 25 patients
    - No information regarding probability of accuracy
  - Interval estimates
  - Estimate a range of values that is likely
    - Confidence interval between two limit values
      - The degree of confidence depends on the probability of including the population mean

$$95\%\ \mathrm{CI} = \bar{X} \pm 1.96\,\sigma_{\bar{x}}$$

$$99\%\ \mathrm{CI} = \bar{X} \pm 2.58\,\sigma_{\bar{x}}$$

When Sample size is small …

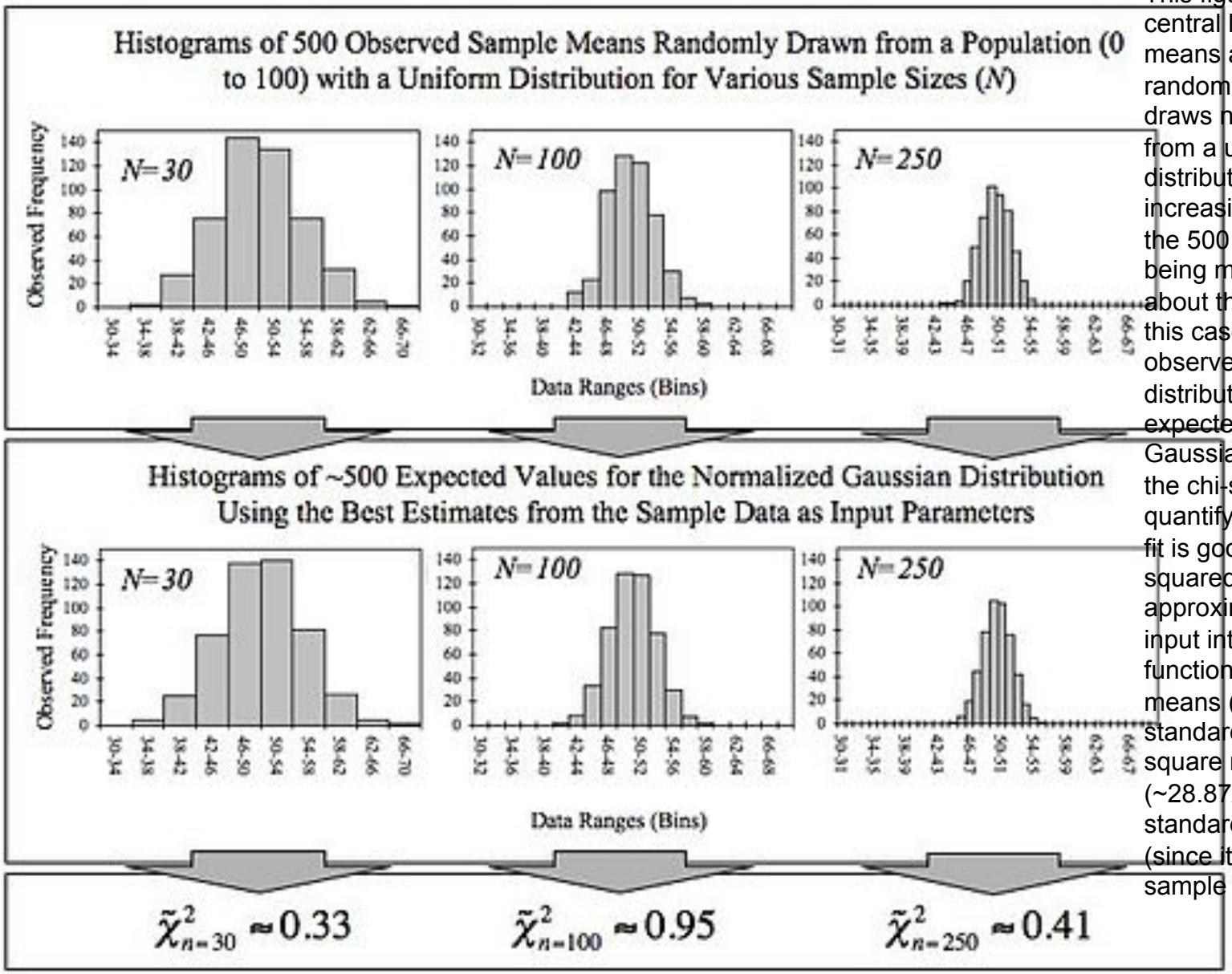
$$95\%\ \mathrm{CI} = \bar{X} \pm 1.96\ \sigma_{\bar{x}}$$



$$95\%\ \mathrm{CI} = \bar{X} \pm t\ S_{\bar{x}}$$

A constant from
Student t Distribution
that depends on confidence
interval and sample size

Uniform distribution [0,100], Pop Mean is 50.



This figure demonstrates the central limit theorem. The sample means are generated using a random number generator, which draws numbers between 0 and 100 from a uniform probability distribution. It illustrates that increasing sample sizes result in the 500 measured sample means being more closely distributed about the population mean (50 in this case). It also compares the observed distributions with the distributions that would be expected for a normalized Gaussian distribution, and shows the chi-squared values that quantify the goodness of the fit (the fit is good if the reduced chi-squared value is less than or approximately equal to one). The input into the normalized Gaussian function is the mean of sample means (~50) and the mean sample standard deviation divided by the square root of the sample size (~28.87/√n), which is called the standard deviation of the mean (since it refers to the spread of sample means).

Notation

- Proposition - statement or assertion about a state of the world
- Variable X is a set of mutually exclusive propositions x_i
- Variables – upper-case
- Propositions – lowercase
  - **Example (X=x, Y=y, Z=z)**
  - **Shortened: (x,y,z)**
- Sets of variables – bold
  - Example: (**X, Y, Z**)
- Latent/Hidden variable – states are inferred but never observed directly

From Axioms: deduce theorems and propositions

- **Math**
  - One strategy in mathematics is to start with a few statements, then build up more mathematics from these statements.
  - The beginning statements are known as axioms.
    - An axiom is typically something that is mathematically self evident.
    - From a relatively short list of axioms deductive logic is used to prove other statements, called theorems or propositions.
- **Probability Thoery has 3 axioms**
  - The area of mathematics known as probability is no different.
  - Underlying probability is a handful of axioms from which we can derive all sorts of results. But what are these probability axioms?
  - Probability can be reduced to three axioms.
  - It presupposes that we have a set of outcomes called the sample space S comprised of subsets called events $E_1, E_2, \ldots, E_n$ and a way of assigning a probability to any event E. The probability of the event E is denoted by P(E).

Axioms of Probabilities

http://statistics.about.com/od/Mathstat/a/what-is-the-power-set.htm

Axiom 1 : $0 \le P(A) \le 1$

Axiom 2: $P(Sure\ Proposition) = 1$

Axiom 3: $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

Marginal probability: $P(A) = P(A, B) + P(A, \neg B)$

$$P(A) = \sum_i P(A, B_i)$$

The first axiom of probability is that the probability of any event is a nonnegative real number. This means that the smallest that a probability can ever be is zero, and that it cannot be infinite.

The third axiom of probability deals with mutually exclusive events. If E1 and E2 are mutually exclusive, meaning that they have an empty intersection and we use U to denote the union, then P(E1 U E2) = P(E1) + P(E2).

Axiom Applications: Pr(an impossible event)

The three axioms set an upper bound for the probability of any event. We denote the complement of the event E by EC. From set theory E and EC have empty intersection and are mutually exclusive. Furthermore E U EC = S, the entire sample space.

These facts, combined with the axioms give us:

$1 = P(S) = P(E \cup E^C) = P(E) + P(E^C)$.

We rearrange the above equation and see that $P(E) = 1 - P(E^C)$. Since we know that probabilities must be nonnegative, we now have that an upper bound for the probability of any event is 1.

By rearranging the formula again we have $P(E^C) = 1 - P(E)$. We also can deduce from this formula that the probability of an event not occurring is one minus the probability that it does occur.

The above equation also provides us a way to calculate the probability of the impossible event, denoted by the empty set. To see this, recall that the empty set is the complement of the universal set, in this case S^C. Since $1 = P(S) + P(S^C) = 1 + P(S^C)$, by algebra we have $P(S^C) = 0$.

Complement space

2nd Axiom: states that the probability of all the events, i.e., the probability of the entire sample space is 1.
Mathematically, if S represents the Sample space, then P(S)=1.
This means that there are no events outside the sample space and it includes all possible events in it.

$$P(A) + P(\neg A) = 1$$

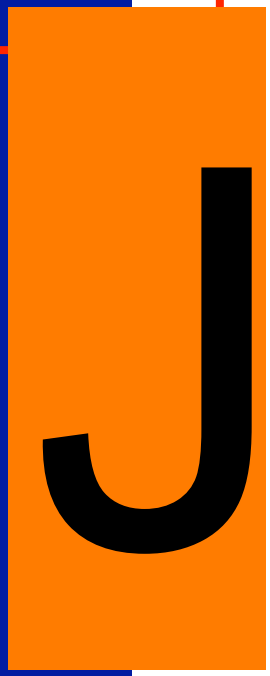
$$P(A \mid B)$$

Pr(A) → P(A|B)

- Belief in A under the assumption that B is known with absolute certainty
- A is conditioned on B

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Sample Space: possible outcomes

- **In probability theory, the sample space of an experiment or random trial is the set of all possible outcomes or results of that experiment.**
- **A sample space is usually denoted using set notation, and the possible outcomes are listed as elements in the set. It is common to refer to a sample space by the labels S, Ω, or U (for "universal set").**
- **E.g.,**
  - For example, if the experiment is tossing a coin, the sample space is typically the set {head, tail}.
  - For tossing two coins, the corresponding sample space would be {(head,head), (head,tail), (tail,head), (tail,tail)}.
  - For tossing a single six-sided die, the typical sample space is {1, 2, 3, 4, 5, 6} (in which the result of interest is the number of pips facing up).

{head, tail}

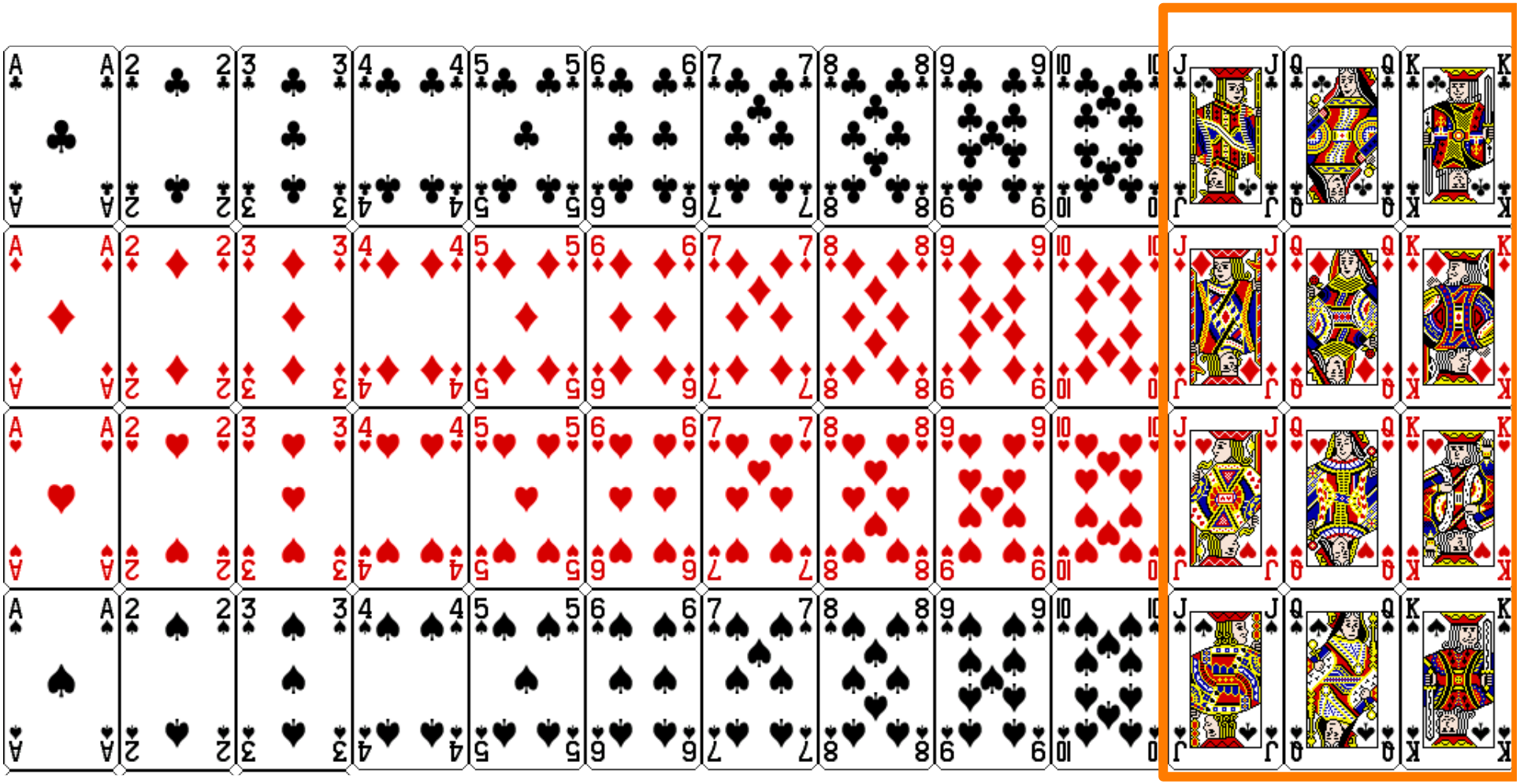
{(head,head), (head,tail), (tail,head), (tail,tail)}

Multiple sample spaces

- **For many experiments, there may be more than one plausible sample space available, depending on what result is of interest to the experimenter.**
- **For example, when drawing a card from a standard deck of fifty-two playing cards,**
  - **Rank sample space**
    - one possibility for the sample space could be the various ranks (Ace through King),
  - **Suits sample space**
    - while another could be the suits (clubs, diamonds, hearts, or spades).
  - A more complete description of outcomes, however, could specify both the denomination and the suit, and a sample space describing each individual card can be constructed as the Cartesian product of the two sample spaces noted above (this space would contain fifty-two equally likely outcomes).
  - Still other sample spaces are possible, such as {right-side up, up-side down} if some cards have been flipped when shuffling.

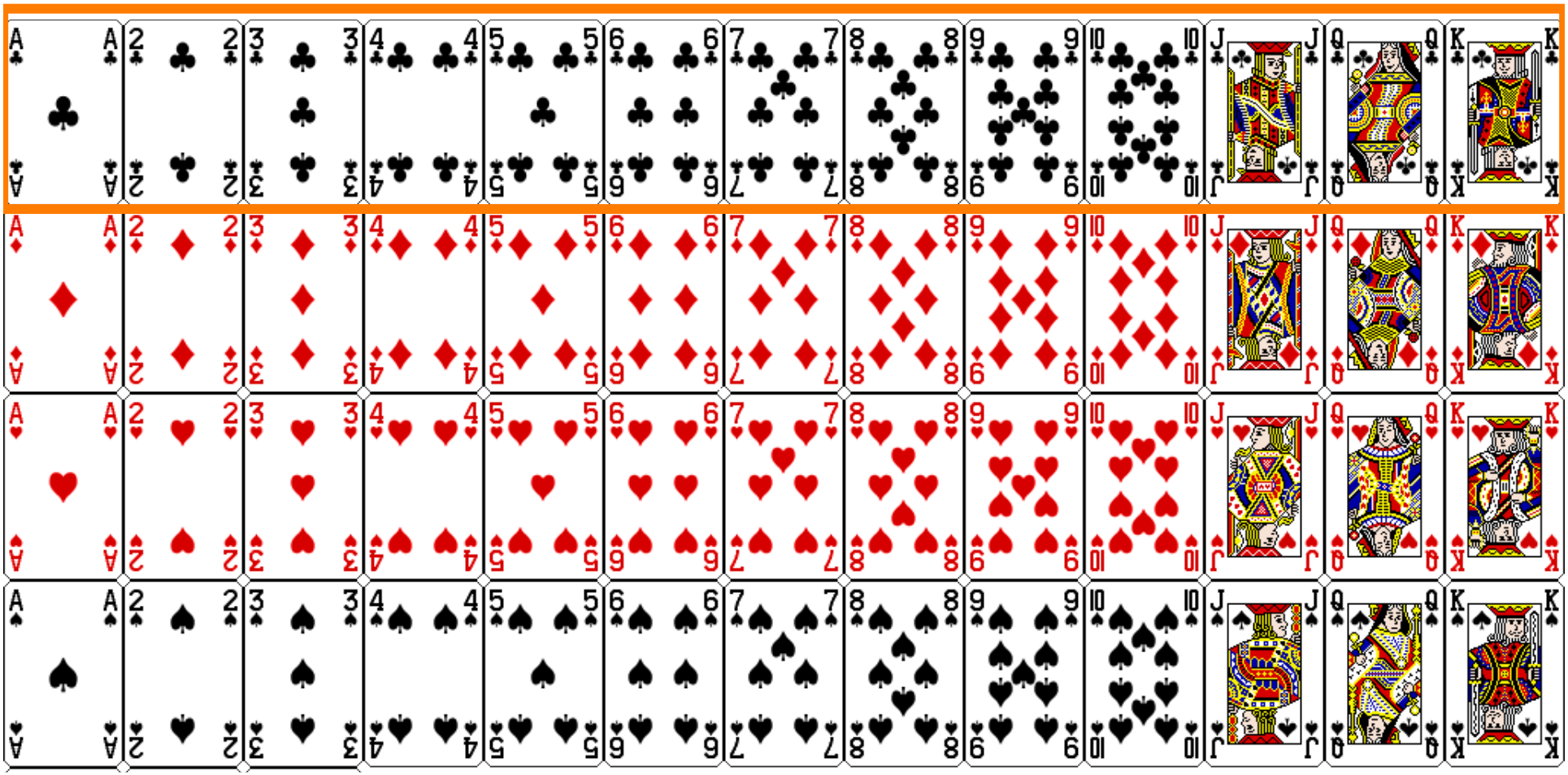
Sample space

Royal Cards = {Jack, Queen, King}



Sample space

Clubs

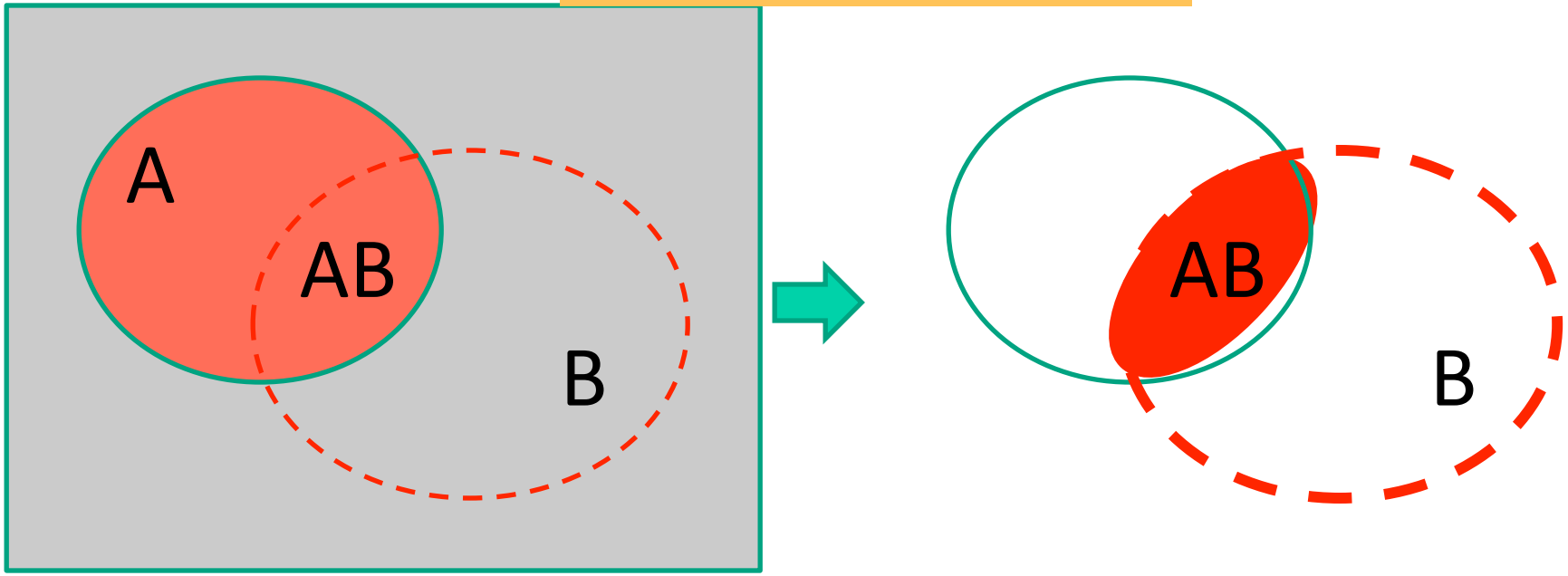


Calculating Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

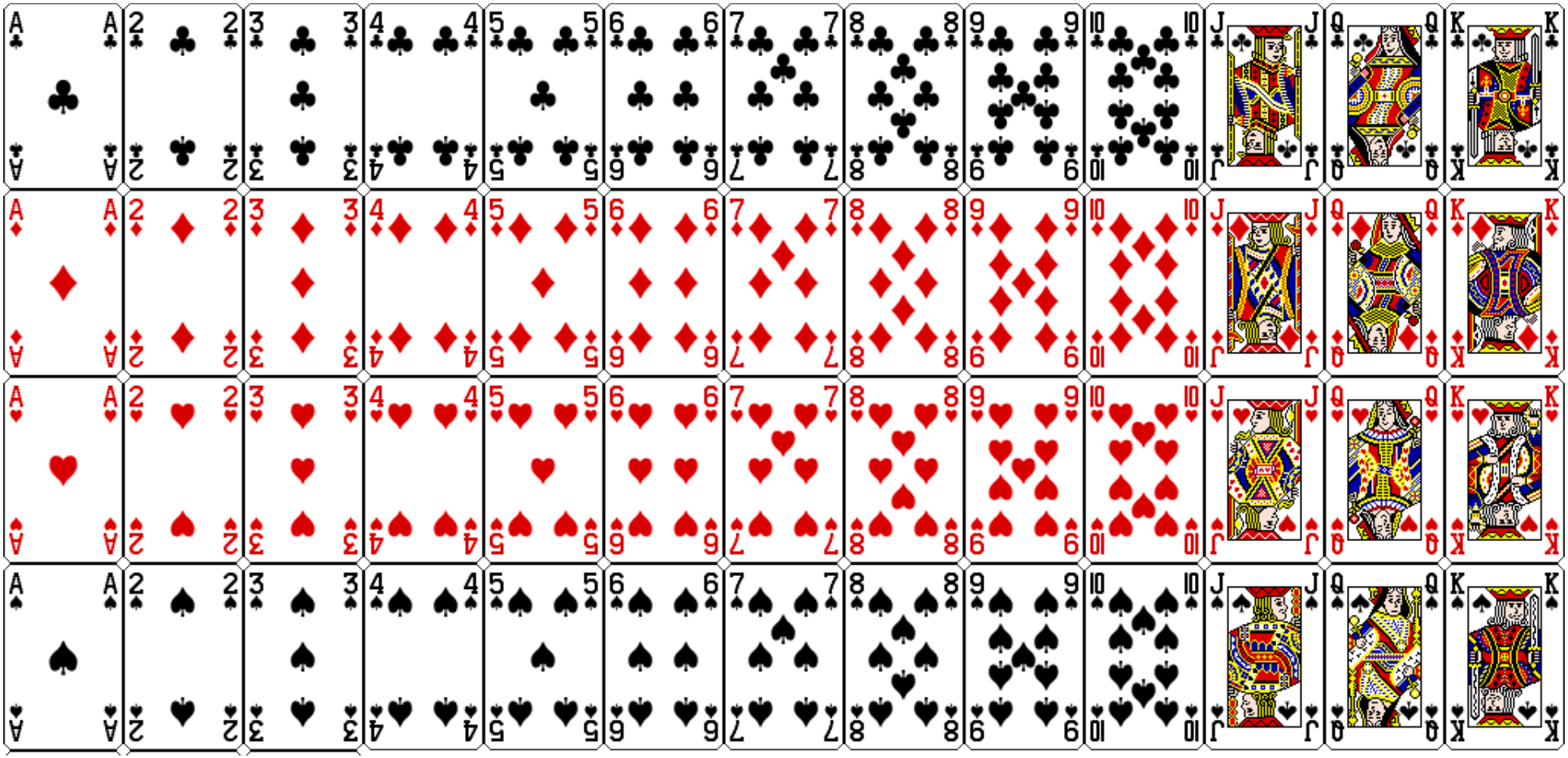
Conditional probability can be seen to be the probability with respect to a reduced sample space. We can illustrate the conditional probability with the Venn diagram.

Pr(A) → P(A|B)

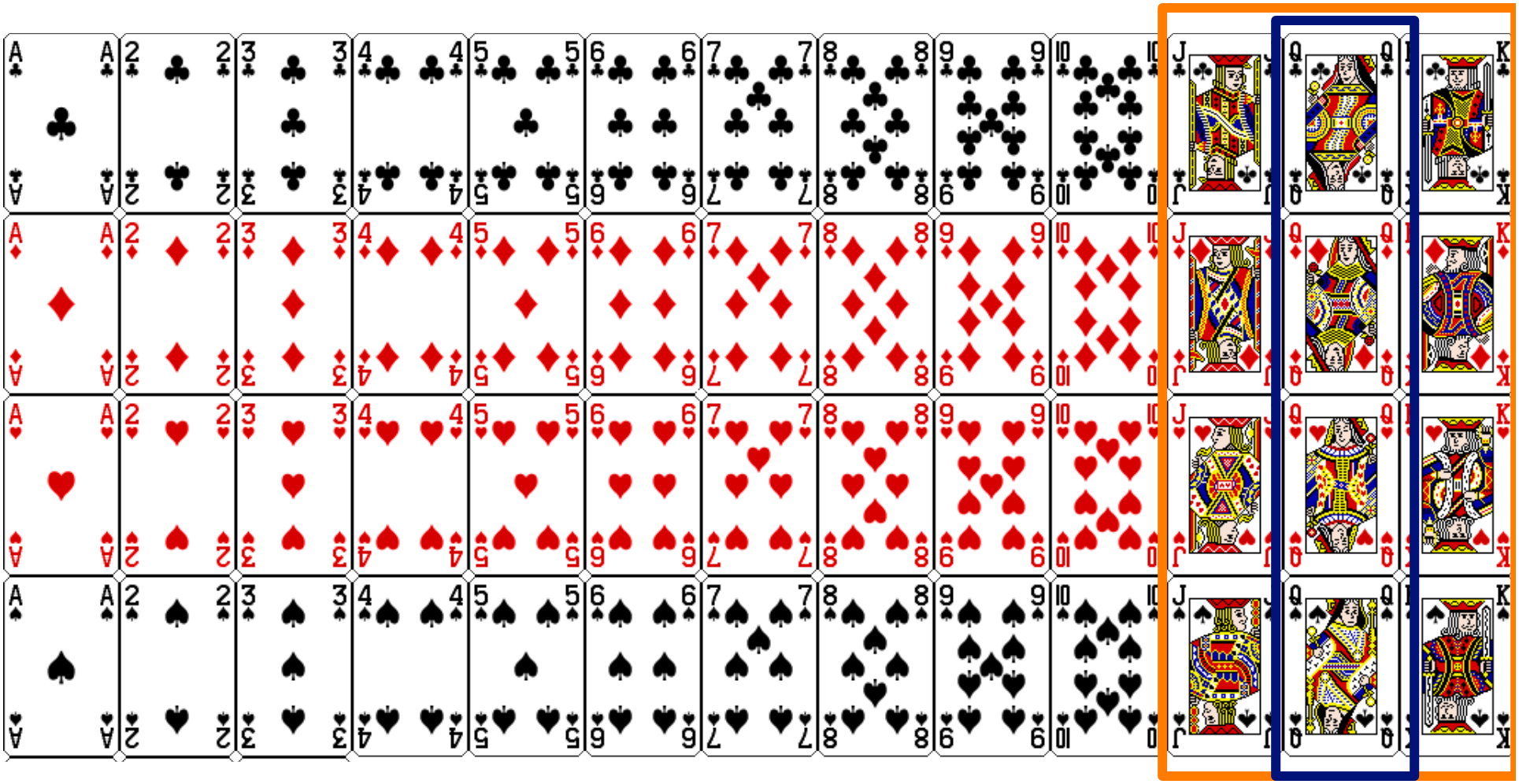


Conditional probability example

Royal Cards = {Jack, Queen, King}

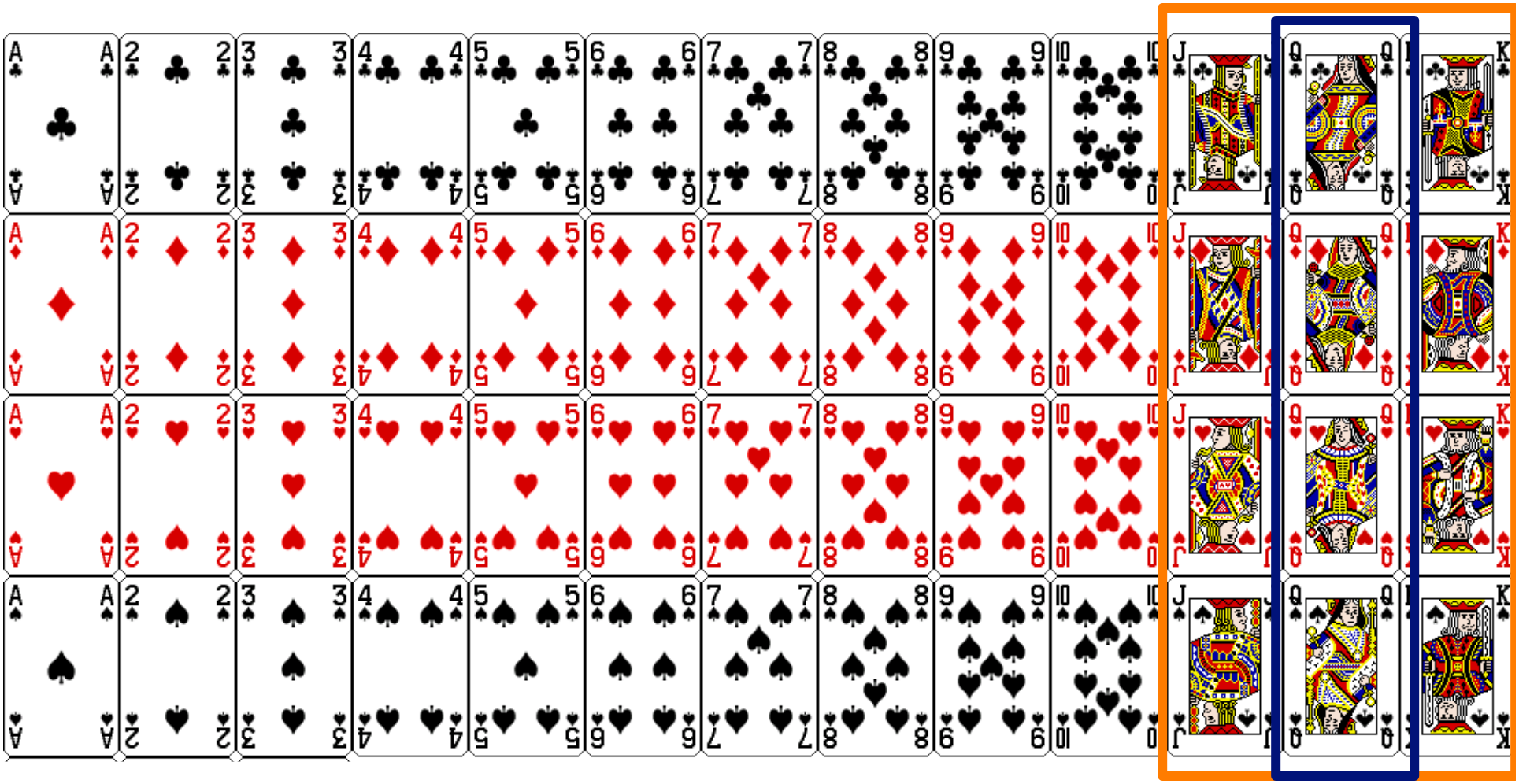


Conditional probability example



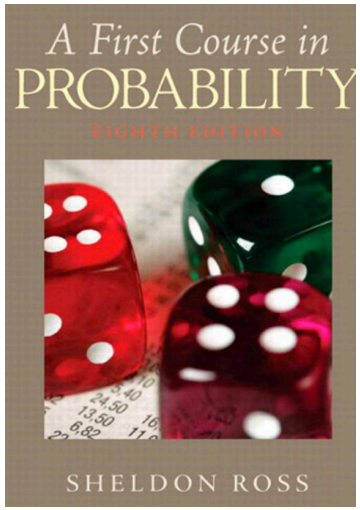
$$P(\text{Queen}|\text{RoyalCard}) = \frac{P(\text{Queen} \cap \text{RoyalCard})}{P(\text{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

Pr(Queen|Club) is independent?



$$P(\text{Queen}|\text{RoyalCard}) = \frac{P(\text{Queen} \cap \text{RoyalCard})}{P(\text{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

Conditional Probabilities Example



3.2 CONDITIONAL PROBABILITIES

Suppose that we toss 2 dice, and suppose that each of the 36 possible outcomes is equally likely to occur and hence has probability $\frac{1}{36}$. Suppose further that we observe that the first die is a 3. Then, given this information, what is the probability that the sum of the 2 dice equals 8? To calculate this probability, we reason as follows: Given that the initial die is a 3, there can be at most 6 possible outcomes of our experiment, namely, (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), and (3, 6). Since each of these outcomes originally had the same probability of occurring, the outcomes should still have equal probabilities. That is, given that the first die is a 3, the (conditional) probability of each of the outcomes (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), and (3, 6) is $\frac{1}{6}$, whereas the (conditional) probability of the other 30 points in the sample space is 0. Hence, the desired probability will be $\frac{1}{6}$.

If we let E and F denote, respectively, the event that the sum of the dice is 8 and the event that the first die is a 3, then the probability just obtained is called the *conditional probability that E occurs given that F has occurred* and is denoted by

$$P(E|F)$$

A general formula for $P(E|F)$ that is valid for all events E and F is derived in the same manner: If the event F occurs, then, in order for E to occur, it is necessary that the actual occurrence be a point both in E and in F; that is, it must be in EF. Now, since we know that F has occurred, it follows that F becomes our new, or reduced, sample space; hence, the probability that the event EF occurs will equal the probability of EF relative to the probability of F. That is, we have the following definition.

58

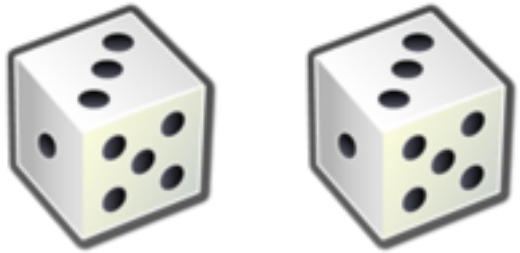
Definition

If $P(F) > 0$, then

$$P(E|F) = \frac{P(EF)}{P(F)} \tag{2.1}$$

Conditional Probabilities Puzzle

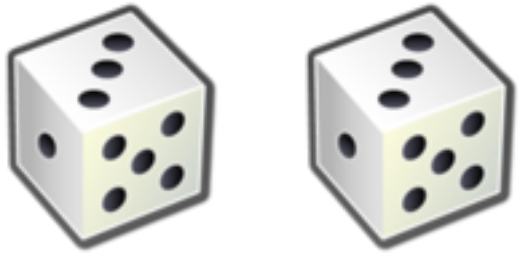
Roll a die twice



Question Roll a die twice. What is the probability that the total we get is 3? Given the information that the first number is 1, what is probability that the total we get is 3?

Conditional Probabilities Puzzle

Roll a die twice



Question Roll a die twice. What is the probability that the total we get is 3? Given the information that the first number is 1, what is probability that the total we get is 3?

Answer: 2/36 and 1/6.

2 of 6x6 = 36 equilikely events: 1+2, 2+1

Since 1st number =1, 2nd number must =2 to get a sum of 3. There are 6 possible 2nd outcomes, 1 of which is to get a 2, hence 1/6

Conditional Probabilities Example 2

Example 1: If the probability that a research project will be well planned is 0.8 and the probability that is will be well planned and well executed is 0.72, what is the probability that a well planned research project will be well executed?

Conditional Probabilities Examples

Example 1: If the probability that a research project will be well planned is 0.8 and the probability that is will be well planned and well executed is 0.72, what is the probability that a well planned research project will be well executed?

Answer: $0.72/0.8 = 0.9$. P[Exec|Plan]=P[Exec&Plan]/P[Plan]

Conditional probability as an easier path to an answer

Sometimes the conditional probability can be determined easily, so we can actually use the conditional probability to calculate probability.

The multiplication rule: $P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$.

Example: There are 3 red balls and 2 blue balls in a box. Randomly take 2 balls from the box. What is the probability that both are red?

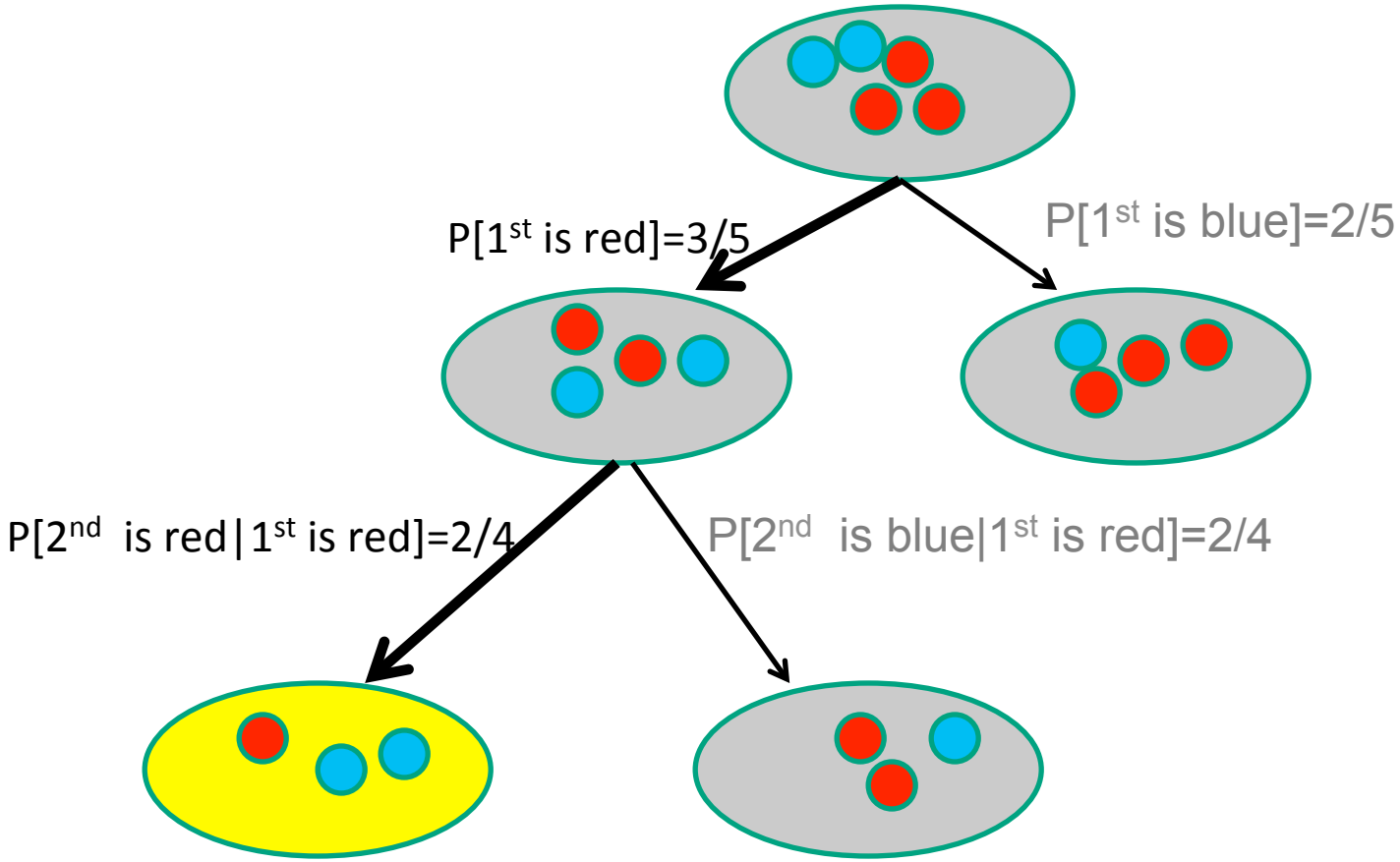
Answer: $P(R_1 \cap R_2) = P(R_1)P(R_2|R_1) = (3/5)(2/4) = 0.3$.

3 of the 5 are red

With 1 red ball gone, 2 of the remaining 4 balls are red

Alternative solution: P[2 red in 2 tries] = Comb(3,2)*Comb(2,0)/ Comb(5,2) = 0.3

Solution via Tree Diagram



P[both are red] = p[2nd is red|1st is red] x P[1st is red] = 2/4 x 3/5 = 6/20 = 0.3

Conditional Probability Example

The initial intuition for conditional probability comes from considering probabilities that are ratios. In the case of ratios, $P(\mathsf{E}|\mathsf{F})$, as defined above, is the fraction of items in F that are also in E. We show this as follows. Let n be the number of items in the sample space, n_{F} be the number of items in F, and n_{EF} be the number of items in $\mathsf{E} \cap \mathsf{F}$. Then

$$\frac{P(\mathsf{E} \cap \mathsf{F})}{P(\mathsf{F})} = \frac{n_{\mathsf{EF}}/n}{n_{\mathsf{F}}/n} = \frac{n_{\mathsf{EF}}}{n_{\mathsf{F}}},$$

which is the fraction of items in F that are also in E. As far as meaning, $P(\mathsf{E}|\mathsf{F})$ means the probability of E occurring given that we know F has occurred.

Example 1.6 *Again consider drawing the top card from a deck of cards, let* Queen *be the set of the* 4 *queens,* RoyalCard *be the set of the* 12 *royal cards, and* Spade *be the set of the* 13 *spades. Then*

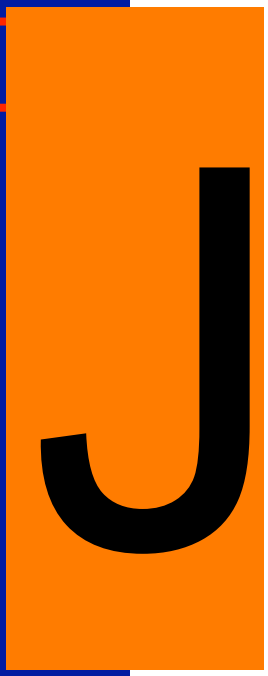
$$P(\mathsf{Queen}) = \frac{1}{13}$$

$$P(\mathsf{Queen}|\mathsf{RoyalCard}) = \frac{P(\mathsf{Queen} \cap \mathsf{RoyalCard})}{P(\mathsf{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

$$P(\mathsf{Queen}|\mathsf{Spade}) = \frac{P(\mathsf{Queen} \cap \mathsf{Spade})}{P(\mathsf{Spade})} = \frac{1/52}{1/4} = \frac{1}{13}.$$

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Product Rule is Fundamental and follows from cond[al] probability

$$P(Queen \mid C\,\text{lub}) = \frac{P(Queen, C\,\text{lub})}{P(C\,\text{lub})} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$$

Cond[al] Prob:

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

Chain Rule, Bayes Rule follow from the Product Rule

$P(A,B)$ is the belief in the joint event of A and B (joint probability)

$P(B)$ is the marginal probability of *B*

PRODUCT RULE :

$$P(A,B) = P(A \mid B)P(B)$$

Product Rule is Fundamental!

$$P(Queen \mid C\,\text{lub}) = \frac{P(Queen, C\,\text{lub})}{P(C\,\text{lub})} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$$

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

Chain Rule, Bayes Rule follow from the Product Rule

PRODUCT RULE :

$$P(A,B) = P(A \mid B)P(B)$$

Product Rule Part 2!
(Often left to your imaginations)

$$P(A \mid B) = \frac{P(A,B)}{P(B)} \; and \; P(B \mid A) = \frac{P(A,B)}{P(A)}$$

$$P(A,B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

Marginalize via Conditional Probability

Simpler calculation: get marginal from joint; decompose joint using product rule

FROM $P(A) = \sum_i P(A, B_i)$ Marginalize A from joint A,B

AND PRODUCT RULE $P(A, B) = P(A \mid B)P(B)$

$$P(A) = \sum_i P(A \mid B_i)P(B_i)$$

the belief in any event *A* is a weighted sum over the beliefs in all the distinct ways that *A* might be realized.

Chain Rule follows from the Product Rule

PRODUCT RULE: $P(A,B) = P(A \mid B)P(B)$

1. Chain Rule = Generalization of PRODUCT RULE.
2. It permits the calculation of any member of the joint distribution of a set of random variables using only conditional probabilities

$$P(E_1, E_2 \ldots E_n) = P(E_n \mid E_{n-1}, \ldots, E_2, E_1) P(E_{n-1}, \ldots, E_2, E_1)$$

let's further decompose!

$$\ldots\ldots\ldots\ldots\ldots\ldots = P(E_n \mid E_{n-1}, \ldots, E_2, E_1) \ldots P(E_2 \mid E_1) P(E_1)$$

- **For example, derive using the chain rule**

$$\mathrm{P}(A, B, C, D) = \mathrm{P}(A \mid B, C, D)\mathrm{P}(B \mid C, D)\mathrm{P}(C \mid D)\mathrm{P}(D)$$

Chain Rule Example: Use to calculate joint prob.

- **The rule is useful in the study of Bayesian networks, which describe a probability distribution in terms of conditional probabilities.**
- **Assume Urn 1 has 1 black balls and 2 white balls and Urn 2 has 1 black ball and 3 white balls. Suppose we pick an urn at random and then select a ball from that urn.**
- **Let event A be choosing the first urn: P(A) = P(~A) = 1/2. Let event B be the chance we choose a white ball. Chance of choosing a white ball, given that we've chose the first urn, is P(B|A) = 2/3. Chance of choosing a white ball, given that we've chosen the second urn is P(B|~A) = 3/4.**
- **Event A, B would be their intersection; choosing the first urn and a white ball from it. The probability can be found by the chain rule for probability:**

$$\mathrm{P}(A, B) = \mathrm{P}(B \mid A)\mathrm{P}(A) = 2/3 \times 1/2 = 1/3$$

[Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/ , p. 496.]

Bayes Rule: 1, 2, 3

1. $P(A \mid B) = \dfrac{P(A,B)}{P(B)}$ $\quad P(B \mid A) = \dfrac{P(A,B)}{P(A)}$ Cond[al] Prob

2. $P(A,B) = P(A \mid B)P(B)$ $\quad P(A,B) = P(B \mid A)P(A)$ Product Rule

3. $P(A \mid B)P(B) = P(B \mid A)P(A)$

Prior Probability

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Posterior probability

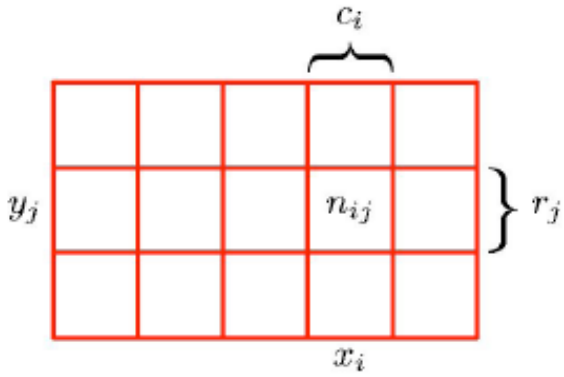
Posterior $\quad$ Likelihood $\quad$ Prior

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{\sum_{i=1}^{n} P(B \mid A = a_i)P(A = a_i)}$$

Marginalize B

Rules of Probability

- Given random variables X and Y
- Sum Rule gives Marginal Probability



$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j) = \frac{c_i}{N}$$

- Product Rule: joint probability in terms of conditional and marginal

$$p(X,Y) = \frac{n_{ij}}{N} = p(Y \mid X)p(X) = \frac{n_{ij}}{c_i} \times \frac{c_i}{N}$$

- Combining we get Bayes Rule

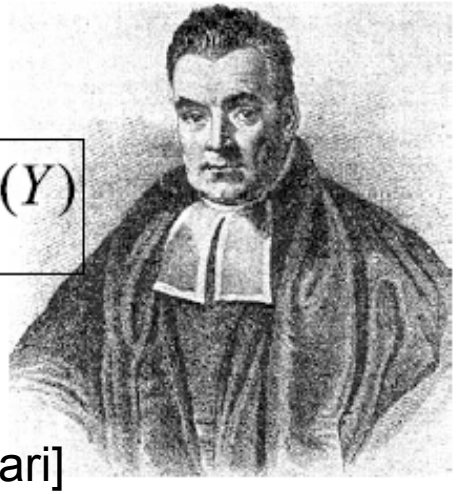
$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}$$

where

$$p(X) = \sum_{Y} p(X \mid Y)p(Y)$$

Viewed as

Posterior α likelihood x prior



[Srihari]

Maximum a Posteriori (MAP)

- **Any such maximally probable hypothesis is called a *maximum a posteriori (MAP) hypothesis***

Bayes Rule Puzzle

- Great example to test your understanding of Bayes Rule in practice
- Please test yourself

Bayes Rule Puzzle

test with two possible outcomes: ⊕ (positive) and ⊖ (negative). We have prior knowledge that over the entire population of people only .008 have this disease. Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result. The above situation can be summarized by the following probabilities:

PUZZLE

Fill in the probabilities based on the above text

Pr(Cancer) = ??
Pr(+|Cancer) =??
Pr(??????|????)=???

Bayes Rule Example

test with two possible outcomes: $\oplus$ (positive) and $\ominus$ (negative). We have prior knowledge that over the entire population of people only .008 have this disease. Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result. The above situation can be summarized by the following probabilities:

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

Patient has cancer

PUZZLE:

Observe a new patient for whom the lab tests return a positive result

Should we diagnose the patient as having cancer or not?

Bayes Rule Example

test with two possible outcomes: $\oplus$ (positive) and $\ominus$ (negative). We have prior knowledge that over the entire population of people only .008 have this disease. Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result. The above situation can be summarized by the following probabilities:

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

Patient has cancer

PUZZLE:

Observe a new patient for whom the lab tests return a positive result

Should we diagnose the patient as having cancer or not?

Use Bayes theorem to calculate
Given Pr(Cancer|+) = Pr(+|Cancer) X Pr(Cancer)/ Pr(+)
Then…

Bayes Rule Example

Patient has cancer

test with two possible outcomes: $\oplus$ (positive) and $\ominus$ (negative). We have prior knowledge that over the entire population of people only .008 have this disease. Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result. The above situation can be summarized by the following probabilities:

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

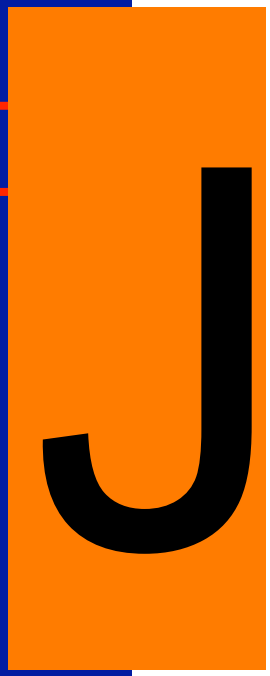
Suppose we now observe a new patient for whom the lab test returns a positive result. Should we diagnose the patient as having cancer or not? The maximum a posteriori hypothesis can be found using Equation (6.2):

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

Thus, $h_{MAP} = \neg cancer$. The exact posterior probabilities can also be determined by normalizing the above quantities so that they sum to 1 (e.g., $P(cancer|\oplus) = \frac{.0078}{.0078+.0298} = .21$). This step is warranted because Bayes theorem states that the posterior probabilities are just the above quantities divided by the probability of the data, $P(\oplus)$. Although $P(\oplus)$ was not provided directly as part of the problem statement, we can calculate it in this fashion because we know that $P(cancer|\oplus)$ and $P(\neg cancer|\oplus)$ must sum to 1 (i.e., either the patient has cancer or they do not). Notice that while the posterior probability of *cancer* is significantly higher than its prior probability, the most probable hypothesis is still that the patient does

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Bayes Theorem for Machine Learning

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

- P(h) = Prior probability of hypothesis
- P(D) = Prior probability of training data D.
- P(h|D) = Probability of h given D.
- P(D|h) = Probability of D given h.

- **Bayesian Learning via**
  - Max Likelihood
  - Bayesian

Bayesian Learning: Discrete input/output variables

- **Here we consider the relationship between supervised learning, or function approximation problems, and Bayesian reasoning.**

- **We begin by considering how to design learning algorithms based on Bayes rule.**
  - Consider a supervised learning problem in which we wish to approximate an unknown target function
    - $f : X \rightarrow Y$, or equivalently $P(Y|X)$.
  - To begin, we will assume Y is a boolean-valued random variable, and X is a vector containing n boolean attributes.
  - In other words, X =<X1,X2 ...,Xn>, where Xi is the boolean random variable denoting the ith attribute of X.

Digression

- **Digress to cover Confidence intervals**

Confidence interval for a binomial distribution

- In statistics, a binomial proportion confidence interval is a confidence interval for a proportion in a statistical population. It uses the proportion estimated in a statistical sample and allows for sampling error. There are several formulas for a binomial confidence interval, but all of them rely on the assumption of a binomial distribution.
- In general, a binomial distribution applies when an experiment is repeated a fixed number of times, each trial of the experiment has two possible outcomes (labeled arbitrarily success and failure), the probability of success is the same for each trial, and the trials are statistically independent.
- A simple example of a binomial distribution is the set of various possible outcomes, and their probabilities, for the number of heads observed when a (not necessarily fair) coin is flipped ten times.
- The observed binomial proportion is the fraction of the flips which turn out to be heads.
- Given this observed proportion, the confidence interval for the true proportion innate in that coin is a range of possible proportions which may contain the true proportion.
- A 95% confidence interval for the proportion, for instance, will contain the true proportion 95% of the times that the procedure for constructing the confidence interval is employed.
- Note that this does not mean that a calculated 95% confidence interval will contain the true proportion with 95% probability. Instead, one should interpret it as follows: the process of drawing a random sample and calculating an accompanying 95% confidence interval will generate a confidence interval that contains the true proportion in 95% of all cases.
- There are several ways to compute a confidence interval for a binomial proportion. The normal approximation interval is the simplest formula, and the one introduced in most basic Statistics classes and textbooks. This formula, however, is based on an approximation that does not always work well.

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}\hat{p}\left(1-\hat{p}\right)}$$

https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval

Confidence interval for a binomial distribution

https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval

Confidence interval for a binomial distribution

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}\hat{p}\left(1-\hat{p}\right)}$$

Election Poll of 1000 people (N=1000)
Assume 45% (P=0.45, Q=1-P=55%) favor candidate A (Donald) versus 49% for Candidate B (Hillary)
Standard error about the mean = SQRT(PQ/N) = SQRT(0.45*0.55/1000)=0.015=1.5%
So the 95% confidence interval surrounding Donald's support of 45% is 45%+/- 2*1.5 = [42,…48]

In machine learning:
Pr(Y=Business) = 100/1000 = 0.1
SQRT(0.1*0.9/1000) =0.01 = 1%
yielding Pr(Y=Business) Confidence interval = 0.1+/- 0.01*2
[0.08,0.12] Note we need 1,000 examples to get this tight CI;

How much data do we need to estimate Pr(Y|X) with confidence?

Learning Classifiers based on Bayes Rule

Applying Bayes rule, we see that $P(Y = y_i|X)$ can be represented as

$$P(Y = y_i|X = x_k) = \frac{P(X = x_k|Y = y_i)P(Y = y_i)}{\sum_j P(X = x_k|Y = y_j)P(Y = y_j)}$$

LHS **RHS**

where y_m denotes the mth possible value for Y, x_k denotes the kth possible vector value for X, and where the summation in the denominator is over all legal values of the random variable Y.

One way to learn $P(Y|X)$ is to use the training data to estimate $P(X|Y)$ and $P(Y)$. We can then use these estimates, together with Bayes rule above, to determine $P(Y|X = x_k)$ for any new instance x_k.

Confidence interval for a binomial distribution

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{n}\hat{p}\,(1-\hat{p})}$$

Estimating P(Y), P(X|Y) from data

- **If we are going to train a Bayes classifier by estimating P(X|Y) and P(Y), then it is reasonable to ask how much training data will be required to obtain reliable estimates of these distributions.**
- **Let us assume training examples are generated by drawing instances at random from an unknown underlying distribution P(X), then allowing a teacher to label this example with its Y value.**
- **A 1,000 independently drawn training examples will usually suffice to obtain a maximum likelihood estimate of P(Y) that is within a few percent of its correct value when Y is a boolean variable.**

Note for a simple proposition the Standard error:
SE(P(Y))=SQRT(0.1*0.9/1000)=0.01 for business labeled docs; assume 100 out 1000

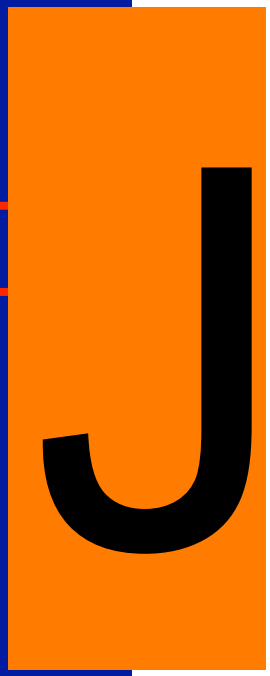
- **However, accurately estimating P(X|Y) typically requires many more examples**

Assume n binary variables: 2^n propositions X 1000 training examples
See example in a moment

Impractical: so what to do?

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Supervised Learning Via Bayes Rule

- **Consider a supervised learning problem in which we wish to approximate an unknown target function *f : X →Y, or equivalently P(Y_j | X).***
- ***For pedagogical reasons assume discrete binary variables for both and input (X)and output (Y)***
  - *To begin, we will* assume *Y is a boolean-valued random variable, and X is a vector containing n* boolean attributes. In other words, *X = hX1;X2 : : : ;Xni, where Xi is the boolean* random variable denoting the *ith attribute of X.*
- **Applying Bayes rule, we see that *P(Y = y_i | X) can be represented as***

$$P(Y = y_i | X = x_k) = \frac{P(X = x_k | Y = y_i) P(Y = y_i)}{\sum_j P(X = x_k | Y = y_j) P(Y = y_j)}$$

- **where *y_m represents the m^{th} possible value for Y, and where the summation in the* denominator is over all legal values of the random variable *Y***

Learning a (FULL) Bayesian Model

- **One way to learn *P(Y | X)***
  1. *Estimate the joint probability distribution P(X |Y) and P(Y) from the training data.*
  2. *Use these estimates, together with Bayes rule above, to determine P(Y | X = x_k) for any new instance x_k.*
- ***A maximum likelihood estimate of P(Y) can be accomplished with just a few hundred examples***
  - *#ExamplesWithLabel/#TotalNumberOfExamples. E.g., 45 examples are in class 1 and 55 are in class2 then Pr(Y=Class1) = 45/100=0.45.*
- ***However, estimating the joint probability distribution P(X |Y) requires an exponential amount training examples (even with this assumption of binary input and output variable)***
  - *Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters)*

Estimating the Joint Prob. Directly?

- **2^n * 2 possible states of the world (parameter estimates)**
  - Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters) that we need to estimate from data
  - $\theta_{iJ} = P(X = x_i | Y = y_J)$; 2^n possible states of the input world 2 possible output states
- **Can reduce 2^n -1 * 2 parameters by exploiting the sum-to-1**
  - Class conditional multinomial needs to sum to 1 so we exploit this and can infer one of the class conditional probs from the rest
  - Each state is a complex combination of feature values
- **Require 200k examples (or more) for reliable estimates of 10 binary variable problem**
  - So for 10 input variables and one output variable we have to estimate 2046 states. To estimate probabilities requiring at least 204,600 examples for reliable estimates.
- **So not very realistic, even in these WWW times**

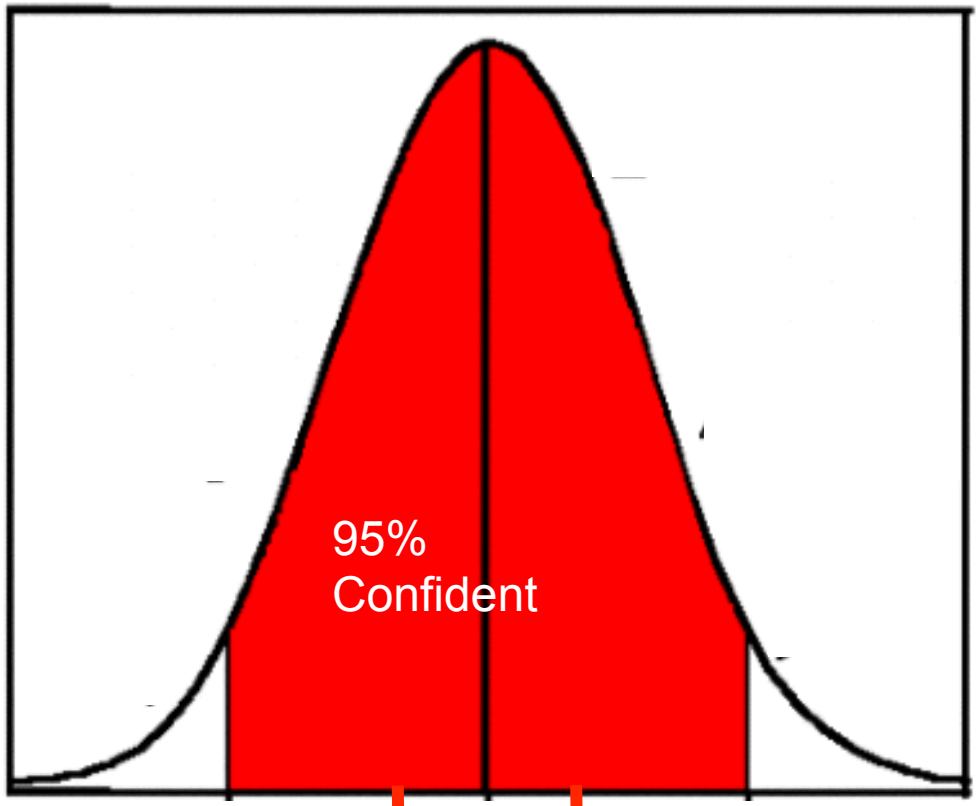
Learning a Bayesian Model

- *2^n * 2 possible states of the world*
  - *Assume n input attributes X_i take 2 discrete values and Y has 2 possible class values; 2^n * 2 possible states of the world (parameters) that* we need to estimate from data
    - $\theta_{iJ} = P(X = x_i \mid Y = y_J)$; *$2^n$ possible states of the input world 2 possible output states*

Index $X=X^i$	X1 $X=x_1^i$)	X2 $X=x_2^i$)	Y $Y=y_j$	$\theta_{ij}=P(X=x^i \mid y=y_j)$
1	1	1	1	θ_{11}
2	0	1	1	θ_{21}
3	1	0	1	θ_{31}
4	0	0	1	θ_{41}
5	1	1	0	θ_{50}
6	0	1	0	θ_{60}
7	1	0	0	θ_{70}
8	0	0	0	θ_{80}

Sum to 1

Estimating Reliable Probabilites



Estimate using Binomial MLE Estimates
I.e., #Clicks/#Impression

$40/1,000 @CPC of $1.60
$400/10,000

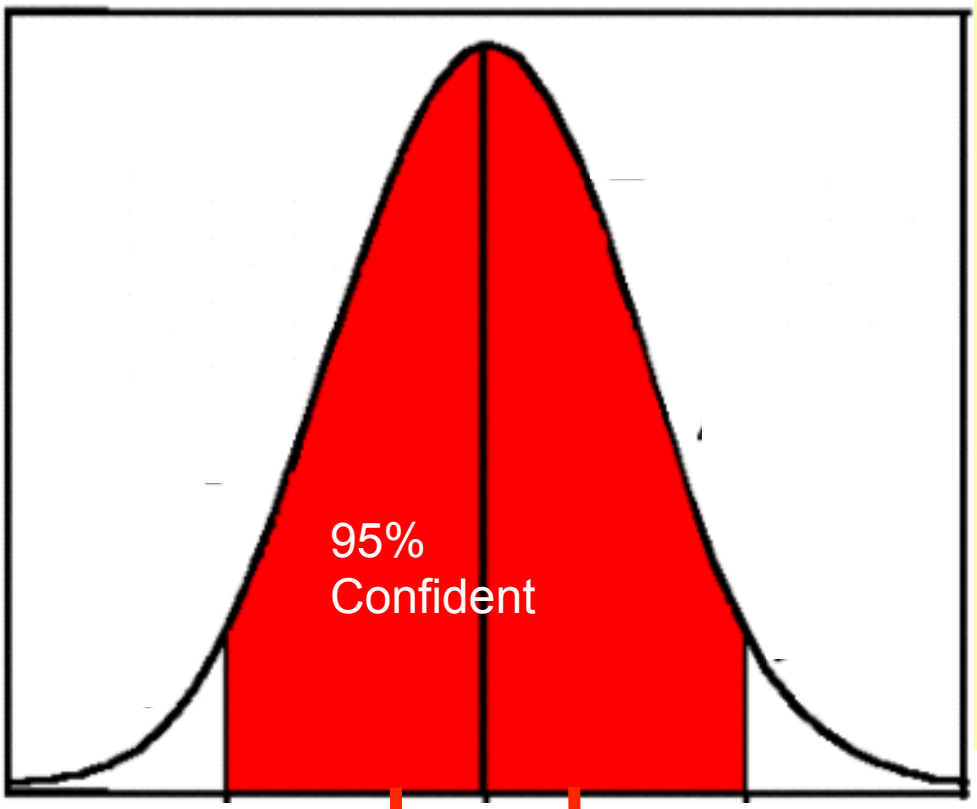
Standard error of the mean of one sample is the estimate of the standard deviation that would be obtained from the means of a large number of samples drawn from that population.

StdErr= sqrt(.026*(1-0.026)/1000) * 1.96=±1%

1.6% 2.6% 3.6% CTR (after 1,000 impressions)

2.3% 2.9% (after 10,000 impressions)

Estimating CTR (and later AR)



For a network of
~10^9 target pages,
~10^6 ads
~10^7 users
……
- Cannot afford this evaluation/auditioning
- Borrow strength, marginalize
- CoD (curse of dim^{ality})

Confidence Intervals in a Nutshell

- **The assumptions required for CI for a population proportion to be valid:**
  - the sample size *n* is large enough (check: $n\hat{p} \geq 10$ and $n(1-\hat{p}) \geq 10$)
  - the data are a *random sample* from that population
- **General Confidence Interval for the Population Proportion *p*:**

$$\hat{p} \pm 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

- **Approximate 95% Confidence Interval for the Population Proportion *p*:**

$$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

```
> 2*sqrt(.4*.6/1000)  #president's satisfaction rating
[1] 0.03098387
```

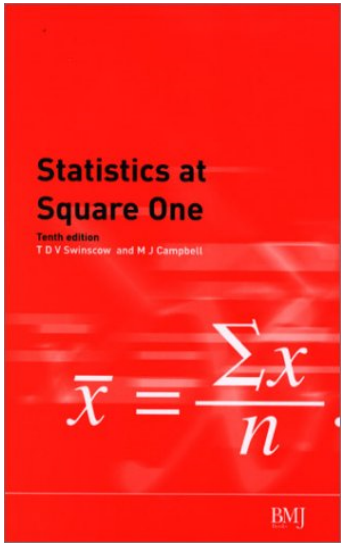
  - Note: standard error of is which is largest when $\hat{p} = \frac{1}{2}$
- **Conservative Confidence Interval for the Population Proportion *p*:**

$$\hat{p} \pm \frac{z^*}{2\sqrt{n}}$$

1/sqrt(n), e.g., 1/sqrt(1000)= ± 3%

Sample Size Needed

- **Sample Size Needed for Desired Confidence Level and Error Margin where *m* is the desired margin of error.**



$$n = \left(\frac{z^*}{2m} \right)^2$$

Measure CTRs confidently p± 0.001
> (1.96/(2*.001))^2
[1] 960,400 sample size

[http://www.bmj.com/statsbk/ Statistics at Square One (Ninth Edition), T D V Swinscow]

Sample Size and Stats

- More later in the context of AB testing…..

Cant Estimate Joint Probability

- **Look for ways to combat the intractable data needs for learning a Bayesian Classifier**
  - Leverage the chain rule and other assumptions (Markov, Naïve Bayes); this leads to Bayesian Networks
  - Make a conditional independence assumption; this leads to a Naïve Bayes classifier
    - Reduces the number of parameters from 2^n -1 * 2 parameters to 2n

Bayesian Networks

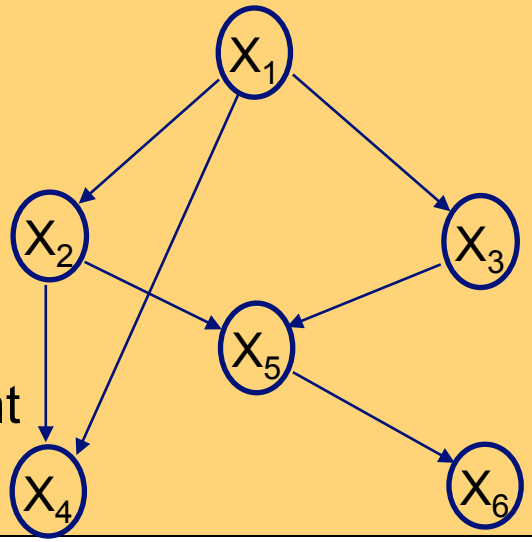
GOAL: 2^N Possibilities -> 2N

P: Joint Probability Distribution

#	X1	X2	X3	X4	X5	X6	Pr(X1,X2, X3, X4, X5, X6)
1	1						
2	1						
..	1						
4..	1						
5	1						
6	1						
7	1						
64							

Factorize the joint

G: Directed Acyclic Graph



$p(x_1, x_2, x_3, x_4, x_5, x_6)$ **1. Partial Order**

2. By Chain Rule

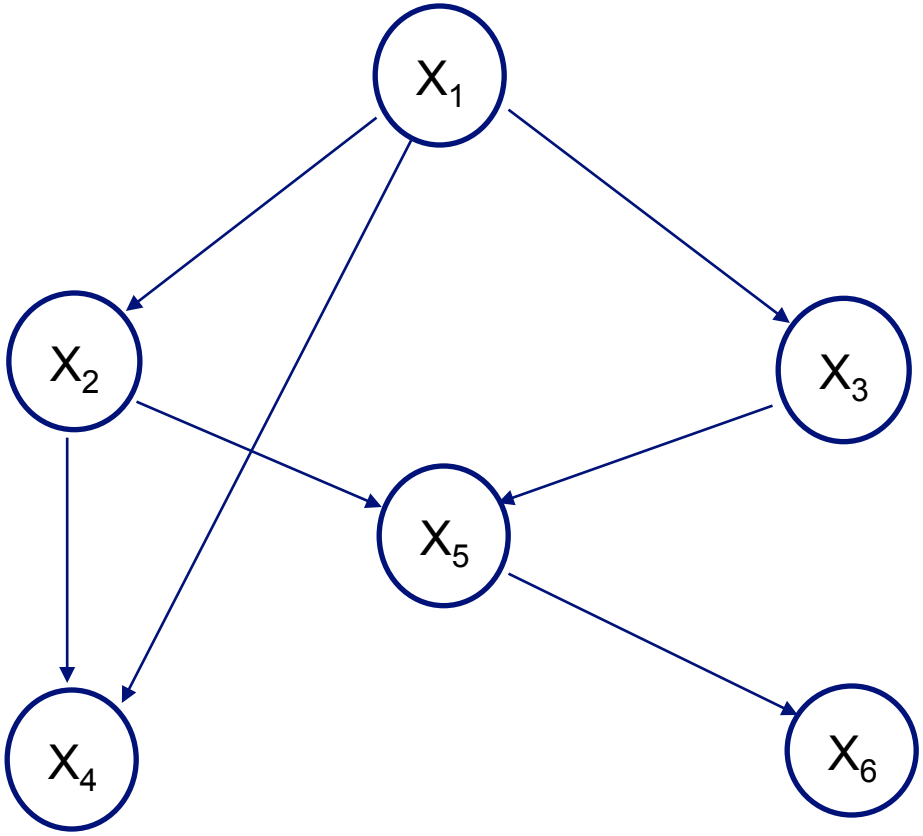
$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2)\, p(x_4 \mid x_1, x_2, x_3)\, p(x_5 \mid x_4,x_3,x_2,x_1)\, p(x_6 \mid x_5,x_4,x_3,x_2,x_1)$

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2, x_1)\, p(x_5 \mid x_3, x_2)\, p(x_6 \mid x_5)$ **3. Markov Property**

$= p(x_1)\, p(x_2)\, p(x_3)\, p(x_4)\, p(x_5)\, p(x_6)$ **4: Independence (see next section)**

$P(y|x_6, x_5,x_4,x_3,x_2,x_1) = p(x_1|y)\, p(x_2|y)\, p(x_3|y)\, p(x_4|y)\, p(x_5|y)\, p(x_6|y)$ **5: Naïve Bayes via Cond. Independence**

1, 2, 3 Example of Factorization



2. By Chain Rule

$p(x_1, x_2, x_3, x_4, x_5, x_6)$ 1. Partial Order

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2)\, p(x_4 \mid x_1, x_2, x_3)\, p(x_5 \mid x_4, x_3, x_2, x_1)\, p(x_6 \mid x_5, x_4, x_3, x_2, x_1)$

$= p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2, x_1)\, p(x_5 \mid x_3, x_2)\, p(x_6 \mid x_5)$ 3. Markov Property

Factorization

- Given a DAG G, topologically sort the variables:

  $X_1, \ldots, X_n$ (s.t. if i<j, then X_j is not an ancestor of X_i)

- For any joint distribution P that is Markov to G, factorize it as follows:

  $P(X_1, \ldots, X_n)$

  $= P(X_1)\, P(X_2|X_1) \ldots P(X_n|X_1, \ldots, X_{n-1})$ By Chain Rule

  $= \prod_i P(X_i \mid Pa(X_i))$ Exploit Local Markov Property

- Markov property: every variable is independent of its non-descendants given its parents.
- Markov Condition requires that every conditional independence in the graph is in the joint probability distribution.

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case)
- **Naïve Bayes (next live session)**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
  - Continuos input variables (Cover later)
- **Case Study: Spam detector in Naïve Bayes**

- **Independence**

Independence

- **In probability theory, two events are independent, statistically independent, or stochastically independent if the occurrence of one does not affect the probability of the other.**
  - Similarly, two random variables are independent if the realization of one does not affect the probability distribution of the other.
  - The concept of independence extends to dealing with collections of more than two events or random variables, in which case the events are pairwise independent if each pair are independent of each other, and the events are mutually independent if each event is independent of each other combination of events.

https://en.wikipedia.org/wiki/Independence_(probability_theory)

Two events [edit]

Two events *A* and *B* are **independent** (often written as $A \perp B$ or $A \perp\!\!\!\perp B$) if and only if their joint probability equals the product of their probabilities:

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B).$$

Why this defines independence is made clear by rewriting with conditional probabilities:

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B) \Leftrightarrow \mathrm{P}(A) = \frac{\mathrm{P}(A)\mathrm{P}(B)}{\mathrm{P}(B)} = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)} = \mathrm{P}(A \mid B)$$

and similarly

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B) \Leftrightarrow \mathrm{P}(B) = \mathrm{P}(B \mid A).$$

Thus, the occurrence of *B* does not affect the probability of *A*, and vice versa. Although the derived expressions may seem more intuitive, they are not the preferred definition, as the conditional probabilities may be undefined if *P*(*A*) or *P*(*B*) are 0. Furthermore, the preferred definition makes clear by symmetry that when *A* is independent of *B*, *B* is also independent of *A*.

More than two events [edit]

A finite set of events $\{A_i\}$ is **pairwise independent** if and only if every pair of events is independent[2]—that is, if and only if for all distinct pairs of indices *m*, *k*,

$$\mathrm{P}(A_m \cap A_k) = \mathrm{P}(A_m)\mathrm{P}(A_k).$$

A finite set of events is **mutually independent** if and only if every event is independent of any intersection of the other events[2]—that is, if and only if for every *n*-element subset $\{A_i\}$,

$$\mathrm{P}\left(\bigcap_{i=1}^{n} A_i\right) = \prod_{i=1}^{n} \mathrm{P}(A_i).$$

This is called the *multiplication rule* for independent events. Note that it is not a single condition involving only the product of all the probabilities of all single events (see below for a counterexample); it must hold true for all subset of events.

For more than two events, a mutually independent set of events is (by definition) pairwise independent; but the converse is not necessarily true (see below for a counterexample).

Two events are independent if…

Definition 1.3 *Two events* E *and* F *are independent if one of the following hold:*

1. $P(\mathsf{E}|\mathsf{F}) = P(\mathsf{E})$ *and* $P(\mathsf{E}) \neq 0,\ P(\mathsf{F}) \neq 0.$
2. $P(\mathsf{E}) = 0$ *or* $P(\mathsf{F}) = 0.$

Notice that the definition states that the two events are independent even though it is based on the conditional probability of E given F. The reason is that independence is symmetric. That is, if $P(\mathsf{E}) \neq 0$ and $P(\mathsf{F}) \neq 0$, then $P(\mathsf{E}|\mathsf{F}) = P(\mathsf{E})$ if and only if $P(\mathsf{F}|\mathsf{E}) = P(\mathsf{F})$. It is straightforward to prove that E and F are independent if and only if $P(\mathsf{E} \cap \mathsf{F}) = P(\mathsf{E})P(\mathsf{F})$.

Independence is symmetric

Independence Example

Example 1.6 *Again consider drawing the top card from a deck of cards, let* Queen *be the set of the* 4 *queens,* RoyalCard *be the set of the* 12 *royal cards, and* Spade *be the set of the* 13 *spades. Then*

$$P(\text{Queen}) = \frac{1}{13}$$

$$P(\text{Queen}|\text{RoyalCard}) = \frac{P(\text{Queen} \cap \text{RoyalCard})}{P(\text{RoyalCard})} = \frac{1/13}{3/13} = \frac{1}{3}$$

$$P(\text{Queen}|\text{Spade}) = \frac{P(\text{Queen} \cap \text{Spade})}{P(\text{Spade})} = \frac{1/52}{1/4} = \frac{1}{13}.$$

Notice in the previous example that $P(\text{Queen}|\text{Spade}) = P(\text{Queen})$. This means that finding out the card is a spade does not make it more or less probable that it is a queen. That is, the knowledge of whether it is a spade is irrelevant to whether it is a queen. We say that the two events are independent in this case, which is formalized in the following definition.

[Learning Bayesian Networks, Prentice-Hall, Richard E. Neapolitan]

Independence more generally on sets: Set A and B are independent

Definition 1.6 *Suppose we have a probability space (Ω, P), and two sets* A *and* B *containing random variables defined on* Ω. *Then the sets* A *and* B *are said to be* ***independent*** *if, for all values of the variables in the sets* a *and* b, *the events* $\mathsf{A} = \mathsf{a}$ *and* $\mathsf{B} = \mathsf{b}$ *are independent. That is, either* $P(\mathsf{a}) = 0$ *or* $P(\mathsf{b}) = 0$ *or*

$$P(\mathsf{a}|\mathsf{b}) = P(\mathsf{a}).$$

When this is the case, we write

$$I_P(\mathsf{A}, \mathsf{B}),$$

where I_P *stands for independent in* P.

Independence Example for Random Variables R, T, S

Example 1.18 *Let Ω be the set of all cards in an ordinary deck, and let P assign $1/52$ to each card. Define random variables as follows:*

Variable	Value	Outcomes Mapped to this Value
R	$r1$	*All royal cards*
	$r2$	*All nonroyal cards*
T	$t1$	*All tens and jacks*
	$t2$	*All cards that are neither tens nor jacks*
S	$s1$	*All spades*
	$s2$	*All nonspades*

Then we maintain the sets $\{R, T\}$ and $\{S\}$ are independent. That is,

$$I_P(\{R, T\}, \{S\}).$$

To show this, we need show for all values of r, t, and s that

$$P(r, t|s) = P(r, t).$$

(Note that it we do not show brackets to denote sets in our probabilistic expression because in such an expression a set represents the members of the set. See the discussion following Example 1.14.) The following table shows this is the case:

Independence Example for Random Variables R, T, S

s	r	t	$P(r,t\|s)$	$P(r,t)$
$s1$	$r1$	$t1$	$1/13$	$4/52 = 1/13$
$s1$	$r1$	$t2$	$2/13$	$8/52 = 2/13$
$s1$	$r2$	$t1$	$1/13$	$4/52 = 1/13$
$s1$	$r2$	$t2$	$9/13$	$36/52 = 9/13$
$s2$	$r1$	$t1$	$3/39 = 1/13$	$4/52 = 1/13$
$s2$	$r1$	$t2$	$6/39 = 2/13$	$8/52 = 2/13$
$s2$	$r2$	$t1$	$3/39 = 1/13$	$4/52 = 1/13$
$s2$	$r2$	$t2$	$27/39 = 9/13$	$36/52 = 9/13$

Definition 1.7 *Suppose we have a probability space* (Ω, P), *and three sets* A, B, *and* C *containing random variable defined on* Ω. *Then the sets* A *and* B *are said to be conditionally independent given the set* C *if, for all values of the variables in the sets* a, b, and c, *whenever* $P(\mathsf{c}) \neq 0$, *the events* $\mathsf{A} = \mathsf{a}$ *and* $\mathsf{B} = \mathsf{b}$ *are conditionally independent given the event* $\mathsf{C} = \mathsf{c}$. *That is, either* $P(\mathsf{a}|\mathsf{c}) = 0$ *or* $P(\mathsf{b}|\mathsf{c}) = 0$ *or*

$$P(\mathsf{a}|\mathsf{b},\mathsf{c}) = P(\mathsf{a}|\mathsf{c}).$$

When this is the case, we write

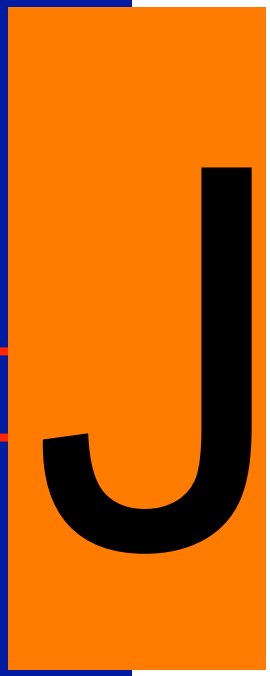
$$I_P(\mathsf{A},\mathsf{B}|\mathsf{C}).$$

Independence: Pairwise and Conditional

- **Pairwise Independence**
  - P(A,B|C)=P(A|C)P(B|C)
    - Since P(A,B|C) = P(A|B,C)P(B|C) [Chain rule]
    - and P(A|B, C)= P(A|C)
- **Conditional Independence (see next section)**
  - P(A|B)=P(A)
  - P(A|B, C)= P(A|C)
  - P(*A, B*)=P(*A*)P(*B*)

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Conditional Independence

Definition: Given random variables X, Y and Z, we say X is **conditionally independent** of Y given Z, if and only if the probability distribution governing X is independent of the value of Y given Z; that is

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

As an example, consider three boolean random variables to describe the current weather: *Rain*, *Thunder* and *Lightning*. We might reasonably assert that *Thunder* is independent of *Rain* given *Lightning*. Because we know *Lightning* causes *Thunder*, once we know whether or not there is *Lightning*, no additional information about *Thunder* is provided by the value of *Rain*. Of course there is a clear dependence of *Thunder* on *Rain* in general, but there is no *conditional* dependence once we know the value of *Lightning*.

Pr(Rain | Lightning, Thunder) == Pr(Rain | Lightning)

P(R|L, T)= P(R|L)

Conditionally Independent

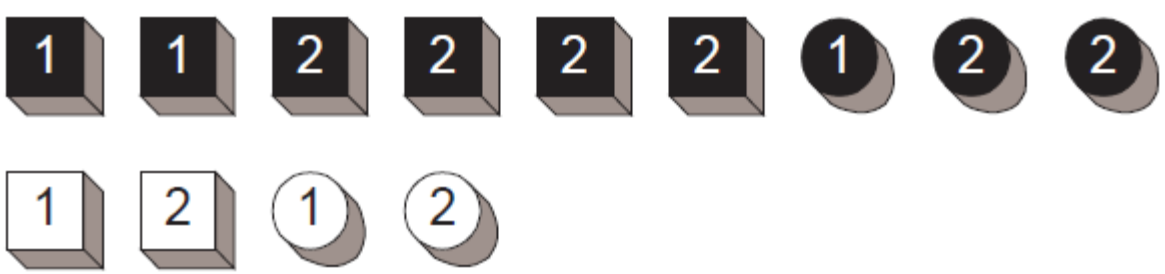


Figure 1.2: Containing a '1' and being a square are not independent, but they are conditionally independent given the object is black and given it is white.

Example 1.19 *Let Ω be the set of all objects in Figure 1.2, and let P assign $1/13$ to each object. Define random variables S (for shape), V (for value), and C (for color) as follows:*

Variable	Value	Outcomes Mapped to this Value
V	$v1$	*All objects containing a '1'*
	$v2$	*All objects containing a '2'*
S	$s1$	*All square objects*
	$s2$	*All round objects*
C	$c1$	*All black objects*
	$c2$	*All white objects*

Then we maintain that $\{V\}$ and $\{S\}$ are conditionally independent given $\{C\}$. That is,

$$I_P(\{V\}, \{S\}|\{C\}).$$

To show this, we need show for all values of v, s, and c that

$$P(v|s, c) = P(v|c).$$

The results in Example 1.8 show $P(v1|s1, c1) = P(v1|c1)$ and $P(v1|s1, c2) = P(v1|c2)$. The table that follows shows the equality holds for the other values of the variables too:

c	s	v	$P(v\|s,c)$	$P(v\|c)$
$c1$	$s1$	$v1$	$2/6 = 1/3$	$3/9 = 1/3$
$c1$	$s1$	$v2$	$4/6 = 2/3$	$6/9 = 2/3$
$c1$	$s2$	$v1$	$1/3$	$3/9 = 1/3$
$c1$	$s2$	$v2$	$2/3$	$6/9 = 2/3$
$c2$	$s1$	$v1$	$1/2$	$2/4 = 1/2$
$c2$	$s1$	$v2$	$1/2$	$2/4 = 1/2$
$c2$	$s2$	$v1$	$1/2$	$2/4 = 1/2$
$c2$	$s2$	$v2$	$1/2$	$2/4 = 1/2$

[Learning Bayesian Networks, Prentice-Hall, Richard E. Neapolitan]

Conditional Independence (Wiki)

Conditional independence [edit]

Main article: Conditional independence

Intuitively, two random variables *X* and *Y* are conditionally independent given *Z* if, once *Z* is known, the value of *Y* does not add any additional information about *X*. For instance, two measurements *X* and *Y* of the same underlying quantity *Z* are not independent, but they are **conditionally independent given *Z*** (unless the errors in the two measurements are somehow connected).

The formal definition of conditional independence is based on the idea of conditional distributions. If *X*, *Y*, and *Z* are discrete random variables, then we define *X* and *Y* to be *conditionally independent given Z* if

$$\mathrm{P}(X \leq x, Y \leq y \mid Z = z) = \mathrm{P}(X \leq x \mid Z = z) \cdot \mathrm{P}(Y \leq y \mid Z = z)$$

for all *x*, *y* and *z* such that $\mathrm{P}(Z = z) > 0$. On the other hand, if the random variables are continuous and have a joint probability density function *p*, then *X* and *Y* are conditionally independent given *Z* if

$$p_{XY|Z}(x, y|z) = p_{X|Z}(x|z) \cdot p_{Y|Z}(y|z)$$

for all real numbers *x*, *y* and *z* such that $p_Z(z) > 0$.

If *X* and *Y* are conditionally independent given *Z*, then

$$\mathrm{P}(X = x|Y = y, Z = z) = \mathrm{P}(X = x|Z = z)$$

for any *x*, *y* and *z* with $\mathrm{P}(Z = z) > 0$. That is, the conditional distribution for *X* given *Y* and *Z* is the same as that given *Z* alone. A similar equation holds for the conditional probability density functions in the continuous case.

Independence can be seen as a special kind of conditional independence, since probability can be seen as a kind of conditional probability given no events.

https://en.wikipedia.org/wiki/Conditional_independence

Conditional independence

- **V is** conditionally independent **of set *Vi* given set *Vj***
  - if p(V | ***Vi, Vj***) = p(V | ***Vj***)
  - notation: I(V, ***Vi*** | ***Vj***) or V ⊥ ***Vi*** | ***Vj***
- **Intuition**
  - if I(V, ***Vi*** | ***Vj***) then knowing ***Vi & Vj*** tells nothing more about V than knowing ***Vj*** alone
  - if we know ***Vj*** we can ignore ***Vi***

$$P(Queen \mid C\,\text{lub}) = \frac{P(Queen, C\,\text{lub})}{P(C\,\text{lub})} = \frac{\frac{1}{52}}{\frac{13}{52}} = \frac{1}{13}$$

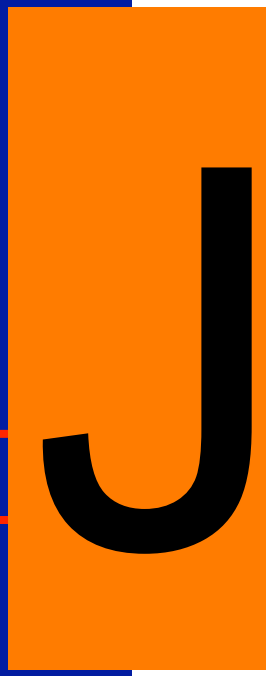
$$P(Queen) = \frac{4}{52} = \frac{1}{13}$$

Pairwise Independence

- **Single Vi conditionally independent of single Vj given *V***
  - that is, I(Vi, Vj | ***V***)
- **From definitions we have that**
  - p(Vi | Vj,**V**) = p(Vi | ***V***) and
  - p(Vi | Vj,**V**) p(Vj | ***V***) = p(Vi,Vj | ***V***)
    (Product Rule: $P(A,B) = P(A \mid B)P(B)$)
- **Thus**
  - p(Vi,Vj | ***V***) = p(Vi | ***V***) p(Vj | ***V***)
  - Vi and Vj is **pairwise independent**

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Key Slide

Derive NB Algorithm

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k) P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j) P(X_1, X_2, .., X_N | Y=y_j)}$$

$$= \frac{P(Y=y_k) \Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j) \Pi_i P(X_i | Y=y_j)}$$

The Naive Bayes algorithm is a classification algorithm based on Bayes rule, that assumes the attributes $X_1 \ldots X_n$ are all conditionally independent of one another, given Y. The value of this assumption is that it dramatically simplifies the representation of $P(X|Y)$, and the problem of estimating it from the training data. Consider, for example, the case where $X = \langle X_1, X_2 \rangle$. In this case

Independence
P(X|Y)=P(X)
P(*X*, *Y*)=P(*X*)P(*Y*)
P(X_1 |X_2, Y)= P(X_1 |Y)

X is vector of <X1, X2>

$$P(X|Y) = P(X_1, X_2|Y)$$

Use Product Rule $P(A,B) = P(A \mid B)P(B$

$$= P(X_1|X_2, Y) P(X_2|Y)$$

$$= P(X_1|Y) P(X_2|Y)$$

Naïve Bayes

Where the second line follows from a general property of probabilities (product Rule), and the third line follows directly from our above definition of conditional independence.

Naïve Bayes Classifier for Text

100 business docs = Pr(Business) =0.1+/- CI 2^N

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j)P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1

Y_2

$$= \frac{P(Y=y_k)\Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i | Y=y_j)}$$

Pr(X="corporation"|Class=Business) = 100/10000 = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k)\Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

Conditional independence dramatically reduces model complexity: to 2n parameters from 2^n

More generally, when X contains n attributes which are conditionally independent of one another given Y, we have

$$P(X_1 \ldots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \quad (1)$$

Notice that when Y and the X_i are boolean variables, we need only $2n$ parameters to define $P(X_i = x_{ik}|Y = y_j)$ for the necessary i, j, k. This is a dramatic reduction compared to the $2(2^n - 1)$ parameters needed to characterize $P(X|Y)$ if we make no conditional independence assumption.

- **High Bias**
- **Assume no interactions between the variables**

Naïve Bayes

- A generative, parametric model
- Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.
- The standard naive Bayes classifier (at least the R implementation) assumes independence of the predictor variables, and Gaussian distribution (given the target class) of metric predictors.
- For attributes with missing values, the corresponding table entries are omitted for prediction.
- Sci-Kit learn behaves similarly

Naïve Bayes and Conditional Independence

- **Make a conditional independence assumption; this leads to a Naïve Bayes classifier**
  - Reduces the number of parameters from 2n -1 * 2 parameters to 2n
- ***Definition: Given random variables X;Y and Z, we say X is conditionally* independent of *Y given Z, if and only if the probability* distribution governing *X is independent of the value of Y given Z;***
  - $(\forall i; j; k)\ P(X = x_i \mid Y = y_j, Z = z_k) = P(X = x_i \mid Z = z_k)$

Class Inference: Classify a new data point

- ..

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k) P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j) P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1

Y_2

$$= \frac{P(Y=y_k) \Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j) \Pi_i P(X_i | Y=y_j)}$$

Pr("corporation"|Class=Business) = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

Probability Basics

- Prior, conditional and joint probability
  - Prior probability: $P(X)$
  - Conditional probability: $P(X_1 \mid X_2), P(X_2 \mid X_1)$
  - Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
  - Relationship: $P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) = P(X_1 \mid X_2)P(X_2)$
  - Independence: $P(X_2 \mid X_1) = P(X_2), P(X_1 \mid X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$
- Bayesian Rule

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \qquad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Probabilistic Classification

- Establishing a probabilistic model for classification
  - Discriminative model

$$P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

  - Generative model

$$P(\mathbf{X} \mid C) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

- MAP classification rule
  - MAP: Maximum A Posterior
  - Assign x to c^* if $P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*, \; c = c_1, \cdots, c_L$
- Generative classification with the MAP rule
  - Apply Bayesian rule to convert $P(C \mid \mathbf{X}) = \dfrac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \propto P(\mathbf{X} \mid C)P(C)$

Naive Bayes for Discrete-Valued Inputs

When the n input attributes X_i each take on J possible discrete values, and Y is a discrete variable taking on K possible values, then our learning task is to estimate two sets of parameters. The first is

$$\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k) \tag{4}$$

for each input attribute X_i, each of its possible values x_{ij}, and each of the possible values y_k of Y. Note there will be nJK such parameters, and note also that only $n(J-1)K$ of these are independent, given that they must satisfy $1 = \sum_j \theta_{ijk}$ for each pair of i, k values.

In addition, we must estimate parameters that define the prior probability over Y:

$$\pi_k \equiv P(Y = y_k) \tag{5}$$

Note there are K of these parameters, $(K-1)$ of which are independent.

NB Smoothing

We can estimate these parameters using either maximum likelihood estimates (based on calculating the relative frequencies of the different events in the data), or using Bayesian MAP estimates (augmenting this observed data with prior distributions over the values of these parameters).

Maximum likelihood estimates for θ_{ijk} given a set of training examples D are given by

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}} \quad (6)$$

where the $\#D\{x\}$ operator returns the number of elements in the set D that satisfy property x.

One danger of this maximum likelihood estimate is that it can sometimes result in θ estimates of zero, if the data does not happen to contain any training examples satisfying the condition in the numerator. To avoid this, it is common to use a "smoothed" estimate which effectively adds in a number of additional "hallucinated" examples, and which assumes these hallucinated examples are spread evenly over the possible values of X_i. This smoothed estimate is given by

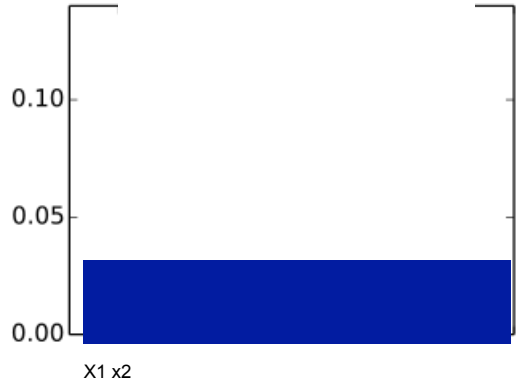
$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\} + l}{\#D\{Y = y_k\} + lJ} \quad (7)$$

where J is the number of distinct values X_i can take on, and l determines the strength of this smoothing (i.e., the number of hallucinated examples is lJ). This expression corresponds to a MAP estimate for θ_{ijk} if we assume a Dirichlet prior distribution over the θ_{ijk} parameters, with equal-valued parameters. If l is set to 1, this approach is called Laplace smoothing.

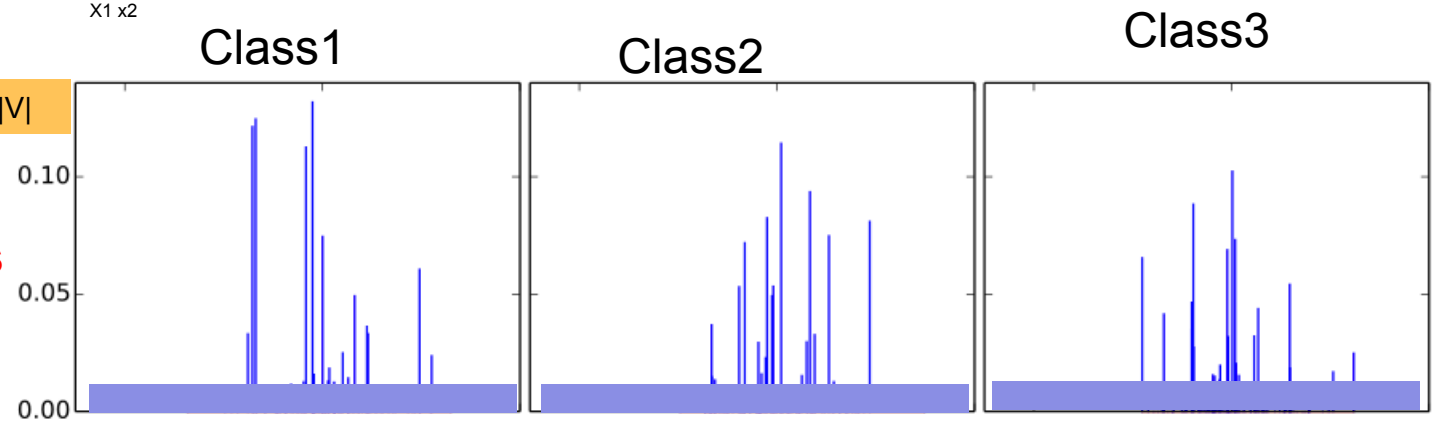
ML Estimates
Learning a NB Via Maximum Likelihood

MAP Estimates
Learning a NB Via Bayesian hierarchical model

MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|
+
Training data

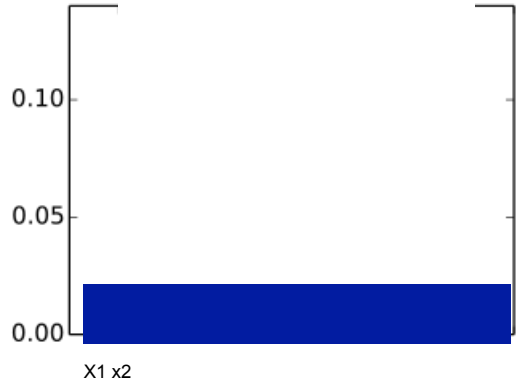


Pr(X=xj|Y=yk) = 1/|V|

MAP Estimates

Huge Probability mass assigned to background model:
Bias versus variance?

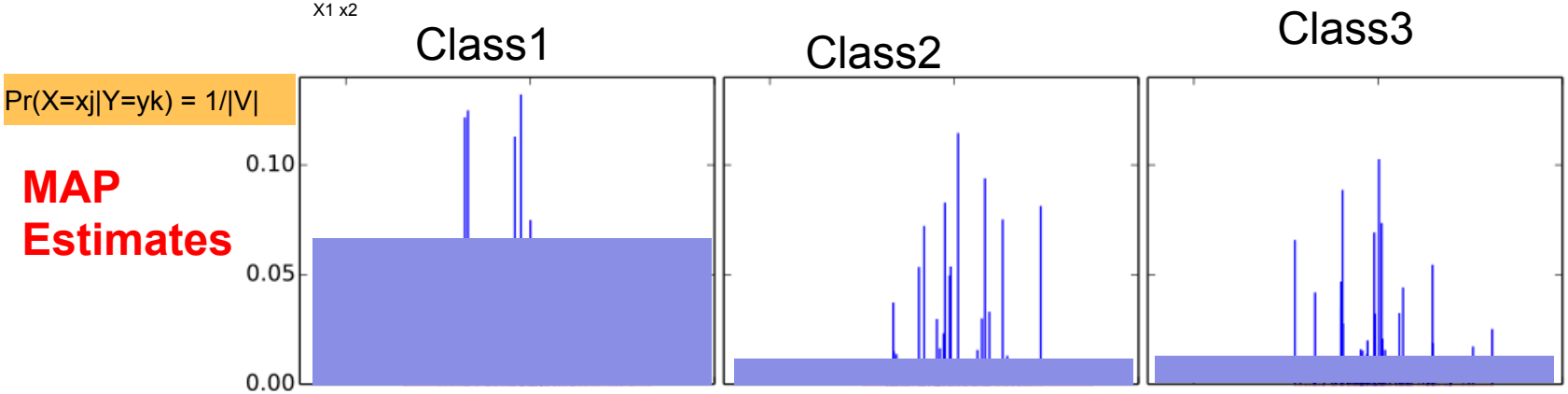
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

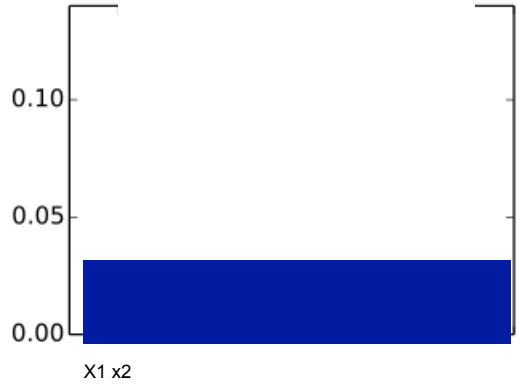
+

Training data



Huge Probability mass assigned to background model:
Bias versus variance?

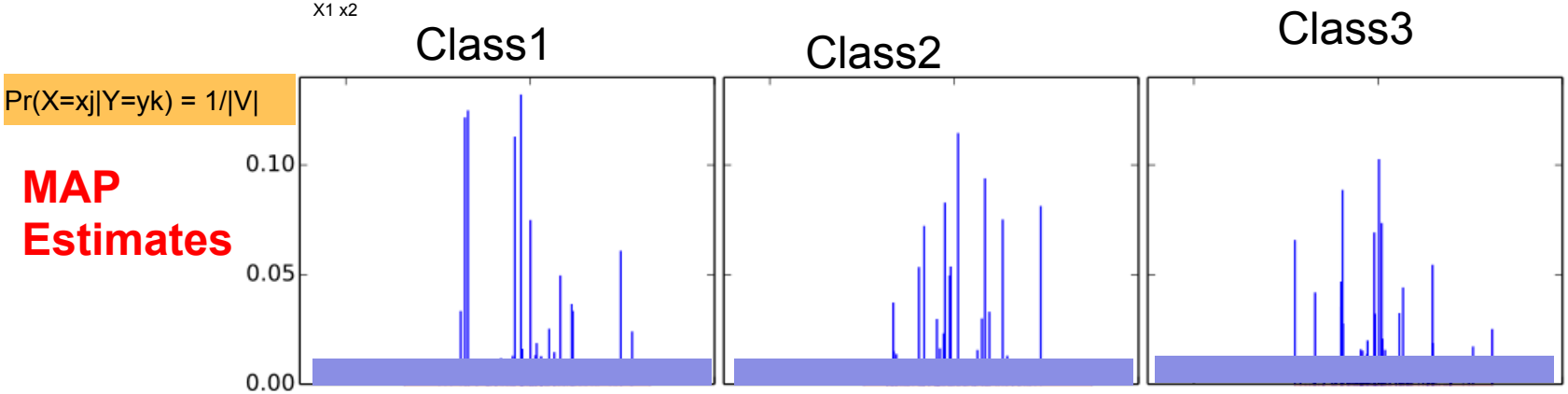
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

+

Training data



Huge Probability mass assigned to background model:
Bias versus variance? High Bias!

Class Prior Estimates

Maximum likelihood estimates for π_k are

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|} \tag{8}$$

where $|D|$ denotes the number of elements in the training set D.

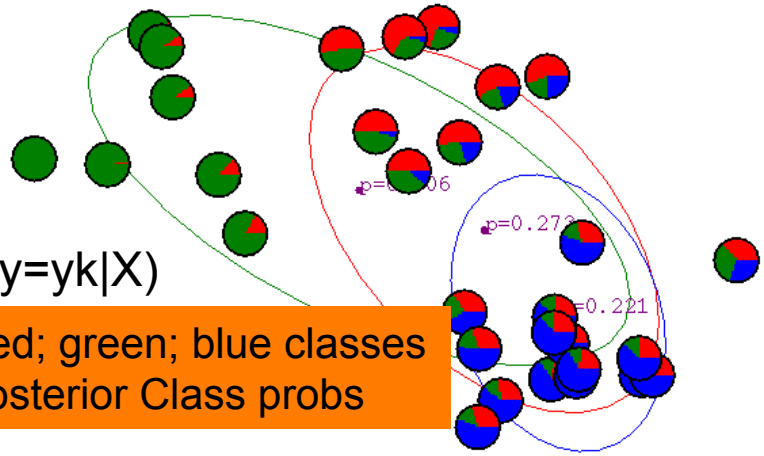
Alternatively, we can obtain a smoothed estimate, or equivalently a MAP estimate based on a Dirichlet prior over the π_k parameters assuming equal priors on each π_k, by using the following expression

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + l}{|D| + lK} \tag{9}$$

where K is the number of distinct values Y can take on, and l again determines the strength of the prior assumptions relative to the observed data D.

Naïve Bayes Classifier

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k) P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j) P(X_1, X_2, .., X_N | Y=y_j)}$$



Pr(y=yk|X)

Red; green; blue classes
Posterior Class probs

$$= \frac{P(Y=y_k) \Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j) \Pi_i P(X_i | Y=y_j)}$$

$$Y \leftarrow argmax_{y_k} P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

Naïve Bayes Classifier for Text

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j)P(X_1, X_2, .., X_N | Y=y_j)}$$

Y_1
Y_2

$$= \frac{P(Y=y_k)\Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i | Y=y_j)}$$

Pr("corporation"|Class=Business) = 1/100
10,000Words in the 10 business documents
"corporation" occurs 100 times

$$Y \leftarrow argmax_{y_k} P(Y = y_k)\Pi_i P(X_i | Y = y_k)$$

argmax_yk means find the value of yk that maximises the expression

Probability Basics

- Prior, conditional and joint probability
  - Prior probability: $P(X)$
  - Conditional probability: $P(X_1 \mid X_2), P(X_2 \mid X_1)$
  - Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
  - Relationship: $P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) = P(X_1 \mid X_2)P(X_2)$
  - Independence: $P(X_2 \mid X_1) = P(X_2), P(X_1 \mid X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$
- Bayesian Rule

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \qquad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Probabilistic Classification

- Establishing a probabilistic model for classification
  - Discriminative model

$$P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

  - Generative model

$$P(\mathbf{X} \mid C) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

- MAP classification rule
  - MAP: Maximum A Posterior
  - Assign x to c^* if $P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*, \; c = c_1, \cdots, c_L$
- Generative classification with the MAP rule
  - Apply Bayesian rule to convert $P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \propto P(\mathbf{X} \mid C)P(C)$

Naive Bayes for Discrete-Valued Inputs

When the n input attributes X_i each take on J possible discrete values, and Y is a discrete variable taking on K possible values, then our learning task is to estimate two sets of parameters. The first is

$$\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k) \tag{4}$$

for each input attribute X_i, each of its possible values x_{ij}, and each of the possible values y_k of Y. Note there will be nJK such parameters, and note also that only $n(J-1)K$ of these are independent, given that they must satisfy $1 = \sum_j \theta_{ijk}$ for each pair of i, k values.

In addition, we must estimate parameters that define the prior probability over Y:

$$\pi_k \equiv P(Y = y_k) \tag{5}$$

Note there are K of these parameters, $(K-1)$ of which are independent.

We can estimate these parameters using either maximum likelihood estimates (based on calculating the relative frequencies of the different events in the data), or using Bayesian MAP estimates (augmenting this observed data with prior distributions over the values of these parameters).

Maximum likelihood estimates for θ_{ijk} given a set of training examples D are given by

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}} \quad (6)$$

where the $\#D\{x\}$ operator returns the number of elements in the set D that satisfy property x.

One danger of this maximum likelihood estimate is that it can sometimes result in θ estimates of zero, if the data does not happen to contain any training examples satisfying the condition in the numerator. To avoid this, it is common to use a "smoothed" estimate which effectively adds in a number of additional "hallucinated" examples, and which assumes these hallucinated examples are spread evenly over the possible values of X_i. This smoothed estimate is given by

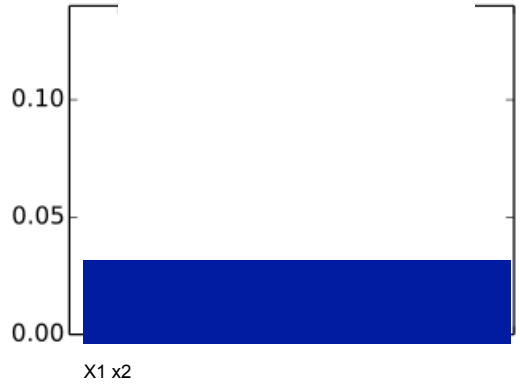
$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\} + l}{\#D\{Y = y_k\} + lJ} \quad (7)$$

where J is the number of distinct values X_i can take on, and l determines the strength of this smoothing (i.e., the number of hallucinated examples is lJ). This expression corresponds to a MAP estimate for θ_{ijk} if we assume a Dirichlet prior distribution over the θ_{ijk} parameters, with equal-valued parameters. If l is set to 1, this approach is called Laplace smoothing.

ML Estimates
Learning a NB Via Maximum Likelihood

MAP Estimates
Learning a NB Via Bayesian hierarchical model

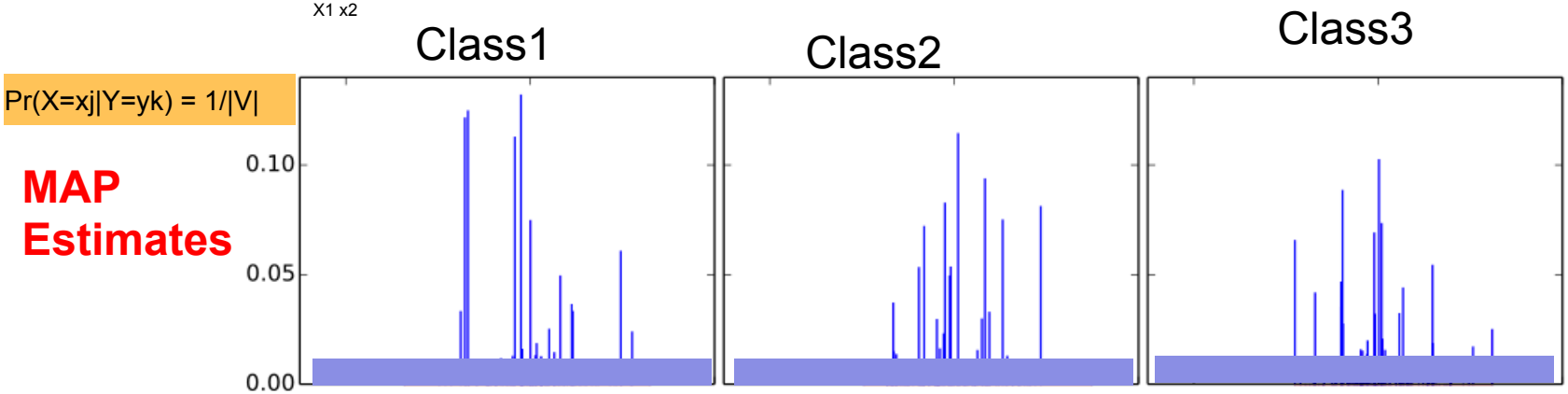
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

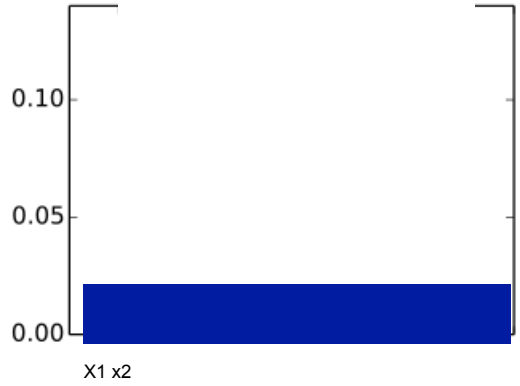
+

Training data



Huge Probability mass assigned to background model:
Bias versus variance?

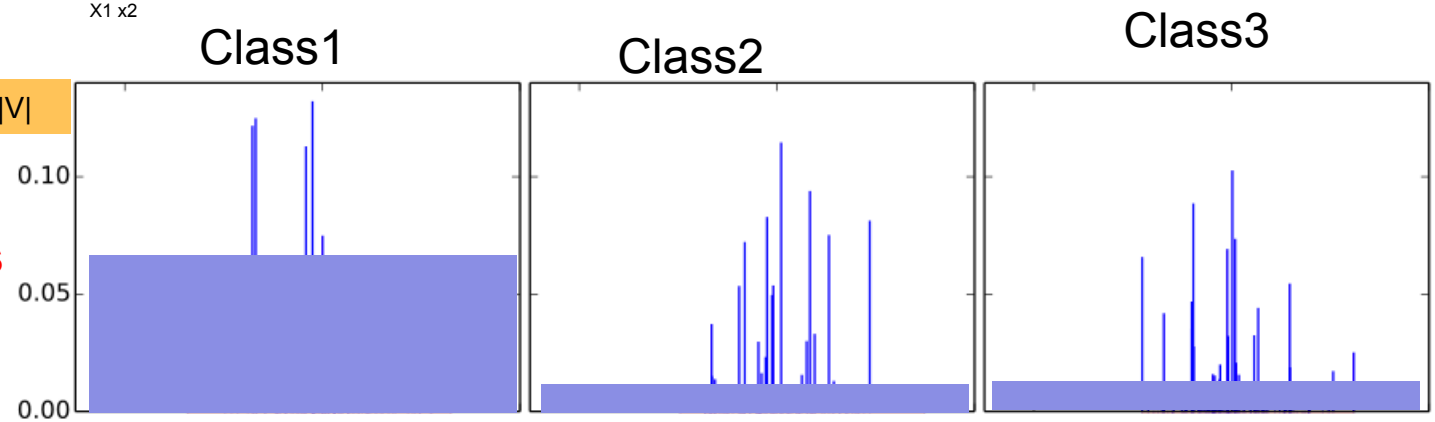
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

+

Training data

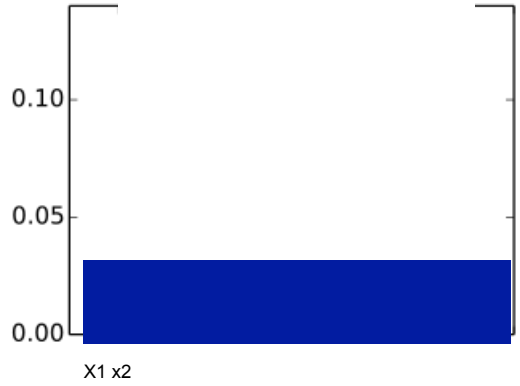


Pr(X=xj|Y=yk) = 1/|V|

MAP Estimates

Huge Probability mass assigned to background model:
Bias versus variance?

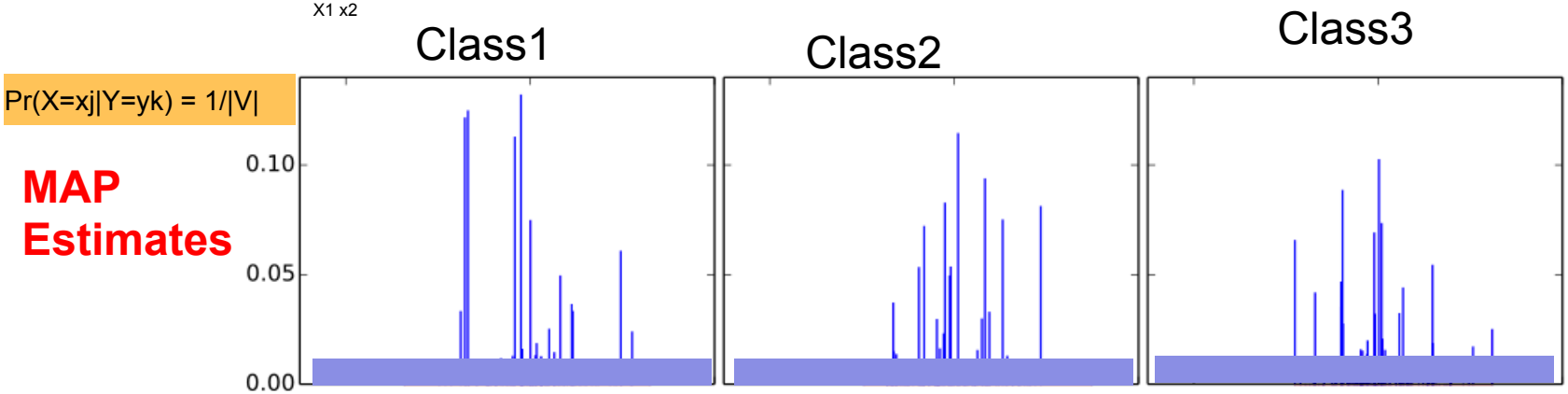
MAP Estimates: Smoothed Probabilities



Uniform Prior: Pr(X=xj|Y=yk) = 1/|V|

+

Training data



Huge Probability mass assigned to background model:
Bias versus variance? High Bias!

Class Prior Estimates

Maximum likelihood estimates for π_k are

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|} \tag{8}$$

where $|D|$ denotes the number of elements in the training set D.

Alternatively, we can obtain a smoothed estimate, or equivalently a MAP estimate based on a Dirichlet prior over the π_k parameters assuming equal priors on each π_k, by using the following expression

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + l}{|D| + lK} \tag{9}$$

where K is the number of distinct values Y can take on, and l again determines the strength of the prior assumptions relative to the observed data D.

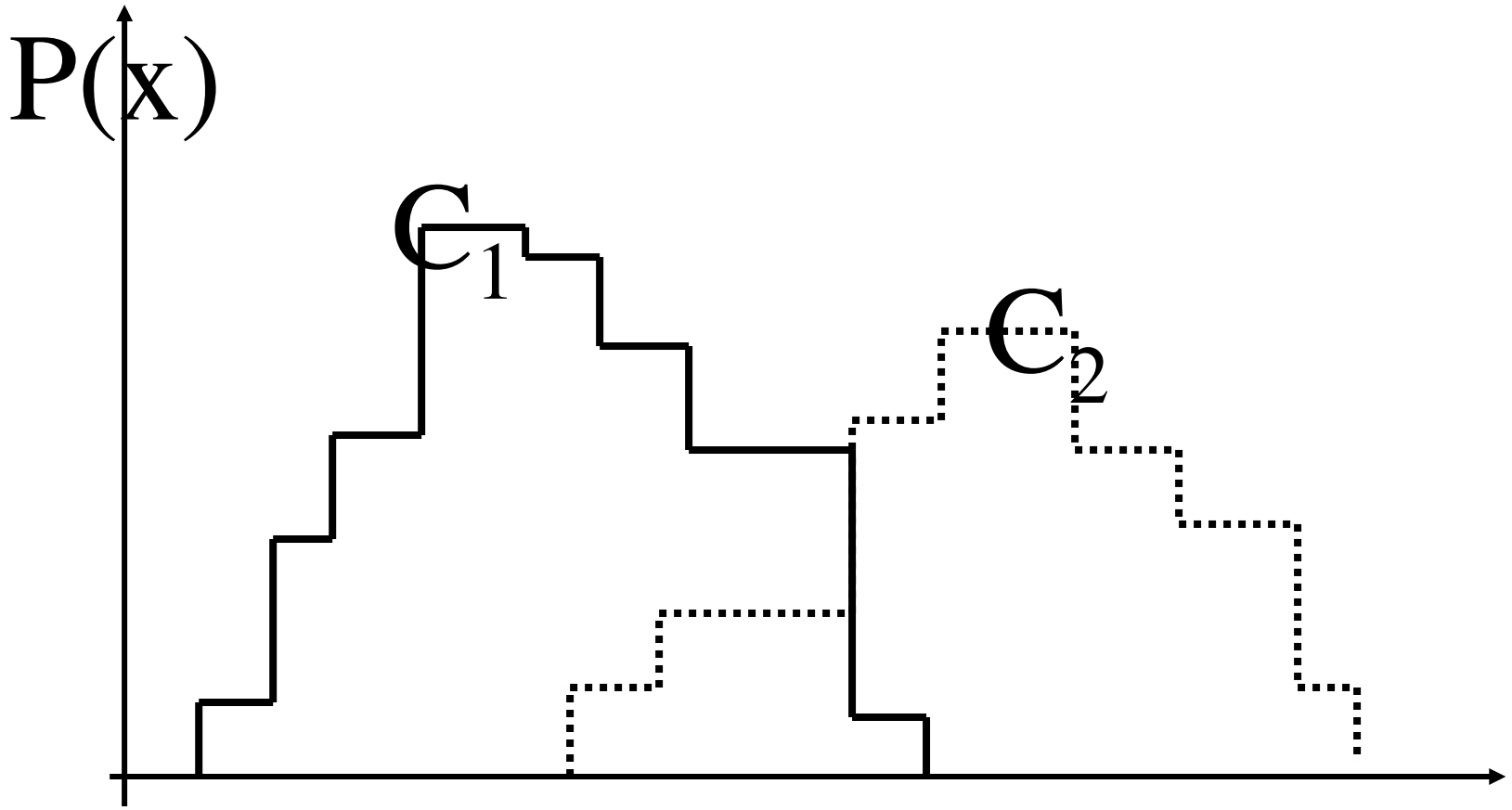
Naive Bayes for Continuous Inputs

- When the Xi are continuous we must choose some other way to represent the distributions
- P(Xi|Y)=Gaussian(Mu, Sigma).
- One common approach is to assume that for each possible discrete value yk of Y, the distribution of each continuous Xi is Gaussian, and is defined by a mean and standard deviation specific to Xi and yk.
- In order to train such a Naïve Bayes classifier we must therefore estimate the mean and standard deviation ians:

$$\mu_{ik} = E[X_i | Y = y_k]$$

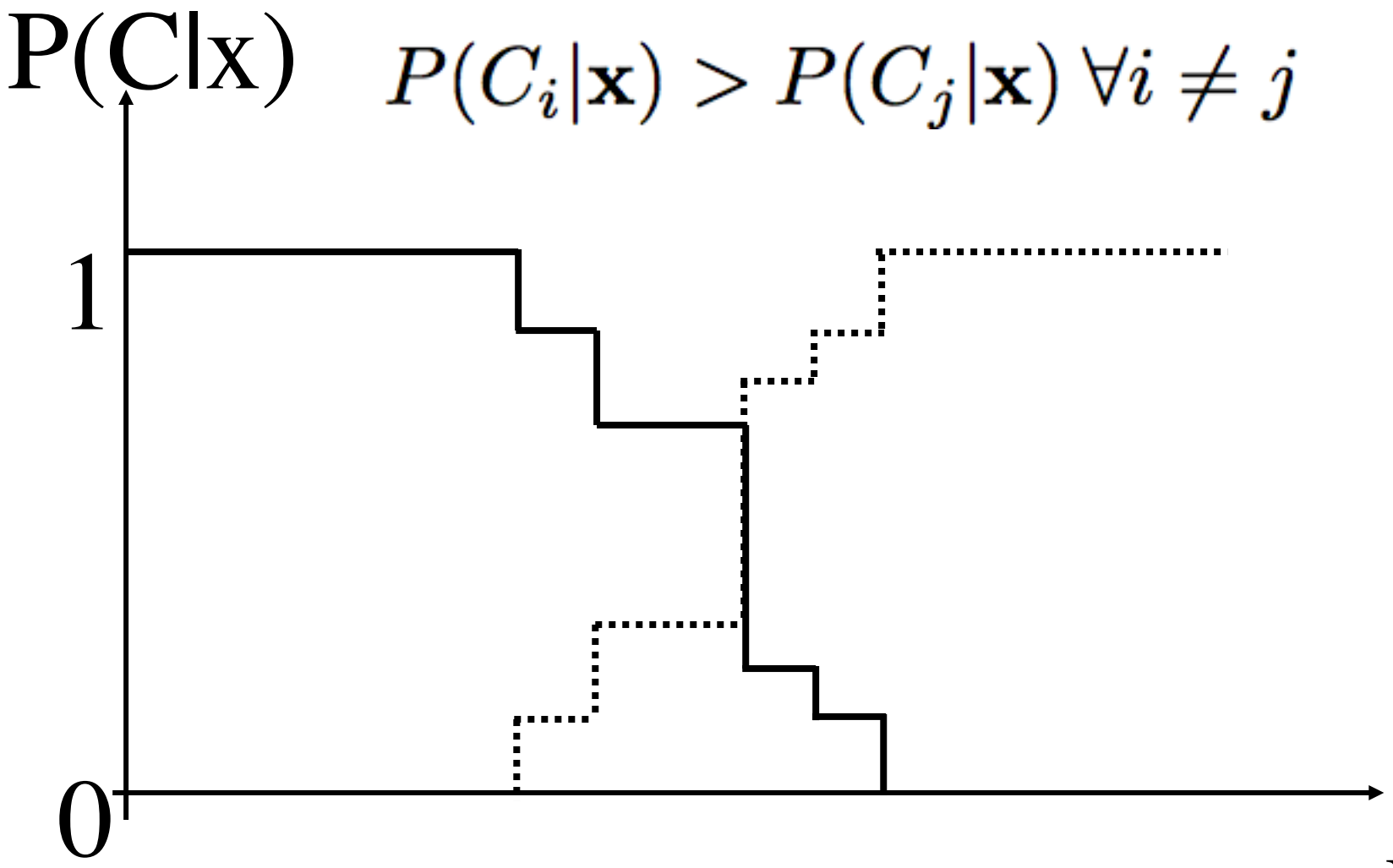
$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 | Y = y_k]$$

Feature Histograms



Slide by Stephen Marsland

Posterior Probability



Slide by Stephen Marsland

MLE-based Estimates

Again, we can use either maximum likelihood estimates (MLE) or maximum a posteriori (MAP) estimates for these parameters. The maximum likelihood estimator for μ_{ik} is

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k) \tag{13}$$

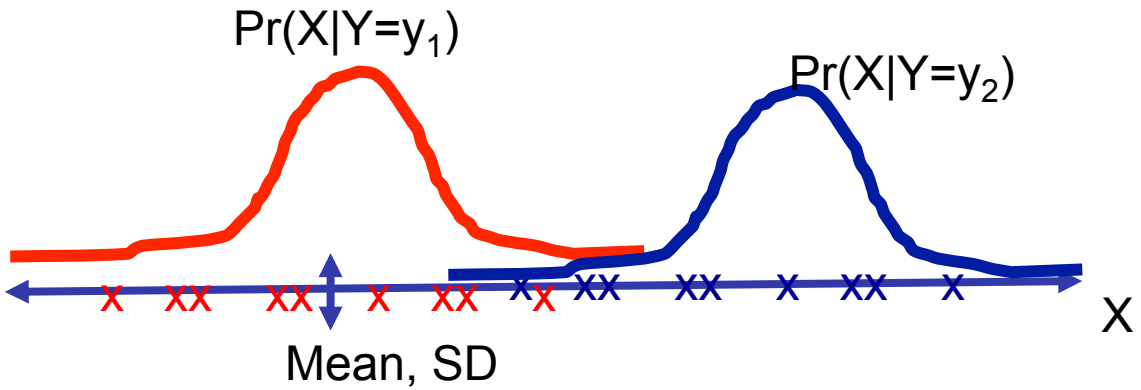
where the superscript j refers to the jth training example, and where $\delta(Y = y_k)$ is 1 if $Y = y_k$ and 0 otherwise. Note the role of δ here is to select only those training examples for which $Y = y_k$.

The maximum likelihood estimator for σ_{ik}^2 is

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k) \tag{14}$$

Estimate μ, σ from data

$$\mu_{ik} = E[X_i | Y = y_k]$$
$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 | Y = y_k]$$



Continuous Inputs: Pr(Y|X) ~ N(μ, σ)

If X is Normally distributed with mean μ and standard deviation σ, we write

$$\boxed{X \sim N(\mu, \sigma^2)}$$

μ and σ are the **parameters** of the distribution.

The probability density of the Normal distribution is given by

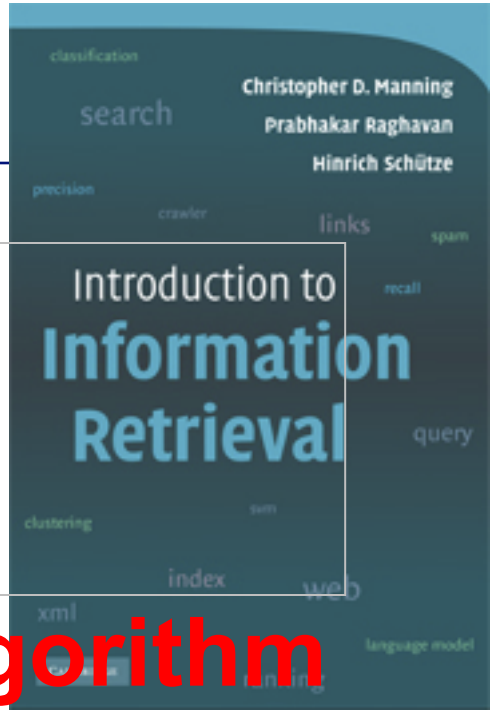
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-(x-\mu)^2/2\sigma^2}$$

For the purposes of this course we do not need to use this expression. It is included here for future reference.

Naïve Bayes

- **Combine discrete input variables with continuous input variables?**
  - Naïve Bayes, Decision trees
- **Whereas these requires feature transformations**
  - Logistic regression: one hot-encoding

- **YES**



The Naïve Bayes algorithm

Some slides Adapted from Lectures by

Prabhakar Raghavan (Yahoo and Stanford) and Christopher Manning (Stanford)

Reading material

- **PDF and HTML versions of Chapter 13 are available here**
  - [Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.] http://nlp.stanford.edu/IR-book/
  - http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture5.pdf
  - http://www.cs.cmu.edu/~tom/10701_sp11/lectures.shtml

Spam filtering: Another text classification task

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

==========
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
==========

Text classification: Naïve Bayes Text Classification

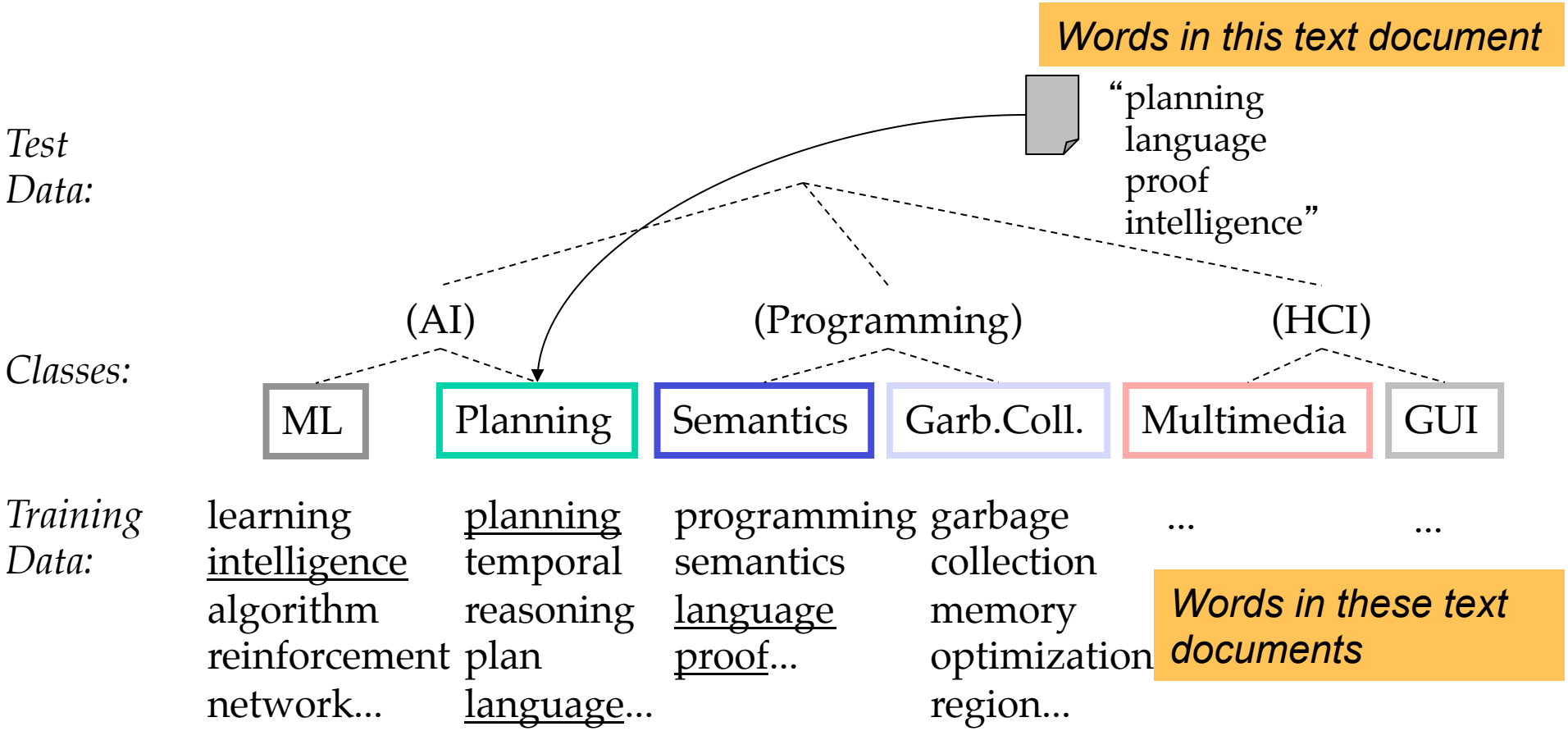
- **Today:**
  - Introduction to Text Classification
    - Also widely known as "text categorization".
  - Probabilistic Language Models
  - Naïve Bayes text classification
    - Multinomial
    - Bernoulli
  - Feature Selection

Categorization/Classification

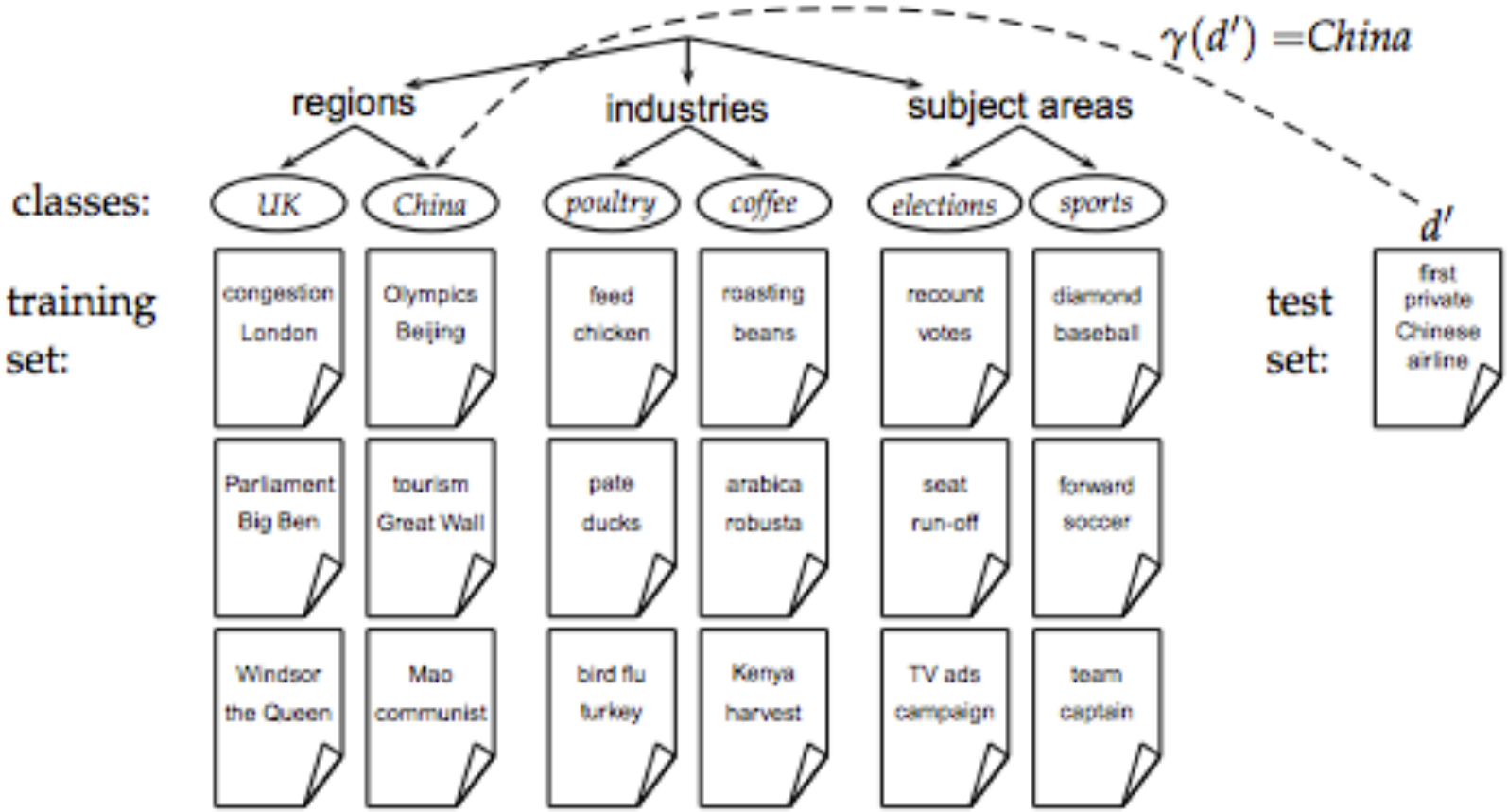
- **Given:**
  - A description of an instance, $x \in X$, where X is the *instance language* or *instance space*.
    - *Issue*: how to represent text documents.
  - A fixed set of classes:

    $C = \{c_1, c_2, \ldots, c_J\}$
- **Determine:**
  - The category of x: $c(x) \in C$, where $c(x)$ is a *classification function* whose domain is X and whose range is C.
    - We want to know how to build classification functions ("classifiers").

Document Classification



(Note: in real life there is often a hierarchy, not present in the above problem statement; and also, you get papers on ML approaches to Garb. Coll.)



► **Figure 13.1** Classes, training set, and test set in text classification .

More Text Classification Examples:

Many search engine functionalities use classification

Assign labels to each document or web-page:

- **Labels are most often topics such as Yahoo-categories**
  e.g., "finance," "sports," "news>world>asia>business"
- **Labels may be genres**
  e.g., "editorials" "movie-reviews" "news"
- **Labels may be opinion on a person/product**
  e.g., "like", "hate", "neutral"
- **Labels may be domain-specific**
  e.g., "interesting-to-me" : "not-interesting-to-me"
  e.g., "contains adult language" : "doesn't"
  e.g., language identification: English, French, Chinese, …
  e.g., search vertical: about Linux versus not
  e.g., "link spam" : "not link spam"

Classification Methods (1)

- **Manual classification**
  - Used by Yahoo! (originally; now downplayed), Looksmart, about.com, ODP, PubMed
  - Very *accurate* when job is done by experts
  - *Consistent* when the problem size and team is small
  - *Difficult and expensive to scale*
    - Means we need automatic classification methods for big problems

Classification Methods (2)

- **Automatic document classification**
  - Hand-coded rule-based systems
    - Used by CS dept's spam filter, Reuters, CIA, etc.
    - Companies (Verity) provide "IDE" for writing such rules
      - E.g., assign category if document contains a given boolean combination of words
    - Standing queries: Commercial systems have complex query languages (everything in IR query languages + accumulators)
      - Accuracy is often very high if a rule has been carefully refined over time by a subject expert
    - Building and maintaining these rules is expensive

Classification Methods (3)

- **Supervised learning of a document-label assignment function**
  - Many systems partly rely on machine learning (Autonomy, MSN, Verity, Enkata, Yahoo!, …)
    - k-Nearest Neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (new, more powerful)
    - … plus many other methods
    - No free lunch: requires hand-classified training data
    - But data can be built up (and refined) by amateurs
- **Note that many commercial systems use a mixture of methods**

Recall a few probability basics

- **For events *a* and *b*:**
- **Bayes' Rule**

Pr(Spade) = ¼
Pr(King) = 1/13
Pr(Spade) and Pr(King) = 1/52
Pr(king|spade) = 1/13
Pr(king|spade) = Pr(Spade) and Pr(King) / Pr(Spac
Pr(spade|King = Pr(Spade) and Pr(King) / Pr(King

$$p(a,b) = p(a \cap b) = p(a \mid b)p(b) = p(b \mid a)p(a)$$

$$p(\bar{a} \mid b)p(b) = p(b \mid \bar{a})p(\bar{a})$$

$$p(a \mid b) = \frac{p(b \mid a)p(a)}{p(b)} = \frac{p(b \mid a)p(a)}{\sum_{x=a,\bar{a}} p(b \mid x)p(x)}$$

Prior

Posterior

- **Odds:**

$$O(a) = \frac{p(a)}{p(\bar{a})} = \frac{p(a)}{1 - p(a)}$$

Bayesian Methods

- **Learning and classification methods based on probability theory.**
  - Bayes theorem plays a critical role in probabilistic learning and classification.
- **Build a *generative model* that approximates how data is produced.**
- **Uses *prior* probability of each category given no information about an item.**
- **Categorization produces a *posterior* probability distribution over the possible categories given a description of an item (and prior probabilities).**

Bayes' Rule

Pr(Spade) = ¼
Pr(King) = 1/13
Pr(Spade) and Pr(King) = 1/52
Pr(king|spade) = 1/13
Pr(king|spade) = Pr(Spade) and Pr(King) / Pr(Spac
Pr(spade|King = Pr(Spade) and Pr(King) / Pr(King

Theorem of total probabilites
P(D) =P(D|C=TRUE) + P(D|C=FALSE)

$$P(C,D) = P(C \mid D)P(D) = P(D \mid C)P(C)$$

$$P(C \mid D) = \frac{P(D \mid C)P(C)}{P(D)}$$

TASK P(class=china|{T, C}) vs P(class=not china|{T, C})

Naive Bayes Classifiers

***Task*: Classify a new instance *D* based on a tuple of attribute values $D = \langle x_1, x_2, \ldots, x_n \rangle$ into one of the classes $c_j \in C$**

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

MAP = Maximum Aposteriori Probability

Naïve Bayes Classifier: Naïve Bayes Assumption

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - O(|X|^n•|C|) parameters
  - Could only be estimated if a very, very large number of training examples was available.

Naïve Bayes Conditional Independence Assumption:

- **Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i|c_j)$.**

Naïve Bayes

- **Two different ways to set up a Naïve Bayes classifier**
- **Different type of input variables (X_i):**
  - Multivariate Bernoulli or Bernoulli Naïve Bayes Model

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

$$\hat{P}(\text{Chinese}|c) = (3+1)/(3+2) = 4/5$$

Count 3 documents with Chinese for YES class

  - Multinomial Naïve Bayes

Parameter estimation for Naïve Bayes

- **Multivariate Bernoulli model:** X_w = t if word present document

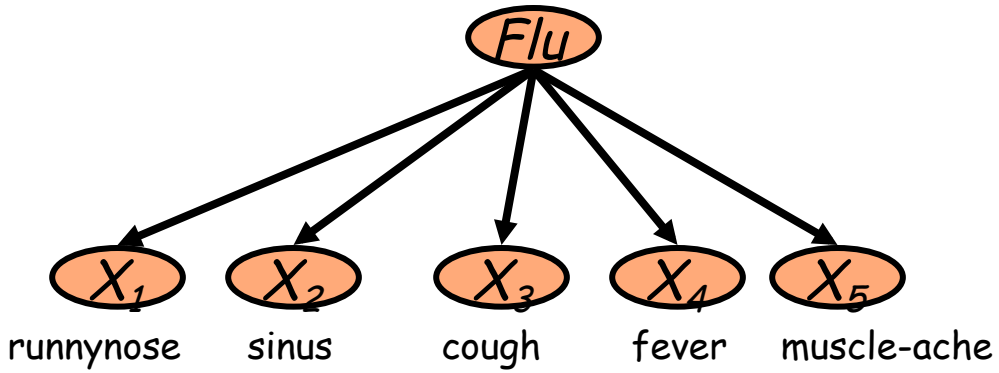
$$\hat{P}(X_w = t \mid c_j) = \text{fraction of documents of topic } c_j \text{ in which word } w \text{ appears}$$

- **Multinomial model:**

$$\hat{P}(X_i = w \mid c_j) = \text{fraction of times in which word } w \text{ appears across all documents of topic } c_j$$

  - Can create a mega-document for topic *j* by concatenating all documents on this topic
  - Use frequency of *w* in mega-document

The Naïve Bayes Classifier

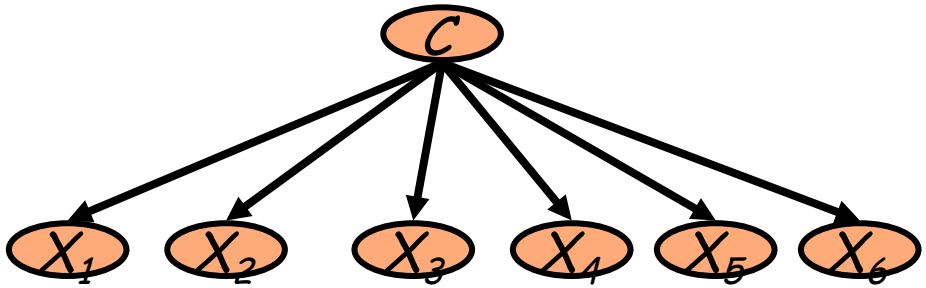


- **Conditional Independence Assumption: Features (term presence) are *independent* of each other given the class:**

$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

- **This model is appropriate for binary variables**
  - Multivariate Bernoulli model

Learning the Model

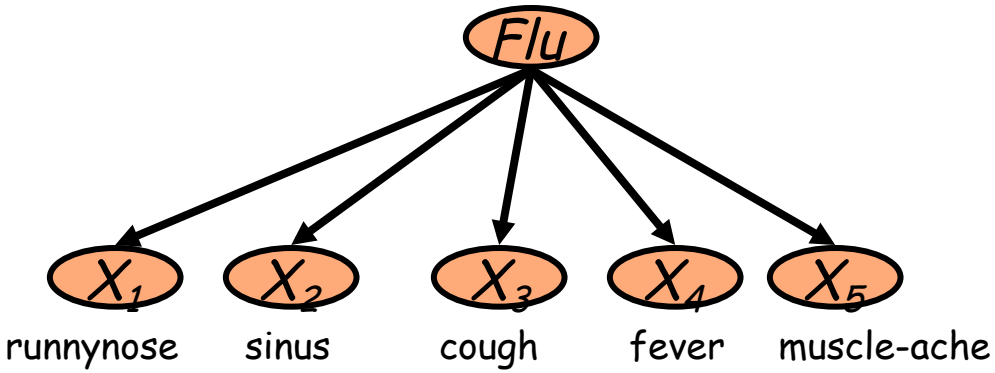


- ***First attempt*: maximum likelihood estimates**
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

Problem with Max Likelihood



$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

- **What if we have seen no training cases where patient had no flu and muscle aches?**

$$\hat{P}(X_5 = t \mid C = nf) = \frac{N(X_5 = t, C = nf)}{N(C = nf)} = 0$$

- **Zero probabilities cannot be conditioned away, no matter the other evidence!**

$$\ell = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Bernoulli model: Smoothing to Avoid Overfitting and Zero probabilities

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

# of values of X_i

Laplace smoothing for Bernoulli model
1 for every vocabulary term – k is size of the vocabulary (view as prior on vocabulary);

Classification

- **Multinomial vs Multivariate Bernoulli?**
- **Multinomial model is almost always more effective in text applications!**
- **See *IR Book* sections 13.2 and 13.3 for worked examples with each model**

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Vocabulary = {Chinese, Beijing, Shanghai, Macao, Tokyo}

Example 13.2: Applying the Bernoulli model to the example in Table 13.1, we have the same estimates for the priors as before: $\hat{P}(c) = 3/4$, $\hat{P}(\overline{c}) = 1/4$. The conditional probabilities are:

Japan does NOT occur class C

$$\begin{aligned}
\hat{P}(\text{Chinese}|c) &= (3+1)/(3+2) = 4/5\\
\hat{P}(\text{Japan}|c) = \hat{P}(\text{Tokyo}|c) &= (0+1)/(3+2) = 1/5\\
\hat{P}(\text{Beijing}|c) = \hat{P}(\text{Macao}|c) = \hat{P}(\text{Shanghai}|c) &= (1+1)/(3+2) = 2/5\\
\hat{P}(\text{Chinese}|\overline{c}) &= (1+1)/(1+2) = 2/3\\
\hat{P}(\text{Japan}|\overline{c}) = \hat{P}(\text{Tokyo}|\overline{c}) &= (1+1)/(1+2) = 2/3\\
\hat{P}(\text{Beijing}|\overline{c}) = \hat{P}(\text{Macao}|\overline{c}) = \hat{P}(\text{Shanghai}|\overline{c}) &= (0+1)/(1+2) = 1/3
\end{aligned}$$

The denominators are $(3+2)$ and $(1+2)$ because there are three documents in c and one document in $\overline{c}$ and because the constant B in Equation (13.7) is 2 – there are two cases to consider for each term, occurrence and nonoccurrence.

The scores of the test document for the two classes are

$$\begin{aligned}
\hat{P}(c|d_5) &\propto \hat{P}(c) \cdot \hat{P}(\text{Chinese}|c) \cdot \hat{P}(\text{Japan}|c) \cdot \hat{P}(\text{Tokyo}|c)\\
&\quad \cdot (1-\hat{P}(\text{Beijing}|c)) \cdot (1-\hat{P}(\text{Shanghai}|c)) \cdot (1-\hat{P}(\text{Macao}|c))\\
&= 3/4 \cdot 4/5 \cdot 1/5 \cdot 1/5 \cdot (1-2/5) \cdot (1-2/5) \cdot (1-2/5)\\
&\approx 0.005
\end{aligned}$$

and, analogously,

$$\begin{aligned}
\hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot (1-1/3) \cdot (1-1/3) \cdot (1-1/3)\\
&\approx 0.022
\end{aligned}$$

Thus, the classifier assigns the test document to $\overline{c}$ = *not-China*. When looking only at binary occurrence and not at term frequency, Japan and Tokyo are indicators for $\overline{c}$ ($2/3 > 1/5$) and the conditional probabilities of Chinese for c and $\overline{c}$ are not different enough (4/5 vs. 2/3) to affect the classification decision.

Bernoulli NB: Smoothing is based on the number of classes

- 6 terms in Vocabulary 6 terms in calculation of P(C|X)
- Count single occurrence of term, .e.g., Chinese is just counted once

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Decision Rule for Bernoulli Model

Use all vocabulary for classification (regardless if the word occurred or not in the test example)

► **Table 13.3** Multinomial versus Bernoulli model.

	multinomial model	Bernoulli model
event model	generation of token	generation of document
random variable(s)	$X = t$ iff t occurs at given pos	$U_t = 1$ iff t occurs in doc
document representation	$d = \langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle, t_k \in V$	$d = \langle e_1, \ldots, e_i, \ldots, e_M \rangle$, $e_i \in \{0, 1\}$
parameter estimation	$\hat{P}(X = t\|c)$	$\hat{P}(U_i = e\|c)$
decision rule: maximize	$\hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(X = t_k\|c)$	$\hat{P}(c) \prod_{t_i \in V} \hat{P}(U_i = e_i\|c)$
multiple occurrences	taken into account	ignored
length of docs	can handle longer docs	works best for short docs
# features	can handle more	works best with fewer
estimate for term the	$\hat{P}(X = \text{the}\|c) \approx 0.05$	$\hat{P}(U_{\text{the}} = 1\|c) \approx 1.0$

- **Multinomial Naïve Bayes**

Stochastic Language Models

- **Models *probability* of generating strings (each word in turn) in the language (commonly all strings over ∑). E.g., unigram model**

Model M

0.2	the
0.1	a
0.01	man
0.01	woman
0.03	said
0.02	likes
...	

the	man	likes	the	woman
0.2	0.01	0.02	0.2	0.01

multiply

P(s | M) = 0.00000008

Stochastic Language Models

- **Model *probability* of generating any string**

Model M1		Model M2	
0.2	the	0.2	the
0.01	class	0.0001	class
0.0001	sayst	0.03	sayst
0.0001	pleaseth	0.02	pleaseth
0.0001	yon	0.1	yon
0.0005	maiden	0.01	maiden
0.01	woman	0.0001	woman

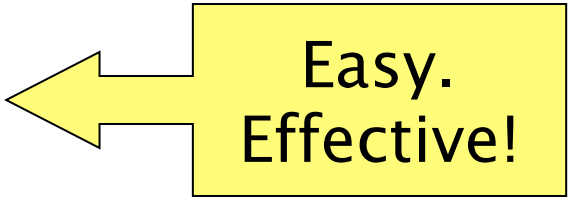
the	class	pleaseth	yon	maiden
0.2	0.01	0.0001	0.0001	0.0005
0.2	0.0001	0.02	0.1	0.01

$$P(s|M2) > P(s|M1)$$

Unigram and higher-order models

- 

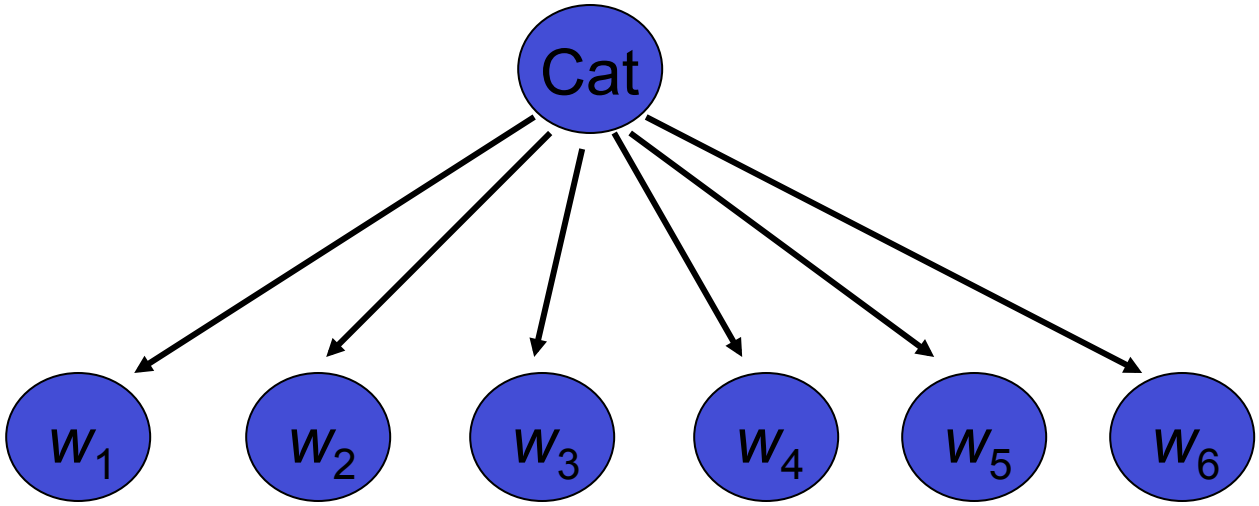
- **Unigram Language Models**



- **Bigram (generally, *n*-gram) Language Models**

- **Other Language Models**
  - Grammar-based models (PCFGs), etc.
    - Probably not the first thing to try in IR

Naïve Bayes via a class conditional language model = multinomial NB



- **Effectively, the probability of each class is done as a class-specific unigram language model**

Using Multinomial Naive Bayes Classifiers to Classify Text: Basic method

- **Attributes are text positions, values are words.**

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_i P(x_i \mid c_j)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(c_j) P(x_1 = \text{"our"} \mid c_j) \cdots P(x_n = \text{"text"} \mid c_j)$$

- Still too many possibilities
- Assume that classification is *independent* of the positions of the words
  - Use same parameters for each position
  - Result is bag of words model (over tokens not types)

Bag of words: Conditional independence of other words, and positions of words

- Even when assuming conditional independence, we still have too many parameters for the multinomial model if we assume a different probability distribution for each position k in the document. The position of a term in a document by itself does not carry information about the class. Although there is a difference between *China sues France* and *France sues China*, the occurrence of China in position 1 versus position 3 of the document is not useful in NB classification because we look at each term separately. The conditional independence assumption commits us to this way of processing the evidence.

  Also, if we assumed different term distributions for each position k, we would have to estimate a different set of parameters for each k. The probability of bean appearing as the first term of a *coffee* document could be different from it appearing as the second term, and so on. This again causes problems in estimation owing to data sparseness.

  For these reasons, we make a second independence assumption for the multinomial model, *positional independence*: The conditional probabilities for a term are the same independent of position in the document.

$$P(X_{k_1} = t|c) = P(X_{k_2} = t|c)$$

  for all positions k_1, k_2, terms t and classes c. Thus, we have a single distribution of terms that is valid for all positions k_i and we can use X as its symbol.[4] Positional independence is equivalent to adopting the bag of words model, which we introduced in the context of ad hoc retrieval in Chapter 6 (page 117).

Naïve Bayes: Learning Algorithm

- **From training corpus, extract *Vocabulary***
- **Calculate required $P(c_j)$ and $P(x_k \mid c_j)$ terms**
  - For each c_j in C do
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j
    - $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- $Text_j \leftarrow$ single document containing all $docs_j$
- for each word x_k in *Vocabulary*
  - $n_k \leftarrow$ number of occurrences of x_k in $Text_j$
  - $$P(x_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

n = total number of tokens (alpha is "tuning/smoothing" factor that can be set to 1)
Multinomial Conditional Independence Naïve Bayes: Learning from Training Set

Naïve Bayes: Classifying

- positions ← all word positions in current document which contain tokens found in *Vocabulary*
- Return c_{NB}, where

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Naïve Bayes: Classifying

- **positions ← all word positions in current document which contain tokens found in *Vocabulary***
- **Return c_{NB}, where**

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Classify a NEW document D with 100 words (not seen in the class=TRUE)
P(W|Class=TRUE) = $1/(10^6 + 10^7)$ ##default probability of a word given *Class=TRUE*

Assume a Vocabulary of 10^6 words; number of words in *class=TRUE* is 10^7
THEN
Pr(Class = TRUE | D) ~ $(1/(10^6 + 10^7))^{100}$ X Prior

Underflow Prevention: log space

- **Multiplying lots of probabilities, which are between 0 and 1, can result in floating-point underflow.**
- **Since log(*xy*) = log(*x*) + log(*y*), it is better to perform all computations *by summing logs of probabilities rather than multiplying probabilities*.**
- **Class with highest final un-normalized log probability score is still the most probable.**

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j) \quad \#\#\text{Log}(\frac{a}{b}) = \log(a) - \log(b); Note: \log(1) = 0$$

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

- **Note that model is now just max of sum of weights…**

Log Rules Refresher

$$\log_a(bc) = \log_a(b) + \log_a(c)$$

$$\log_a(b^c) = c\log_a(b)$$

$$\log_a(1/b) = -\log_a(b)$$

$$\log_a(1) = 0$$

$$\log_a(a) = 1$$

$$\log_a(a^r) = r$$

$$\log_{1/a}(b) = -\log_a(b)$$

$$\log_a(b)\log_b(c) = \log_a(c)$$

$$\log_b(a) = \frac{1}{\log_a(b)}$$

$$\log_{a^m}(a^n) = \frac{n}{m}, \qquad m \neq 0$$

http://grockit.com/blog/act-logarithms-explained/

Naïve Bayes: Classifying: Extreme example!

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

- **positions ← all word positions in current document which contain tokens found in *Vocabulary***
- **Return c_{NB}, where**

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j) \quad \#\#\text{Log}(\frac{a}{b}) = \log(a) - \log(b); Note: \log(1) = 0$$

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

In R
> log (1/(10^6+ 10^7)) *100
[1] -1621.341
> exp(-1621.341)
[1] 0 #super small number; we underflow

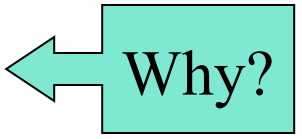
Classify a NEW document D with 100 words (all 100 words are not seen in the class=TRUE)
P(W|Class=TRUE) = 1/(10^6+ 10^7) ##default probability of a word given *Class=TRUE*

Assume a Vocabulary of 10^6 words; number of words in *class=TRUE* is 10^7
THEN
Pr(Class = TRUE | D) ~ (1/(10^6+ 10^7))^{100} X Prior

Approximate with log base 10 (assume (10^6+ 10^7) =10^7)
P(W|Class=TRUE) =~ (0 – log(10^7, 10)) X 100 = ~ -7
P(W|Class=TRUE) =~ -7 x 100 #in log space

Naive Bayes: Time Complexity

- **Training Time: $O(|D|L_d + |C||V|))$ where L_d is the average length of a document in D.**
  - Assumes V and all D_i, n_i, and n_{ij} pre-computed in $O(|D|L_d)$ time during one pass through all of the data.
  - Generally just $O(|D|L_d)$ since usually $|C||V| < |D|L_d$



- **Test Time: $O(|C|\ L_t)$ where L_t is the average length of a test document.**
  - Very efficient overall, linearly proportional to the time needed to just read in all the data.
  - Plus, robust in practice

- **Exercise**

Exercise: Multinomial Naive Bayes

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

$$P(C \mid D) = \frac{P(D \mid C)P(C)}{P(D)}$$

- Estimate parameters of Multinomial Naive Bayes classifier
- Classify test document

Exercise: Multinomial Naive Bayes

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Priors Pr(Class=China) = ¾ Pr(Class=Not China) = ¼
Class conditional probabilities; likelihoods
Pr(Chinese | Class=China) = 5/8
Pr(Chinese | Class=NOT China) = 1/3
What is the class of this document {Tokyo,chinese}
Pr(Class = China|{Tokyo,chinese}) = ¾ X 0/8 X 5/8 = 0
Pr(Class = Not China|{Tokyo,chinese}) = ¼ X 1/3 X 1/3 = 1/36
Pr(Class = China|{Tokyo,chinese})=0/ 0+ 1/36 = 0
Pr(Class = Not China|{Tokyo,chinese}) = 1/36/ 0+ 1/36 = 1

$$P(C \mid D) = \frac{P(D \mid C)P(C)}{P(D)}$$

Pr(Class = China|{Tokyo,chinese}) = ¾ X 0+ 1/8+6 X 5+1/8+6 = 0.023
Pr(Class = Not China|{Tokyo,chinese}) = ¼ X 1+1/3+6 X 1+1/3+6 = 0.012
Pr(Class = China|{Tokyo,chinese})=0.023 / (0.023 + 0.012) = 0.65
Pr(Class = Not China|{Tokyo,chinese}) = 1 – 0.65

Exercise: Multinomial Naive Bayes

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Pr(Class=China) = ¾ Pr(Class = not China) = ¼

$$P(C \mid D) = \frac{P(D \mid C)P(C)}{P(D)}$$

Pr(Chinese|Class = China) = 5 / 8, 1/8, 1/8, 1/8 =1
TEST = {Japan, Chinese}
Pr(class =China| {Japan, Chinese}) = Prior X Likelihood = ¾ X (5/14 X 1/14) = 0.022; PC|D) = 0.022/(0.022+0.0123)= 0.64
Pr(class = NOT China| {Japan, Chinese}) = Prior X Likelihood = 1/4 X (**1+1/(3+6)** X 1+1/(3+6) = 0.0123 = ; P(C|D) = 1 - 0.64

6 unique
Pr(Chinese|Class = China) = 5+1/ (8+6)

- Estimate parameters of Multinomial Naive Bayes classifier
- Classify test document

Example: Parameter estimates

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Conditional probabilities:

$$\begin{aligned}
\hat{P}(\textsc{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\
\hat{P}(\textsc{Tokyo}|c) = \hat{P}(\textsc{Japan}|c) &= (0+1)/(8+6) = 1/14 \\
\hat{P}(\textsc{Chinese}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\
\hat{P}(\textsc{Tokyo}|\overline{c}) = \hat{P}(\textsc{Japan}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

The denominators are (8 + 6) and (3 + 6) because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B is 6 as the vocabulary consists of six terms.

Example: Classification of d_5

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

$$\hat{P}(c|d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003$$

$$\hat{P}(\overline{c}|d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001$$

Thus, the classifier assigns the test document to c = *China*. The reason for this classification decision is that the three occurrences of the positive indicator CHINESE in d_5 outweigh the occurrences of the two negative indicators JAPAN and TOKYO.

Note: Two Models: Multivariate Bernoulli

- ***Model 1*: Multivariate Bernoulli**
  - One feature X_w for each word in dictionary
  - X_w = true in document d if w appears in d
  - Naive Bayes assumption:
    - Given the document's topic, appearance of one word in the document tells us nothing about chances that another word appears
- **This is the model used in the binary independence model in classic probabilistic relevance feedback in hand-classified data**

Two Models: Multinomial NB

- ***Model 2*: Multinomial = Class conditional unigram**
  - One feature X_i for each word pos in document
    - feature's values are all words in dictionary
  - Value of X_i is the word in position *i*
  - Naïve Bayes assumption:
    - Given the document's topic, word in one position in the document tells us nothing about words in other positions
  - Second assumption:
    - Word appearance does not depend on position

$$P(X_i = w \mid c) = P(X_j = w \mid c)$$

for all positions *i,j*, word *w*, and class *c*

- **Puzzle**

Example: Parameter estimates

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

taiwan

Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Conditional probabilities:

$$\begin{aligned}
\hat{P}(\text{CHINESE}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\
\hat{P}(\text{TOKYO}|c) = \hat{P}(\text{JAPAN}|c) &= (0+1)/(8+6) = 1/14 \\
\hat{P}(\text{CHINESE}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\
\hat{P}(\text{TOKYO}|\overline{c}) = \hat{P}(\text{JAPAN}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

The denominators are (8 + 6) and (3 + 6) because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B is 6 as the vocabulary consists of six terms.

Example: Parameter estimates

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Condition[illegible]ties:

$$\hat{P}(\text{CHINESE}|c) = [illegible]/(8+6) = 6/14 = 3/7$$

$$\hat{P}(\text{TOKYO}|c) = \hat{P}(\text{JAP}[illegible]) = (0+1)/(8+6) = 1/14$$

$$\hat{P}([illegible]|c) = (1+1)/(3+6) = 2/9$$

$$\hat{P}(\text{TO}[illegible]\ (\text{JAPAN}|\overline{c}) = (1+1)/(3+6) = 2/9$$

Which type of Naïve Bayes model is this?

The denominators are (8 + 6) and (3 + 6) because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B is 6 as the vocabulary consists of six terms.

Example: Parameter estimates

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Conditional probabilities:

$$\hat{P}(\text{CHINESE}|c) = (5+1)/(8+6) = 6/14 = 3/7$$

Which type of Naïve Bayes model is this?

Which type of Naïve Bayes model is this?

Multinomial….look at *Pr(Chinese|c)* is made up of 5 occurences of the word in the class c (versus 3 in the Bernoulli model for the same class conditional, *Pr(Chinese|c)*)

Underflow Prevention: log space

- Multiplying lots of probabilities, which are between 0 and 1, can result in floating-point underflow.
- Since log(*xy*) = log(*x*) + log(*y*), it is better to perform all computations *by summing logs of probabilities rather than multiplying probabilities*.
- Class with highest final un-normalized log probability score is still the most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

- Note that model is now just max of sum of weights…

Multinomial vs Bernoulli Naïve Bayes

► **Table 13.3** Multinomial versus Bernoulli model.

	multinomial model	Bernoulli model
event model	generation of token	generation of document
random variable(s)	$X = t$ iff t occurs at given pos	$U_t = 1$ iff t occurs in doc
document representation	$d = \langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle, t_k \in V$	$d = \langle e_1, \ldots, e_i, \ldots, e_M \rangle,$ $e_i \in \{0, 1\}$
parameter estimation	$\hat{P}(X = t\|c)$	$\hat{P}(U_i = e\|c)$
decision rule: maximize	$\hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(X = t_k\|c)$	$\hat{P}(c) \prod_{t_i \in V} \hat{P}(U_i = e_i\|c)$
multiple occurrences	taken into account	ignored
length of docs	can handle longer docs	works best for short docs
# features	can handle more	works best with fewer
estimate for term the	$\hat{P}(X = \text{the}\|c) \approx 0.05$	$\hat{P}(U_{\text{the}} = 1\|c) \approx 1.0$

Multinomial Naïve Bayes for text

- ..

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k) P(X_1, X_2, .., X_N | Y=y_k)}{\sum_j P(Y=y_j) P(X_1, X_2, .., X_N | Y=y_j)}$$

$$= \frac{P(Y=y_k) \Pi_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j) \Pi_i P(X_i | Y=y_j)}$$

$$Y \leftarrow argmax_{y_k} P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

Naïve Bayes Classifier for Text

- Given the training data what are the parameters to be estimated?

$P(Y)$ $P(X|Y_1)$ $P(X|Y_2)$

Diabetes : 0.8
Hepatitis : 0.2

the: 0.001
diabetic : 0.02
blood : 0.0015
sugar : 0.02
weight : 0.018
...

the: 0.001
diabetic : 0.0001
water : 0.0118
fever : 0.01
weight : 0.008
...

13.2 Naive Bayes text classification

MULTINOMIAL NAIVE BAYES

The first supervised learning method we introduce is the *multinomial Naive Bayes* or *multinomial NB* model, a probabilistic learning method. The probability of a document d being in class c is computed as

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c) \tag{13.2}$$

where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c.[1] We interpret $P(t_k|c)$ as a measure of how much evidence t_k contributes that c is the correct class. $P(c)$ is the prior probability of a document occurring in class c. If a document's terms do not provide clear evidence for one class versus another, we choose the one that has a higher prior probability. $\langle t_1, t_2, \ldots, t_{n_d} \rangle$ are the tokens in d that are part of the vocabulary we use for classification and n_d is the number of such tokens in d. For example, $\langle t_1, t_2, \ldots, t_{n_d} \rangle$ for the one-sentence document *Beijing and Taipei join the WTO* might be ⟨Beijing, Taipei, join, WTO⟩, with $n_d = 4$, if we treat the terms and and the as stop words.

In text classification, our goal is to find the *best* class for the document. The best class in NB classification is the most likely or *maximum a posteriori* (MAP) class c_{map}:

MAXIMUM A POSTERIORI CLASS

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c). \tag{13.3}$$

http://nlp.stanford.edu/IR-book/pdf/13bayes.pdf

We write $\hat{P}$ for P because we do not know the true values of the parameters $P(c)$ and $P(t_k|c)$, but estimate them from the training set as we will see in a moment.

In Equation (13.3), many conditional probabilities are multiplied, one for each position $1 \le k \le n_d$. This can result in a floating point underflow. It is therefore better to perform the computation by adding logarithms of probabilities instead of multiplying probabilities. The class with the highest log probability score is still the most probable; $\log(xy) = \log(x) + \log(y)$ and the logarithm function is monotonic. Hence, the maximization that is

- ..

ADD-ONE SMOOTHING

To eliminate zeros, we use *add-one* or *Laplace smoothing*, which simply adds one to each count (cf. Section 11.3.2):

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'}) + B'} \tag{13.7}$$

where $B = |V|$ is the number of terms in the vocabulary. Add-one smoothing can be interpreted as a uniform prior (each term occurs once for each class) that is then updated as evidence from the training data comes in. Note that this is a prior probability for the occurrence of a *term* as opposed to the prior probability of a *class* which we estimate in Equation (13.5) on the document level.

NB Multinomial example

D1
D2
D
3

http://nlp.stanford.edu/IR-book/pdf/13bayes.pdf

► **Table 13.1** Data for parameter estimation examples.

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

► **Table 13.2** Training and test times for NB.

mode	time complexity
training	$\Theta(\mathbb{D}L_{ave} + \mathbb{C}\mathbb{V})$
testing	$\Theta(L_a + \mathbb{C}M_a) = \Theta(\mathbb{C}M_a)$

We have now introduced all the elements we need for training and applying an NB classifier. The complete algorithm is described in Figure 13.2.

Example 13.1: For the example in Table 13.1, the multinomial parameters we need to classify the test document are the priors $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ and the following conditional probabilities:

$$\begin{aligned}
\hat{P}(\text{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\
\hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) &= (0+1)/(8+6) = 1/14 \\
\hat{P}(\text{Chinese}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\
\hat{P}(\text{Tokyo}|\overline{c}) = \hat{P}(\text{Japan}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

The denominators are $(8+6)$ and $(3+6)$ because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B in Equation (13.7) is 6 as the vocabulary consists of six terms.

We then get:

$$\begin{aligned}
\hat{P}(c|d_5) &\propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003. \\
\hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.
\end{aligned}$$

Thus, the classifier assigns the test document to $c = China$. The reason for this classification decision is that the three occurrences of the positive indicator Chinese in d_5 outweigh the occurrences of the two negative indicators Japan and Tokyo.

Multinomial Naive Bayes with Laplace smoothing. Here the vocabulary over two classes consists of 6 unique words. The length of the not class (China = no) is 3 (one doc with just three tokens) so the default probability for a word in the model but not in the Not-Class is 1/(3+6)

NB Multinomial example

► **Table 13.1** Data for parameter estimation examples.

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

► **Table 13.2** Training and test times for NB.

mode	time complexity
training	$\Theta(\|\mathbb{D}\|L_{ave} + \|\mathbb{C}\||V|)$
testing	$\Theta(L_a + \|\mathbb{C}\|M_a) = \Theta(\|\mathbb{C}\|M_a)$

D1,1, Chinese Beijing, Chinese
D2,1, Chinese Chinese, Shanghai
D3,1, Chinese, Macao
D4,1, Tokyo Japan,Chinese
D5,1, Chinese Chinese, Chinese Tokyo Japar

We have now introduced all the elements we need for training and applying an NB classifier. The complete algorithm is described in Figure 13.2.

Example 13.1: For the example in Table 13.1, the multinomial parameters we need to classify the test document are the priors $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ and the following conditional probabilities:

$$\begin{aligned}
\hat{P}(\text{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\
\hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) &= (0+1)/(8+6) = 1/14 \\
\hat{P}(\text{Chinese}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\
\hat{P}(\text{Tokyo}|\overline{c}) = \hat{P}(\text{Japan}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

The denominators are $(8+6)$ and $(3+6)$ because the lengths of $text_c$ and $text_{\overline{c}}$ are 8 and 3, respectively, and because the constant B in Equation (13.7) is 6 as the vocabulary consists of six terms.

We then get:

$$\begin{aligned}
\hat{P}(c|d_5) &\propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003. \\
\hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.
\end{aligned}$$

Thus, the classifier assigns the test document to $c = China$. The reason for this classification decision is that the three occurrences of the positive indicator Chinese in d_5 outweigh the occurrences of the two negative indicators Japan and Tokyo.

Model file can be a csv with three columns

► **Table 13.1** Data for parameter estimation examples.

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Word	Pr(Word|Class)	Pr(Word| NotClass)
CLASSPRIORs	¾ = 0.75	¼ = 0.25
Tokyo	0+1/8+6 =1/14	1+1/3+6 = 2/9
Chinese	5+1/8+6 = 6/14	1+1/3+6 = 2/9
…..		

Model file can be a csv with three columns

► Table 13.1 Data for parameter estimation examples.

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Mapper

For document emit 1
Wordstats[“priors”] = [1,0]
Wordstats[“Chinese”] = [1,0]
Wordstats[“Beijing”] = [1,0]
Wordstats[“Chinese”] = [1,0]
*vocab, Chinese
*vocab, Bejing
*vocab, Chinese

Word	Pr(Word|Class)	Pr(W NotC
		¼ =
		+1/
		+1/

#In python use a dictionary/hash as following to store the model or intermediate versions of the model
modelStats = {}
modleStats[“priors”] = [1,0]
modelStats[“Chinese”]= [5, 1]
modelStats[“Tokyo”]= [0, 1]

Another Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

Learning Phase

P(Outlook=*o* | Play=*b*)

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

P(Temperature=*t* | Play=*b*)

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

P(Humidity=*h* | Play=*b*)

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

P(Wind=*w* | Play=*b*)

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

Example

- Test Phase
  - Given a new instance,

    $\mathbf{x}'$ =(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)
  - Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=S*unny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

  - MAP rule

P(*Yes*|$\mathbf{x}'$): [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

P(*No*|$\mathbf{x}'$): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|$\mathbf{x}'$) < P(*No*|$\mathbf{x}'$), we label $\mathbf{x}'$ to be "*No*".

Relevant Issues

- Violation of Independence Assumption
  - For many real world tasks, $P(X_1,\cdots,X_n|C) \neq P(X_1|C)\cdots P(X_n|C)$
  - Nevertheless, naïve Bayes works surprisingly well anyway!
- Zero conditional probability Problem
  - If no example contains the attribute value $X_j = a_{jk}$, $\hat{P}(X_j = a_{jk}|C = c_i) = 0$
  - In this circumstance, $\hat{P}(x_1|c_i)\cdots\hat{P}(a_{jk}|c_i)\cdots\hat{P}(x_n|c_i) = 0$ during test
  - For a remedy, conditional probabilities estimated with **Laplace smoothing**

$$\hat{P}(X_j = a_{jk}|C = c_i) = \frac{n_c + mp}{n + m}$$

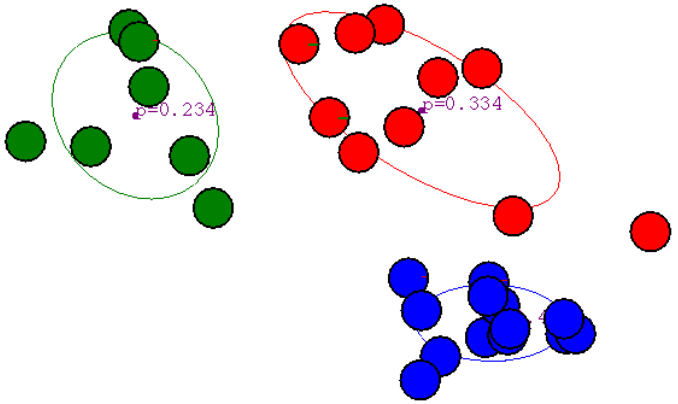
n_c : number of training examples for which $X_j = a_{jk}$ and $C = c_i$

n : number of training examples for which $C = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of X_j)

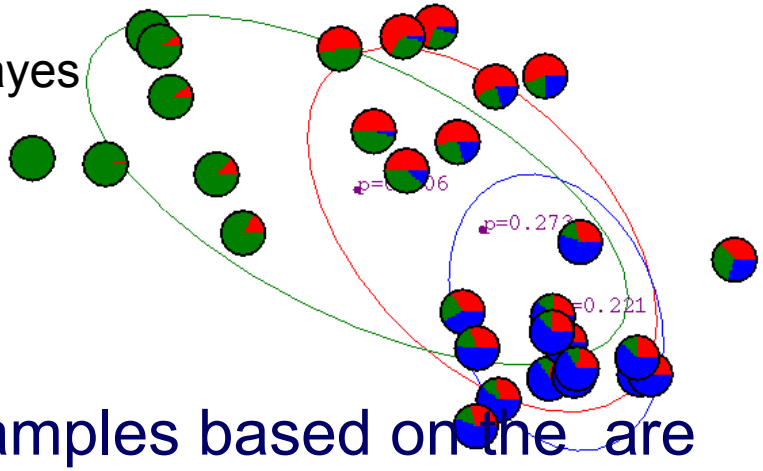
m : weight to prior (number of "virtual" examples, $m \geq 1$)

Flat Clustering: Hard versus Soft



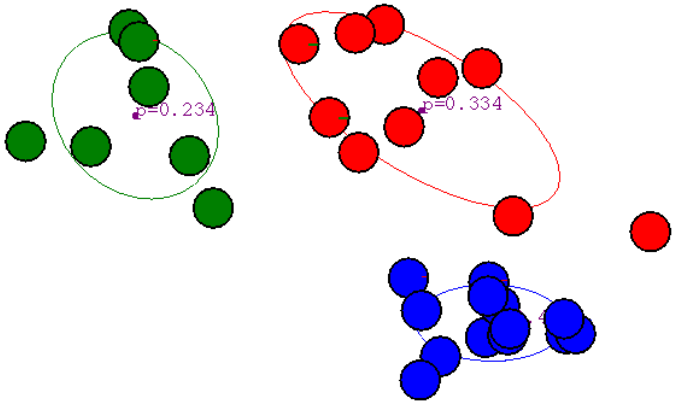
- **Hard Clustering**
  - Hard assignment; Kmeans (EM-like)
- **Soft Clustering**
  - This set of assignment probabilities (aka responsibilities) defines a soft clustering.
  - Model-based Clustering using EM and Maximum Likelihood

Naïve Bayes



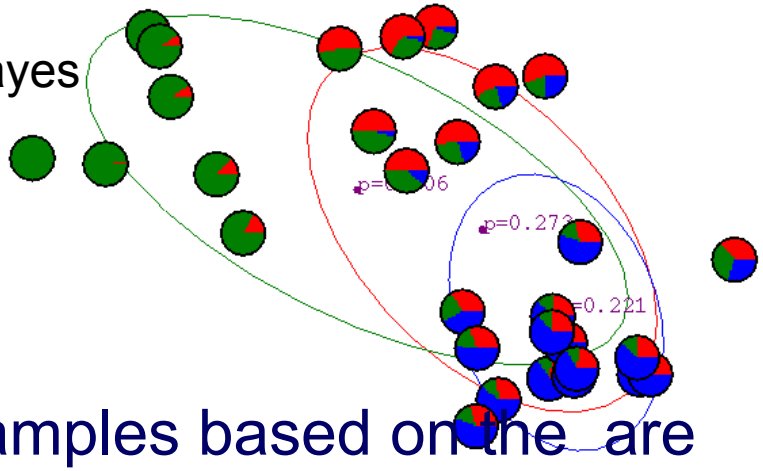
  - Weighted EM-like
    - EM Centroids are weighted examples based on the are the cluster/class probabilities assignment probabilities (aka responsibilites)

Flat Clustering: Hard versus Soft



- **Hard Clustering**
  - Hard assignment; Kmeans (EM-like)
- **Soft Clustering**
  - This set of assignment probabilities (aka responsibilities) defines a soft clustering.
  - Model-based Clustering using EM and Maximum Likelihood

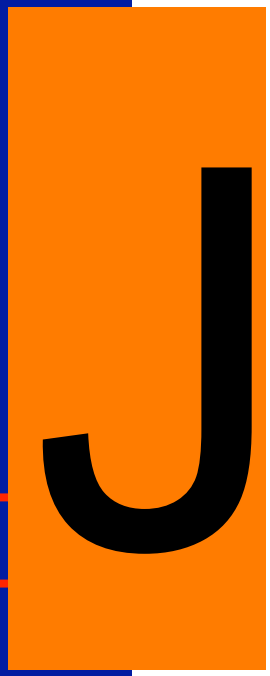
Naïve Bayes



  - Weighted EM-like
    - EM Centroids are weighted examples based on the are the cluster/class probabilities assignment probabilities (aka responsibilites)

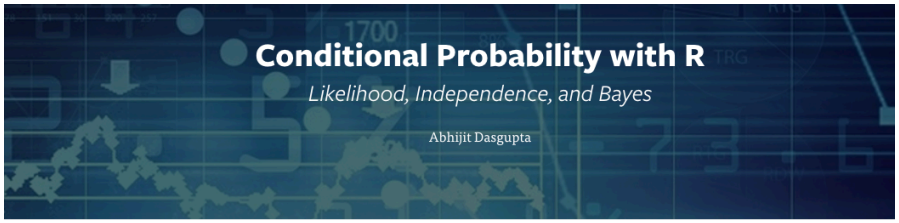
Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Probability Theory Primer Notebook

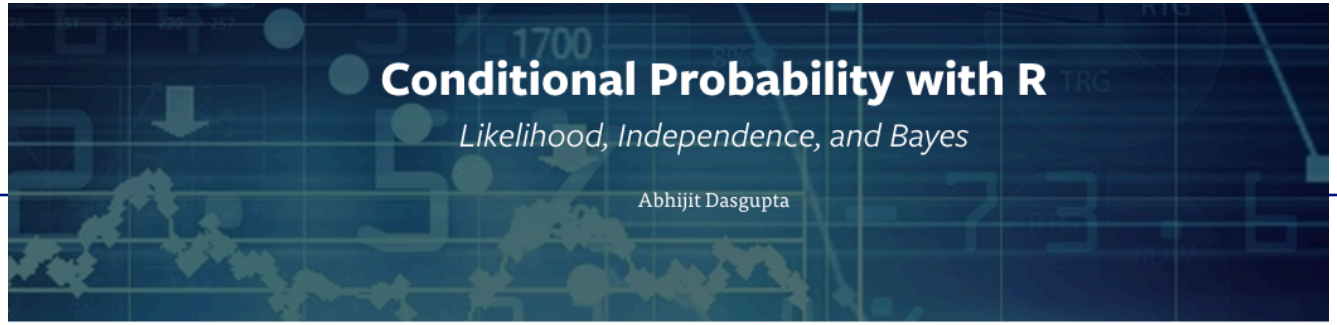
- http://blog.districtdatalabs.com/conditional-probability-with-r
- https://www.dropbox.com/s/non3tju37stqynl/District%20Data%20Labs%20-%20Conditional%20Probability%20with%20R.htm?dl=0



In addition to regular probability, we often want to figure out how probability is affected by observing some event. For example, the NFL season is rife with possibilities. From the beginning of each season, fans start trying to figure out how likely it is that their favorite team will make the playoffs. After every game the team plays, these probabilities change based on whether they won or lost. This post won't speak to how these probabilities are updated. That's the subject for a future post on Bayesian statistics. What we *will* explore is the concept of **conditional probability**, which is the probability of seeing some event knowing that some other event has actually occurred.

Some more examples of where we might encounter such conditional probabilities:

- Inveterate bridge players like my dad would keep track of cards as they got exposed in the pile, for that (and the bids) provided information about the likelihoods of what hand each player had. Such card counting and conditional probabilities (what's the likelihood of each hand, given what I have seen) is one of the (frowned upon) strategies for trying to beat the casinos in blackjack and poker (see the movie 21 for a Hollywood version of real-life card counting in casinos).
- When we go to the doctor to test for a disease (say tuberculosis or HIV or even, more commonly, strep throat and flu), we get a yes or no answer. However, no test is perfect. A positive test still means we might not have the disease, and testing negative might mean we have it, though hopefully with very little likelihood. For us, the important thing to know is, if we



In addition to regular probability, we often want to figure out how probability is affected by observing some event. For example, the NFL season is rife with possibilities. From the beginning of each season, fans start trying to figure out how likely it is that their favorite team will make the playoffs. After every game the team plays, these probabilities change based on whether they won or lost. This post won't speak to how these probabilities are updated. That's the subject for a future post on Bayesian statistics. What we *will* explore is the concept of **conditional probability**, which is the probability of seeing some event knowing that some other event has actually occurred.

Some more examples of where we might encounter such conditional probabilities:

- Inveterate bridge players like my dad would keep track of cards as they got exposed in the pile, for that (and the bids) provided information about the likelihoods of what hand each player had. Such card counting and conditional probabilities (what's the likelihood of each hand, given what I have seen) is one of the (frowned upon) strategies for trying to beat the casinos in blackjack and poker (see the movie 21 for a Hollywood version of real-life card counting in casinos).
- When we go to the doctor to test for a disease (say tuberculosis or HIV or even, more commonly, strep throat and flu), we get a yes or no answer. However, no test is perfect. A positive test still means we might not have the disease, and testing negative might mean we have it, though hopefully with very little likelihood. For us, the important thing to know is, if we tested positive (an observed event), what is the chance that we truly have the disease (an unobserved event).
- Weather forecasting is based on conditional probabilities. When the forecast says that there is a 30% chance of rain, that probability is based on all the information that the meteorologists know up until that point. It's not just a roll of the dice (though sometimes, it feels that way).

Statistically Independent Events

We can compare the probability of an event (A) and how it changes if we know that another event (B) has happened. How does the chance of catching flu (A) change if you're vaccinated (B)? How does a football team's chance of going to the playoffs (A) change if the quarterback is injured (B)? In both these cases, we think those chances will change.

But will the chance of the Pittsburgh Steelers beating New England Patriots (sacrilegious to some, I know) in the 4 pm game depend on the Seattle Seahawks beating the San Francisco 49ers (*caveat*: I'm from Seattle) during the same time? We think (and hope) not.

When knowledge of one event does not change the probability of another event happening, the two events are called **statistically independent**. We see a lot of things that are independent in this sense. Successive tosses of a coin are independent, or so we believe. So are successive dice rolls and slot machine plays.

Statistical independence has some mathematical consequences. It implies that

```
P(A|B) = P(A)
```

which directly implies, from the definition, that

```
P(A and B) = P(A)P(B)
```

This means that we can compute the probability of two independent events happening together by merely multiplying the individual probabilities.

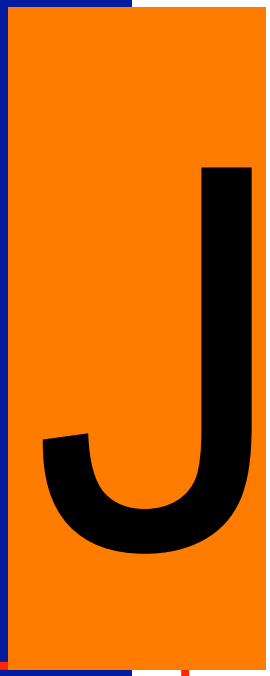
Caution: You'll often find probabilities of joint events like this computed as the product of the individual events. However, this is *only* true if the assumption of statistical independence is valid. Often times, it is not, and so you must be careful interpreting such computations.

Let's do a little experiment in R. We'll toss two fair dice, just as we did in an earlier post, and see if the results of the two dice are independent. We first roll the dice 100,000 times, and then compute the joint distribution of the results of the rolls from the two dice.

```
dice <- function(no_of_rolls=1){
  x <- sample(1:6, size=no_of_rolls, replace=TRUE)
  y <- sample(1:6, size=no_of_rolls, replace=TRUE)
  return(cbind(x,y))
}
```

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**

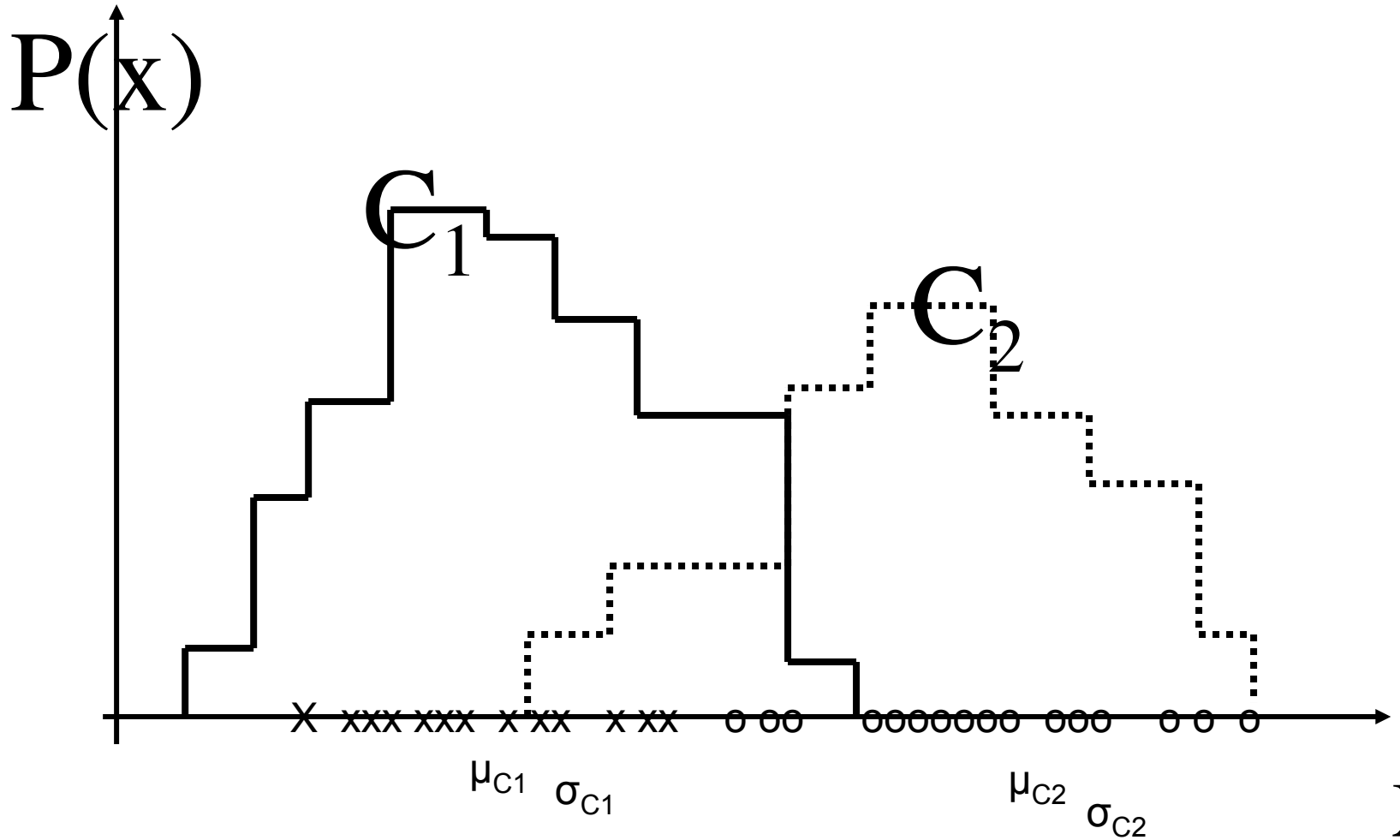


Naive Bayes for Continuous Inputs

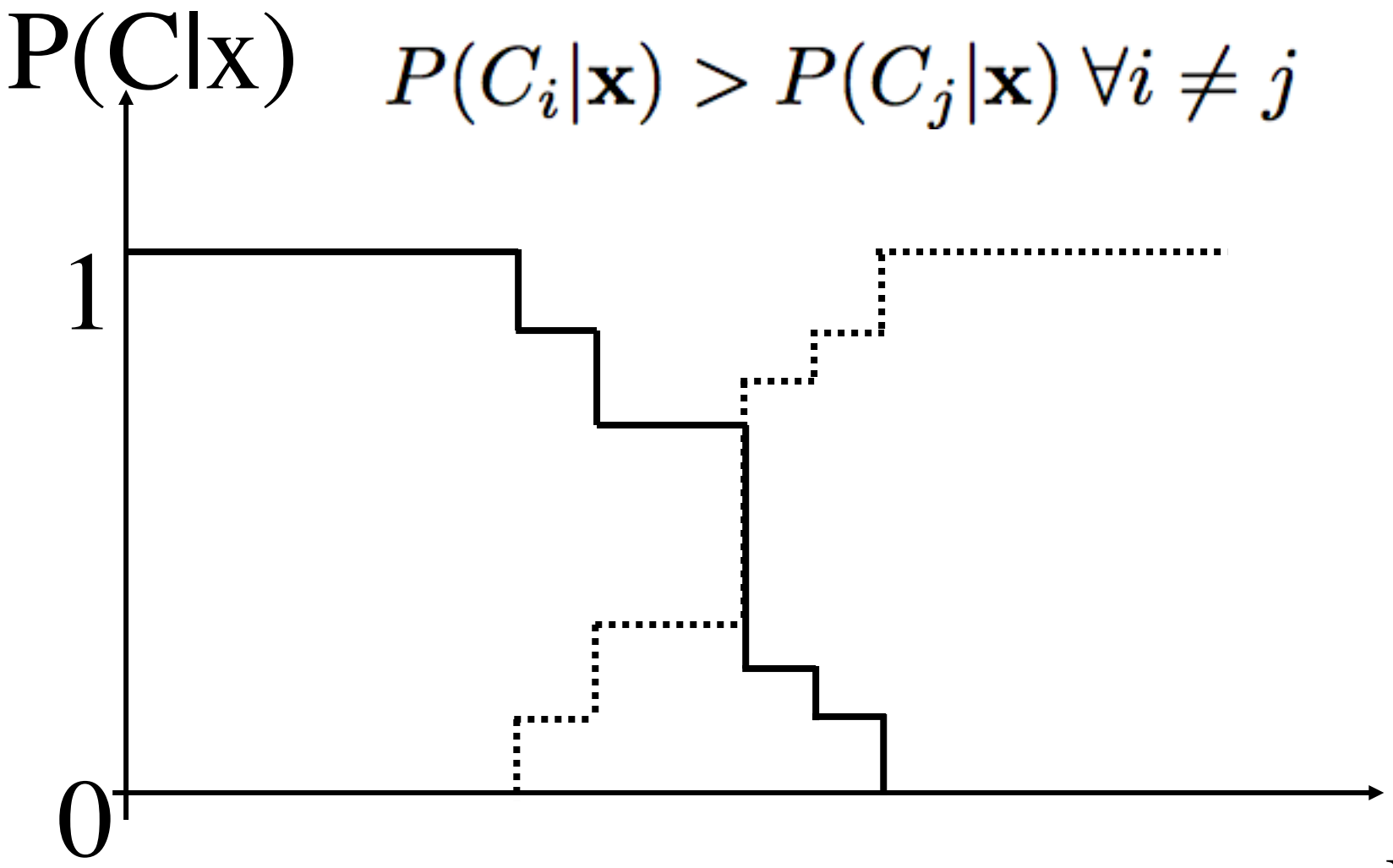
- When the Xi are continuous we must choose some other way to represent the distributions
- P(Xi|Y)=Gaussian(Mu, Sigma).
- One common approach is to assume that for each possible discrete value yk of Y, the distribution of each continuous Xi is Gaussian, and is defined by a mean and standard deviation specific to Xi and yk.
- In order to train such a Naïve Bayes classifier we must therefore estimate the mean and standard deviation ians:

$$\mu_{ik} = E[X_i | Y = y_k]$$
$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 | Y = y_k]$$

Gaussian Naïve Bayes



Posterior Probability



Slide by Stephen Marsland

MLE-based Estimates

Again, we can use either maximum likelihood estimates (MLE) or maximum a posteriori (MAP) estimates for these parameters. The maximum likelihood estimator for μ_{ik} is

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k) \tag{13}$$

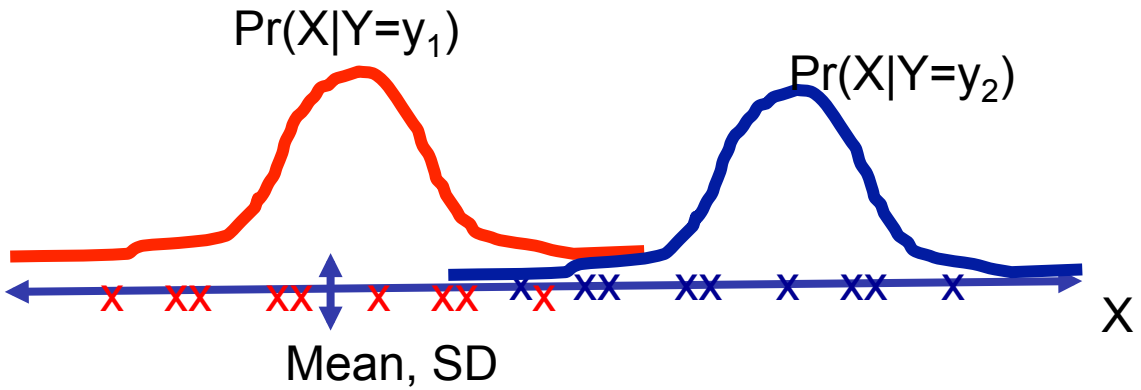
where the superscript j refers to the jth training example, and where $\delta(Y = y_k)$ is 1 if $Y = y_k$ and 0 otherwise. Note the role of δ here is to select only those training examples for which $Y = y_k$.

The maximum likelihood estimator for σ_{ik}^2 is

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k) \tag{14}$$

Estimate μ, σ from data

$$\mu_{ik} = E[X_i | Y = y_k]$$
$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 | Y = y_k]$$



Continuous Inputs: Pr(Y|X) ~ N(μ, σ)

If X is Normally distributed with mean μ and standard deviation σ, we write

$$\boxed{X \sim N(\mu, \sigma^2)}$$

μ and σ are the **parameters** of the distribution.

The probability density of the Normal distribution is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-(x-\mu)^2/2\sigma^2}$$

For the purposes of this course we do not need to use this expression. It is included here for future reference.

Naïve Bayes

- **Combine discrete input variables with continuous input variables?**
  - Naïve Bayes, Decision trees
- **Whereas these requires feature transformations**
  - Logistic regression: one hot-encoding

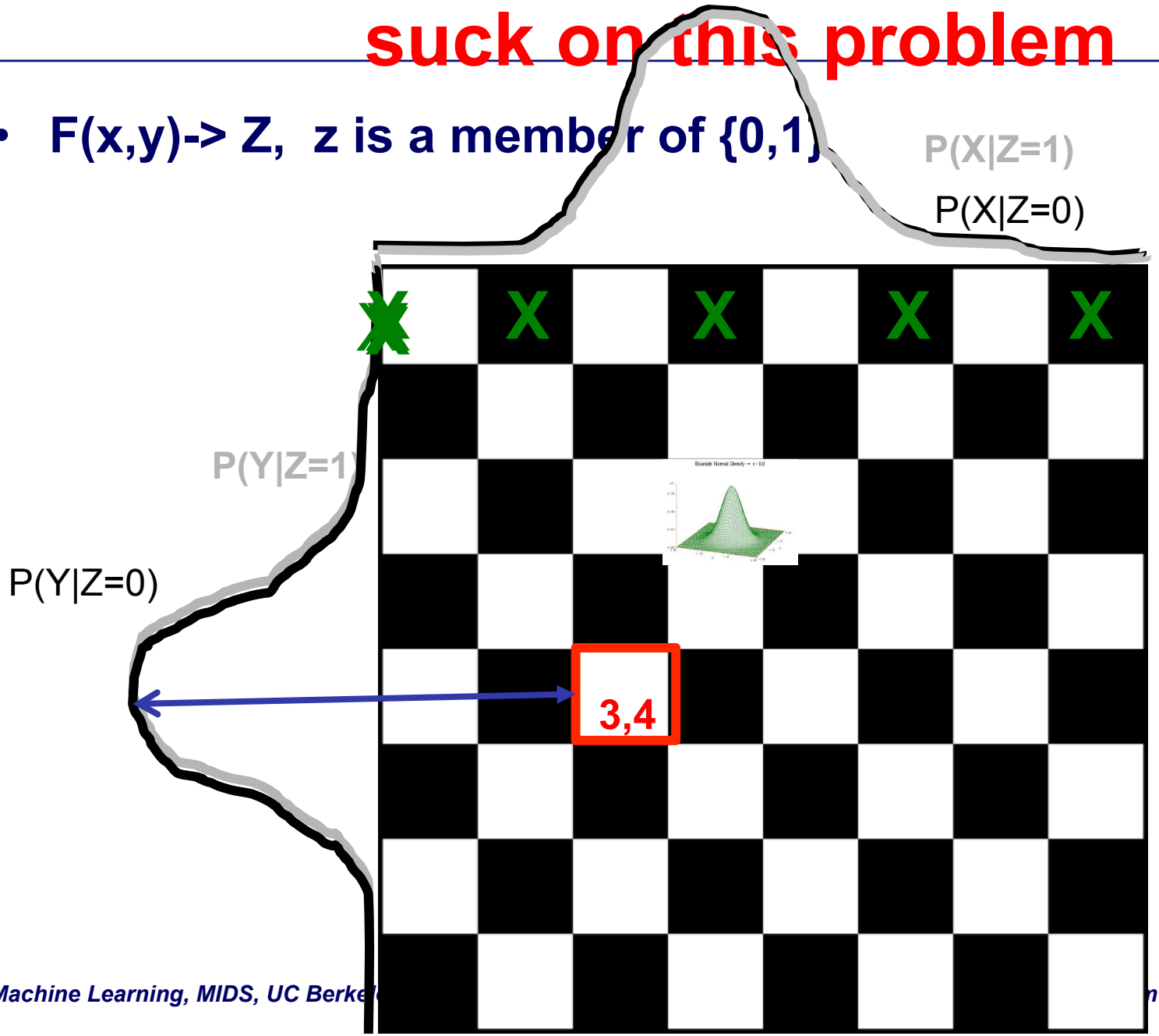
- **YES**

Continuous NB: Complexity

- **For each attribute Xi and each possible value yk of Y. Note there are $2nK$ of these parameters, all of which must be estimated independently.**
  - K classes and n continuous input variables

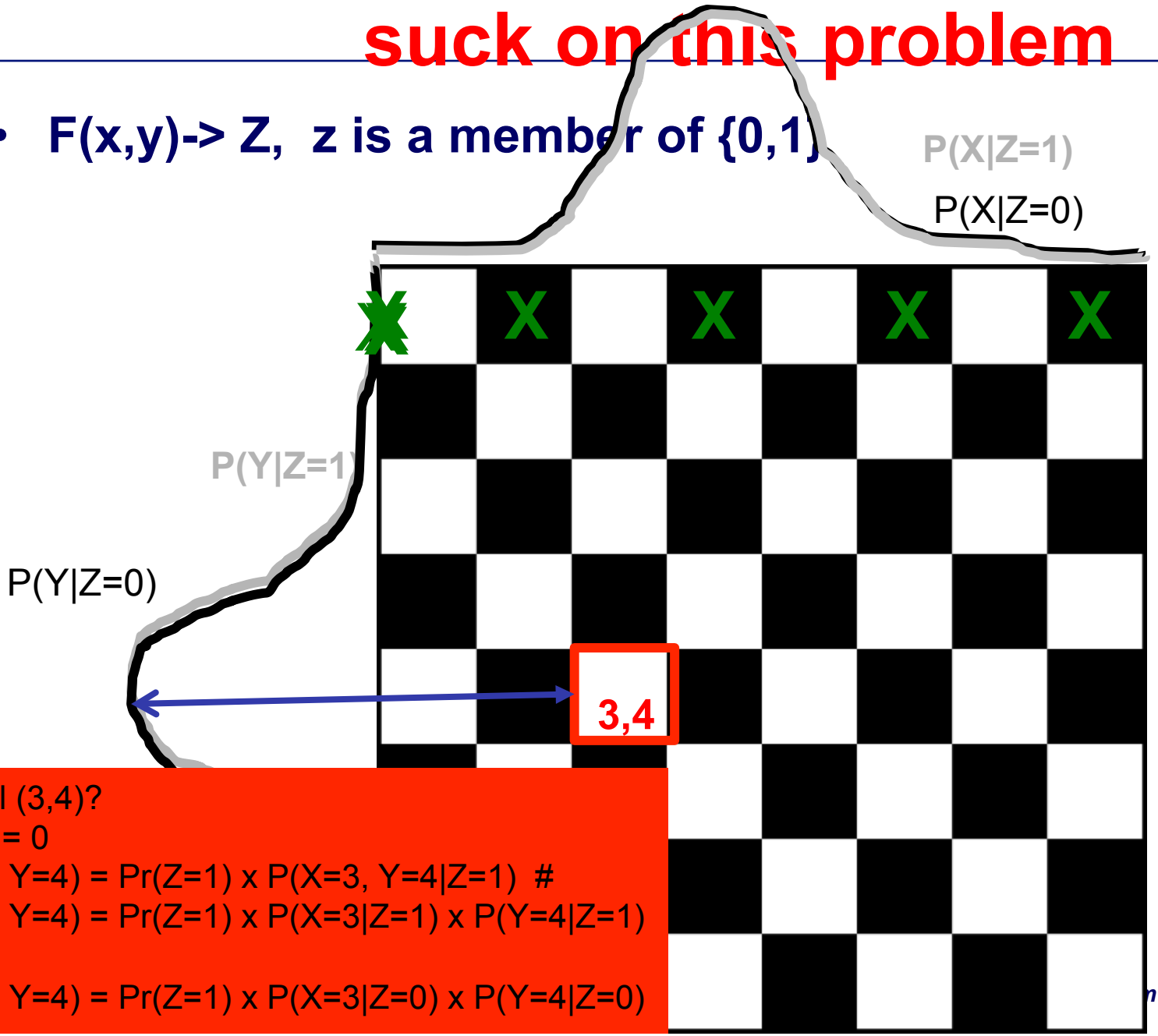
Continuous input Naïve Bayes will suck on this problem

- F(x,y)-> Z, z is a member of {0,1}



Continuous input Naïve Bayes will suck on this problem

- F(x,y)-> Z, z is a member of {0,1}

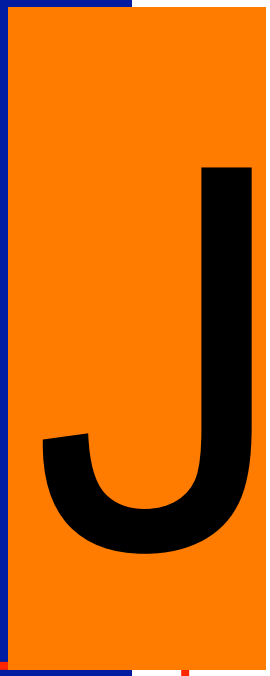


Classify cell (3,4)?
Z =1 and Z = 0
P(Z=1|X=3, Y=4) = Pr(Z=1) x P(X=3, Y=4|Z=1) #
P(Z=1|X=3, Y=4) = Pr(Z=1) x P(X=3|Z=1) x P(Y=4|Z=1)

P(Z=0|X=3, Y=4) = Pr(Z=1) x P(X=3|Z=0) x P(Y=4|Z=0)

Probability Basics → Naïve Bayes Models

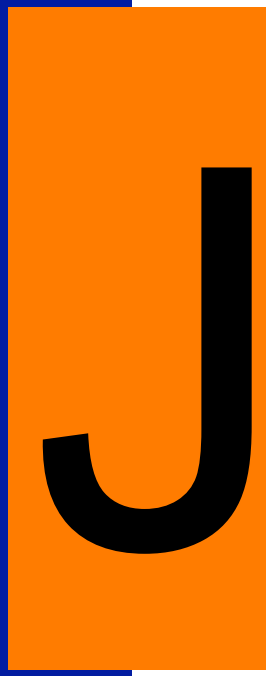
- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



- **Need to insert slides here for**
  - Discrete input variables (2 flavors: Bernoulli, multinomial)

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**



Case Study: Spam detector in Naïve Bayes

- Enron Spam Dataset
- Word count in MapReduce terms
- From word counts to a mulitnomial naïve Bayes classifier

Homework HW1: Data set:Enron SPAM Mail

- The Enron Corpus is a large database of over 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission during its investigation after the company's collapse.
- The Enron data was originally collected at Enron Corporation headquarters in Houston during two weeks in May 2002 by Joe Bartling,[3] a litigation support and data analysis contractor working for Aspen Systems, now Lockheed Martin, whom the Federal Energy Regulatory Commission (FERC) had hired to preserve and collect the vast amounts of data in the wake of the Enron Bankruptcy in December 2001.

ENRON SPAM Data

- **This SPAM/HAM dataset for HW1 contains 100 records from the Enron SPAM/HAM corpus.**
- **There are about 93,000 emails in the original SPAM/HAM corpus. There are several versions of the SPAM/HAM corpus.**
- **Other Enron-Spam datasets are available from**
  - http://www.iit.demokritos.gr/skel/i-config/ and
  - http://www.aueb.gr/users/ion/publications.html in both raw and pre-processed form.

Enron Spam Data: Examples

Date and employee\tSPAM Tag \t Subject \t Email body

	A	B	C	D	E	F	G	H	I	J	K
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA							
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.							
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the							
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks, project apollo and beyond will conduct its final sap implementatio								
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs generating like mid-year documenting business unit performance							
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get some coffee with you, but remember that annoying project that m							
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high island larger block a-1 # 2 commenced on saturday at 2:00 p.m. at a							
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution of the central maine sos deal! this is another great example of wha							
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email address was obtained from a purchased list, reference # 2020 mid = 330							
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! space saving computer in a keyboard eliminate that big box comp							
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe hendricks (mrs) of rsa. i am writing this letter to you with the hop							
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine. i have talked to shirley and have directions. thanks, bob vince j							
13	0003.1999-12-14.farmer	0	calpine daily gas nomination	-calpine daily gas nomination 1. doc							
14	0003.2000-01-17.beck	0	re: additional responsibility	congratulations on this additional responsibility! i will be more than happy to help support your ne							
15	0003.2001-02-08.kitchen	0	re: key hr issues going forward	all is under control: a-we've set up a " work-out " group under cindy skinner and will be producing t							
16	0003.2003-12-18.GP	1	fw: account over due wfxu ppmfztdtet	eliminate your credit card debt without bankruptcy! tired of making minimum payments and barely							
17	0003.2004-08-01.BG	1	whats new in summer? bawled	carolyn regretful watchfully procrustes godly summer 2004 was too hot for the software manufactu							
18	0004.1999-12-10.kaminski	0	research group move to the 19 th floor	hello all: in case any of you feel energetic, " the boxes are here ". they are located at 2963 b (micha							
19	0004.1999-12-14.farmer	0	re: issue	fyi-see note below-already done. stella ----------forwarded by stella l morris/hou/ect on 12							
20	0004.2001-04-02.williams	0	enrononline desk to desk id and password	bill, the epmi-st-wbom book has been set up as an internal counterparty for desk-to-desk trading o							
21	0004.2001-06-12.SA_and_HP	1	spend too much on your phone bill? 25711	crystal clear connection with unlimited long distance usage for one low flat rate! now try it for fre							
22	0004.2004-08-01.BG	1	NA	h$ ello dea 54 r home owner, we have beetcn notiffiyved that your morayt " goage r [ate is fixed a							
23	0005.1999-12-12.kaminski	0	christmas baskets	the christmas baskets have been ordered. we have ordered several baskets. individual earth-sat fr							
24	0005.1999-12-14.farmer	0	meter 7268 nov allocation	fyi. ----------forwarded by lauri a allen/hou/ect on 12/14/99 12:17 pm----------							
25	0005.2000-06-06.lokay	0	transportation to resort	please be informed, a mini-bus has been reserved for your convenience in transporting you to the s							

Examples 1-4 (not SPAM) and 9 (SPAM)

1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. your vendor selection team will receive an update and even more information later in the week. on the lunch & learn for energy operations, the audience and focus will be your group. …….[TRUNCATED]
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks, project apollo and beyond will conduct its final sap implementation) [TRUNCATED] NOTE: No Body text for this email	
9				oo thank you, your email address was obtained from a purchased list, reference # 2020 mid = 3300. if you wish to unsubscribe from this list, please click here and enter your name into the remove box. if you have previously unsubscribed and are still receiving this message, you may email our abuse control center, or call 1-888-763-2497, or write us at: nospam, 6484

Simple Spark Apps: *WordCount*

WordCount Example 3

Definition:

count how often each word appears in a collection of text documents

This simple program provides a good test case for parallel processing, since it:

- requires a minimal amount of code
- demonstrates use of both symbolic and numeric values
- isn't many steps away from search indexing
- serves as a "Hello World" for Big Data apps

```
void map (String doc_id, String text):
  for each word w in segment(text):
    emit(w, "1");


void reduce (String word, Iterator group):
  int count = 0;

  for each pc in group:
    count += Int(pc);

  emit(word, String(count));
```

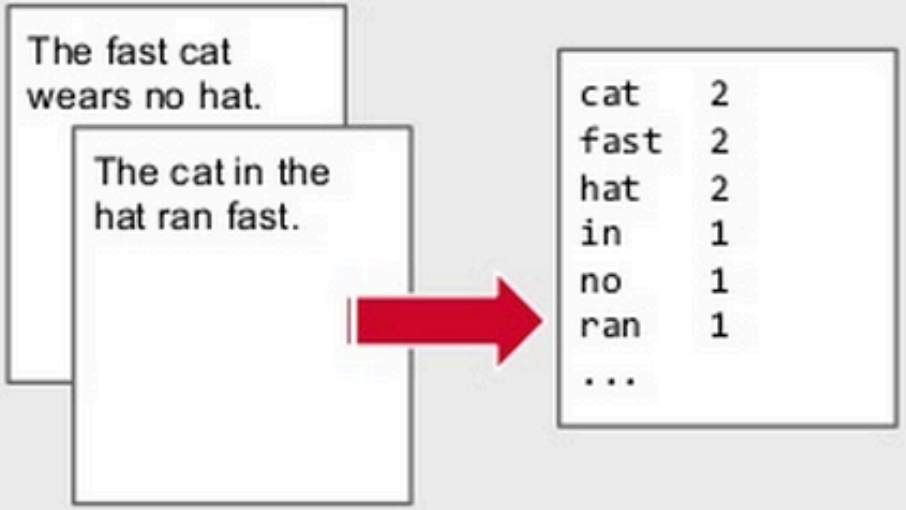
A distributed computing framework that can run WordCount **efficiently in parallel at scale** can likely handle much larger and more interesting compute problems

Word Count in Map-Reduce

```
def map(key, value):
    [redacted]
        emit(word, 1)

def reduce(key, values):
    [redacted]
        count += val
    emit(key, count)

# emit is a function that performs distributed I/O
```



Each document is passed to a mapper, which does the tokenization. The output of the mapper is reduced by key (word) and then counted.

What is the data flow for word count?

Word Count broken down

```
1 hello hi hi hallo
2 bonjour hola hi ciao
3 nihao konnichiwa ola
4 hola nihao hello
```

[hello, hi, hi, hallo, bonjour, hola, hi, ciao,
ao, konnichiwa, ola, hola, nihao, hello]

Mapper

(hello,1),
(hi,1),
(hi,1),
(hallo,1),
(bonjour,1),
(hola,1),
(hi,1),
(ciao,1),
(nihao,1),
(konnichiwa,1),
(ola,1),
(hola,1),
(nihao,1),
(hello,1)

reduce ByKey

(u'ciao', 1)
(u'bonjour', 1)
(u'nihao', 2)
(u'hola', 2)
(u'konnichiwa', 1)
(u'hallo', 1)
(u'hi', 3)
(u'hello', 2)
(u'ola', 1)

SPAM Data Word Count : HW1.2

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! s
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. You… [TRUNCATED BODY]**

WordCount example reads text files and counts how often words occur. The input is text files and the output is text files, each line of which contains a word and the count of how often it occured, separated by a tab.

MAPPER: Each mapper takes a line as input and breaks it into words. It then emits a key/value pair of the word and 1.

REDUCER: Each reducer sums the counts for each word and emits a single key/value with the word and sum.

SPAM Data Word Count : HW1.2

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!!
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below, is an update on the leadership pilot. You… [TRUNCATED BODY]**

As an optimization, the reducer is also used as a combiner on the map outputs. This reduces the amount of data sent across the network by combining each word into a single record. [See Lecture 3]

WordCount example … ccur. The input is text files and the outp… word and the count of how often it …

MAPPER: Each map… rds. It then emits a key/value pair of the word and 1.

REDUCER: Each reducer sums the counts for each word and emits a single key/value with the word and sum.

Word Count : HW1.2: Mapper

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D	E	F
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA		
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.		
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s		
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,			
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs		
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get		
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is		
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution		
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres		
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! s		
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe		
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine		

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. You… [TRUNCATED BODY]**

For each record

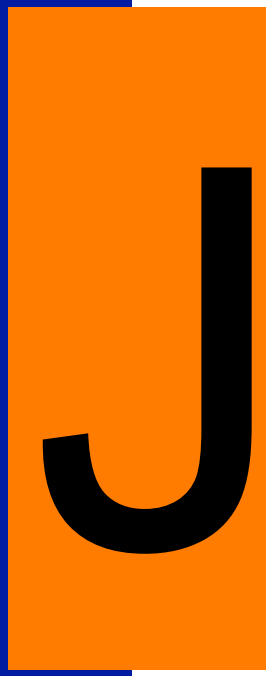
- extract field 3 and 4 (Subject and Email Body) and split into tokens (words/numbe
- Out a list of token-count pairs

Output from Mapper

Mapper Output	Key=Word	Value=Count
Line 1	leadership	1
2	development	1
3	you	5

Probability Basics → Naïve Bayes Models

- **Probability Basics**
  - Probability Axioms
  - Conditional probabilities
  - Product Rule, Chain Rule, Bayes Rule
- **Bayes Nets And Naïve Bayes**
  - Learning
  - Independence
  - Conditional independence
  - Naïve Bayes derivation (discrete case plus smoothing)
  - Notebook with examples
- **Naïve Bayes**
  - Continuous input variables
  - Discrete input variables (2 flavors: Bernoulli, multinomial)
- **Case Study: Spam detector in Naïve Bayes**
  - Multinomial Naïve Bayes in Hadoop



Naïve Bayes in Hadoop

- **Naïve Bayes in Hadoop**
- **Passing a model to Hadoop**
- **Zero reducer jobs:**
  - How to deal with them and why?

Case Study: Spam detector in Naïve Bayes

- Enron Spam Dataset
- Word count in MapReduce terms
- From word counts to a mulitnomial naïve Bayes classifier

Homework HW1: Data set:Enron SPAM Mail

- The Enron Corpus is a large database of over 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission during its investigation after the company's collapse.
- The Enron data was originally collected at Enron Corporation headquarters in Houston during two weeks in May 2002 by Joe Bartling,[3] a litigation support and data analysis contractor working for Aspen Systems, now Lockheed Martin, whom the Federal Energy Regulatory Commission (FERC) had hired to preserve and collect the vast amounts of data in the wake of the Enron Bankruptcy in December 2001.

ENRON SPAM Data

- **This SPAM/HAM dataset for HW1 contains 100 records from the Enron SPAM/HAM corpus.**
- **There are about 93,000 emails in the original SPAM/HAM corpus. There are several versions of the SPAM/HAM corpus.**
- **Other Enron-Spam datasets are available from**
  - http://www.iit.demokritos.gr/skel/i-config/ and
  - http://www.aueb.gr/users/ion/publications.html in both raw and pre-processed form.

Enron Spam Data: Examples

Date and employee\tSPAM Tag \t Subject \t Email body

	A	B	C	D	E	F	G	H	I	J	K
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA							
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.							
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the							
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks, project apollo and beyond will conduct its final sap implementatio								
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs generating like mid-year documenting business unit performance							
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get some coffee with you, but remember that annoying project that m							
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high island larger block a-1 # 2 commenced on saturday at 2:00 p.m. at a							
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution of the central maine sos deal! this is another great example of wha							
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email address was obtained from a purchased list, reference # 2020 mid = 330							
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! space saving computer in a keyboard eliminate that big box comp							
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe hendricks (mrs) of rsa. i am writing this letter to you with the hop							
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine. i have talked to shirley and have directions. thanks, bob vince j							
13	0003.1999-12-14.farmer	0	calpine daily gas nomination	-calpine daily gas nomination 1. doc							
14	0003.2000-01-17.beck	0	re: additional responsibility	congratulations on this additional responsibility! i will be more than happy to help support your ne							
15	0003.2001-02-08.kitchen	0	re: key hr issues going forward	all is under control: a-we've set up a " work-out " group under cindy skinner and will be producing t							
16	0003.2003-12-18.GP	1	fw: account over due wfxu ppmfztdtet	eliminate your credit card debt without bankruptcy! tired of making minimum payments and barely							
17	0003.2004-08-01.BG	1	whats new in summer? bawled	carolyn regretful watchfully procrustes godly summer 2004 was too hot for the software manufactu							
18	0004.1999-12-10.kaminski	0	research group move to the 19 th floor	hello all: in case any of you feel energetic, " the boxes are here ". they are located at 2963 b (micha							
19	0004.1999-12-14.farmer	0	re: issue	fyi-see note below-already done. stella ----------forwarded by stella l morris/hou/ect on 12							
20	0004.2001-04-02.williams	0	enrononline desk to desk id and password	bill, the epmi-st-wbom book has been set up as an internal counterparty for desk-to-desk trading o							
21	0004.2001-06-12.SA_and_HP	1	spend too much on your phone bill? 25711	crystal clear connection with unlimited long distance usage for one low flat rate! now try it for fre							
22	0004.2004-08-01.BG	1	NA	h$ ello dea 54 r home owner, we have beetcn notiffiyved that your morayt " goage r [ate is fixed a							
23	0005.1999-12-12.kaminski	0	christmas baskets	the christmas baskets have been ordered. we have ordered several baskets. individual earth-sat fr							
24	0005.1999-12-14.farmer	0	meter 7268 nov allocation	fyi. ----------forwarded by lauri a allen/hou/ect on 12/14/99 12:17 pm----------							
25	0005.2000-06-06.lokay	0	transportation to resort	please be informed, a mini-bus has been reserved for your convenience in transporting you to the s							

Examples 1-4 (not SPAM) and 9 (SPAM)

1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. your vendor selection team will receive an update and even more information later in the week. on the lunch & learn for energy operations, the audience and focus will be your group. …….[TRUNCATED]
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks, project apollo and beyond will conduct its final sap implementation) [TRUNCATED] NOTE: No Body text for this email	
9				oo thank you, your email address was obtained from a purchased list, reference # 2020 mid = 3300. if you wish to unsubscribe from this list, please click here and enter your name into the remove box. if you have previously unsubscribed and are still receiving this message, you may email our abuse control center, or call 1-888-763-2497, or write us at: nospam, 6484

Simple Spark Apps: *WordCount*

WordCount Example 3

Definition:

count how often each word appears in a collection of text documents

This simple program provides a good test case for parallel processing, since it:

- requires a minimal amount of code
- demonstrates use of both symbolic and numeric values
- isn't many steps away from search indexing
- serves as a "Hello World" for Big Data apps

```
void map (String doc_id, String text):
  for each word w in segment(text):
    emit(w, "1");


void reduce (String word, Iterator group):
  int count = 0;

  for each pc in group:
    count += Int(pc);

  emit(word, String(count));
```

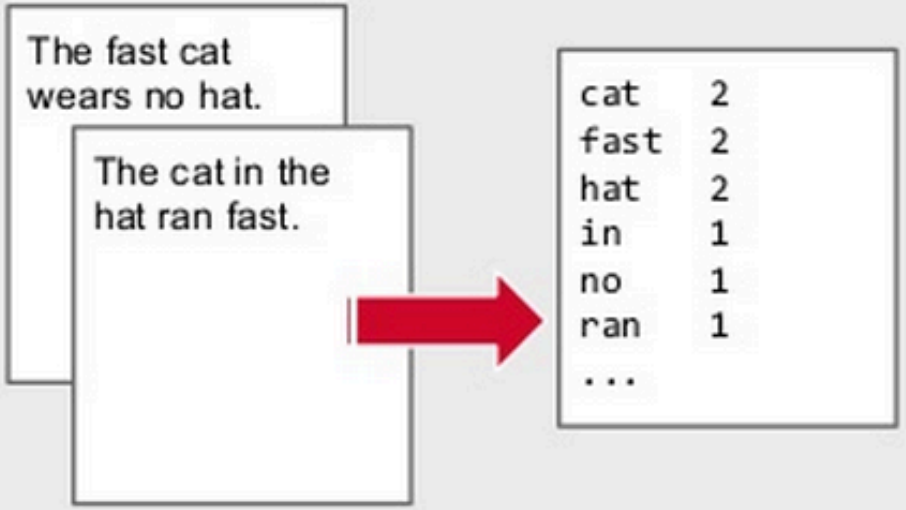
A distributed computing framework that can run WordCount **efficiently in parallel at scale** can likely handle much larger and more interesting compute problems

Word Count in Map-Reduce

```
def map(key, value):
    [redacted]
        emit(word, 1)

def reduce(key, values):
    [redacted]
        count += val
    emit(key, count)

# emit is a function that performs distributed I/O
```



Each document is passed to a mapper, which does the tokenization. The output of the mapper is reduced by key (word) and then counted.

What is the data flow for word count?

Word Count broken down

```
1 hello hi hi hallo
2 bonjour hola hi ciao
3 nihao konnichiwa ola
4 hola nihao hello
```

[hello, hi, hi, hallo, bonjour, hola, hi, ciao,
ao, konnichiwa, ola, hola, nihao, hello]

Mapper

(hello,1),
(hi,1),
(hi,1),
(hallo,1),
(bonjour,1),
(hola,1),
(hi,1),
(ciao,1),
(nihao,1),
(konnichiwa,1),
(ola,1),
(hola,1),
(nihao,1),
(hello,1)

reduce ByKey

(u'ciao', 1)
(u'bonjour', 1)
(u'nihao', 2)
(u'hola', 2)
(u'konnichiwa', 1)
(u'hallo', 1)
(u'hi', 3)
(u'hello', 2)
(u'ola', 1)

SPAM Data Word Count : HW1.2

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! s
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. You… [TRUNCATED BODY]**

WordCount example reads text files and counts how often words occur. The input is text files and the output is text files, each line of which contains a word and the count of how often it occured, separated by a tab.

MAPPER: Each mapper takes a line as input and breaks it into words. It then emits a key/value pair of the word and 1.

REDUCER: Each reducer sums the counts for each word and emits a single key/value with the word and sum.

SPAM Data Word Count : HW1.2

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!!
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below, is an update on the leadership pilot. You… [TRUNCATED BODY]**

As an optimization, the reducer is also used as a combiner on the map outputs. This reduces the amount of data sent across the network by combining each word into a single record. [See Lecture 3]

WordCount example … ccur. The input is text files and the outp… word and the count of how often it …

MAPPER: Each map… rds. It then emits a key/value pair of the word and 1.

REDUCER: Each reducer sums the counts for each word and emits a single key/value with the word and sum.

Word Count : HW1.2: Mapper

Input to Mapper

KEY: Date and employee\t

VALUE: SPAM \t Subject \t Email body

	A	B	C	D	E
1	0001.1999-12-10.farmer	0	christmas tree farm pictures	NA	
2	0001.1999-12-10.kaminski	0	re: rankings	thank you.	
3	0001.2000-01-17.beck	0	leadership development pilot	sally: what timing, ask and you s	
4	0001.2000-06-06.lokay	0	key dates and impact of upcoming sap implementation over the next few weeks,		
5	0001.2001-02-07.kitchen	0	key hr issues going forward	a) year end reviews-report needs	
6	0001.2001-04-02.williams	0	re: quasi	good morning, i'd love to go get	
7	0002.1999-12-13.farmer	0	vastar resources, inc.	gary, production from the high is	
8	0002.2001-02-07.kitchen	0	congrats!	contratulations on the execution	
9	0002.2001-05-25.SA_and_HP	1	fw: this is the solution i mentioned lsc	oo thank you, your email addres	
10	0002.2003-12-18.GP	1	adv: space saving computer to replace that big l	revolutionary!!! full featured!!! s	
11	0002.2004-08-01.BG	1	advs	greetings, i am benedicta lindiwe	
12	0003.1999-12-10.kaminski	0	re: visit to enron	vince, dec. 29 at 9:00 will be fine	

0001.2000-01-17.beck **0** leadership development pilot **sally: what timing, ask and you shall receive. as per our discussion, listed below is an update on the leadership pilot. You… [TRUNCATED BODY]**

For each record

- extract field 3 and 4 (Subject and Email Body) and split into tokens (words/numbe
- Out a list of token-count pairs

Output from Mapper

Mapper Output	Key=Word	Value=Count
Line 1	leadership	1
2	development	1
3	you	5

HW2

- **Repeat of HW1 modulo do everything in Hadoop and other minor tweaks**
  - Lift and shift mappers and reducers from HW1!

Naïve Bayes Questions

- **How many map-reduce jobs**

- **Why do we get 100% accuracy on the training set**
  - For smoothed NB
  - For Unsmoothed NB

From Word Counts to Mulitnomial Naïve Bayes HW1.3

- **Mapper(Key=DocID, Value=(SPAM, Subject, Body)**
- **Output: 3 different types of information**
  - Words and their class conditional counts (partial)
    - E.g., Assistance SPAM, 3
  - Class word counts
    - E.g., SPAM Word Count, 45
    - E.g., SPAM Doc Count, 1

- **Reducer(Key, Value)**
- **Output: Naïve Bayes Model**
  - Class conditionals Pr(X|Y); E.g., SPAM, Assistance 0.0001
  - Class prior Pr(Y); E.g., SPAM Prior =0.5

How many map-reduce jobs

- **How many Map-Reduce jobs do we need for the Multinomial Naïve Bayes problem?**

How many map-reduce jobs

- **How many Map-Reduce jobs do we need for the Multinomial Naïve Bayes problem?**
  - Hint: Train the classifier; classify each document using the learnt classifier
  -

How many map-reduce jobs

- **How many Map-Reduce jobs do we need for the Multinomial Naïve Bayes problem?**
  - Hint: Train the classifier; classify each document using the learnt classifier

- **In the second job do we use a mapper or reducer or both to do the classification?**

DocID	Class	w1	w2	w3	w4	Pr(SPAM) X P(SPAM| $DOC_{DocID=1}$)	Pr(HAM) X P(HAM|$DOC_{DocID=1}$)	
1	*SPAM*	*1*	*1*			*0.5 X (1/2 X ¼) = 1/16*	***0.5 X (2/6 X 0) = 0***	***SPAM***
2	SPAM	1				0.5 X (1/2)	0.5 X (2/6)	SPAM
3	**SPAM**			**1**		**0.5 X (¼)**	**0.5 X (1/3)**	**HAM**
4	*HAM*	*1*			*1*	*0.5 X (1/2) X 0*	*0.5 X (2/6 X 2/6)*	*HAM*
5	*HAM*			*1*	*1*	*0.5 X (1/4) X 0*	*0.5 X (2/6 X 2/6)*	*HAM*
6	**HAM**	**1**		**1**		**0.5 X (1/2) X 1/4**	**0.5 X (2/6 X 2/6)**	**SPAM**

We do NOT get 100% accuracy on the training set

P(SPAM|Doc) = P(Doc|SPAM) X Pr(SPAM) /Pr(Doc)

P(HAM|Doc) = P(Doc|HAM) X Pr(HAM) /Pr(Doc)

Joint conditional distribution

Pr(w1 and w2 and w3 and w4 |SPAM) = 0

Pr(w1 and w2 and w3 and w4 |HAM) = 0

Pr(w1 and w2…..w10000000|HAM)= 0

Assume Conditional independence

Pr(w1 and w2 and w3 and w4 |SPAM) = Pr(w1|SPAM) X Pr(w2|SPAM) X Pr(w3|SPAM) X Pr(w4|SPAM)

Independence

Pr(A, B) = P(A) X P(B) E.g., P(King) = 4/52=1/13 Tell Alex: we have spades! What is Pr(King|Spades)? 1/13

Class conditional probabilites

Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0
Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6

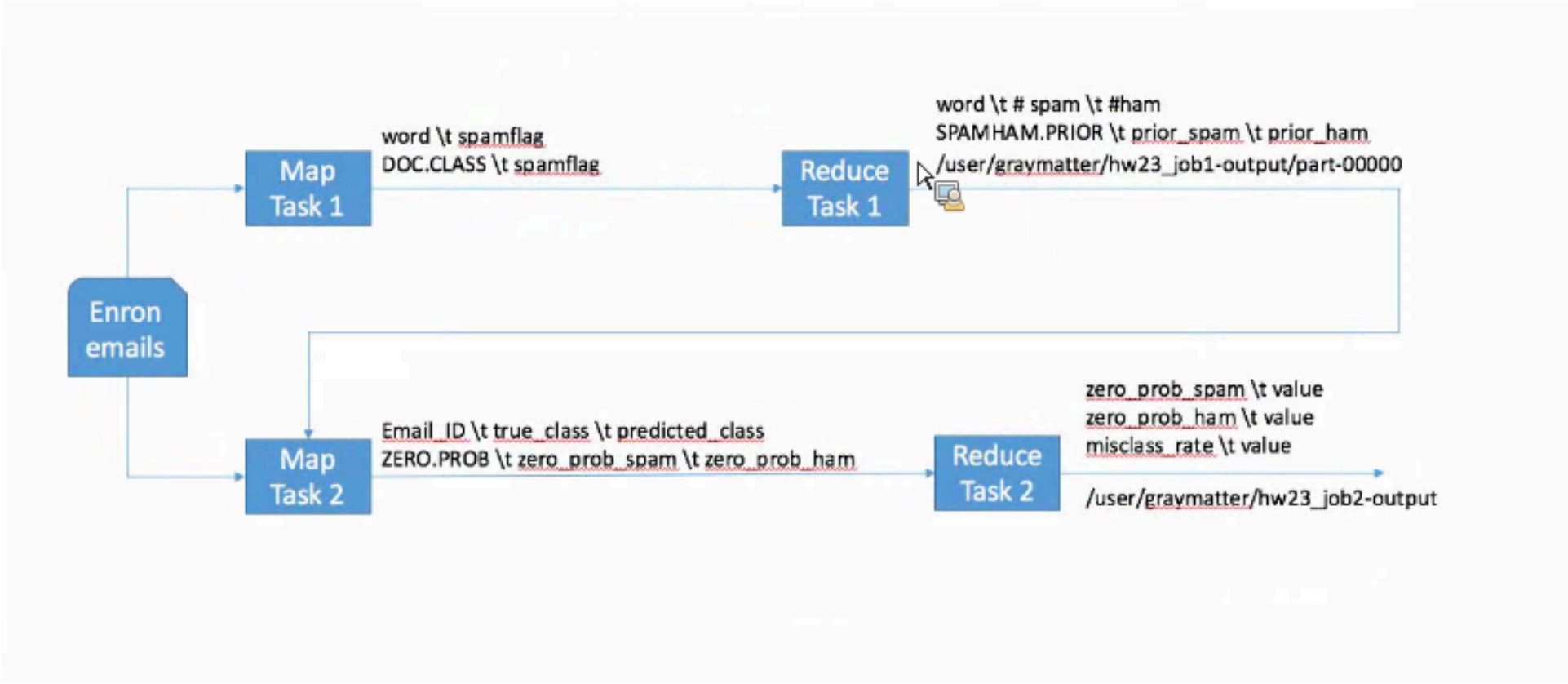
Pr(SPAM)=3/6
Pr(HAM)= 3/6

Word	unsmoothed Pr(Word| SPAM)	unsmoothed Pr(Word| SPAM)	Laplace+1 Smoothed Pr(Word| SPAM)	Laplace+1 Smoothed Pr(Word|HAM)
w1	2/4	2/6	(2+1)/(6+4)	
w2	1/4	0		
w3	1/4	2/6		
w4	0	2/6		

Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0

Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6

MapReduce Jobs for Multinomial NB



```
In [28]: #usr/local/Cellar/hadoop/2.6.0/libexec/share/hadoop/tools/lib
dataDir = "/Users/jshanahan/Dropbox/lectures-uc-berkeley-ml-class-2015/Notebooks/WordCount"

!hadoop jar /usr/local/Cellar/hadoop/2.6.0/libexec/share/hadoop/tools/lib/hadoop-streaming*.jar \
   -mapper WordCount/mapper.py \
   -reducer WordCount/reducer.py \
   -input historical_tours.txt \
   -output gutenberg-output  \
   -numReduceTasks 2
   #--D mapreduce.job.reduces=2
#-input historical_tours.txt  file on Hadoop
#output directory on Hadoop
```

```
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=9
        Map-Reduce Framework
                Map input records=1941
                Map output records=13741
                Map output bytes=108364
                Map output materialized bytes=135858
                Input split bytes=109
                Combine input records=0
                Combine output records=0
                Reduce input groups=3736
                Reduce shuffle bytes=135858
                Reduce input records=13741
                Reduce output records=3736
                Spilled Records=27482
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=25
```

Multiple reducers in Hadoop result in partition result files

part-00000

….

part-0000N

```
In [27]: pwd
Out[27]: u'/Users/jshanahan/Dropbox/lectures-uc-berkeley-ml-class-2015/Notebooks'

In [18]: !hdfs dfs -ls
16/01/27 20:49:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builti
n-java classes where applicable
Found 2 items
drwxr-xr-x   - jshanahan supergroup          0 2016-01-27 20:49 gutenberg-output
-rw-r--r--   1 jshanahan supergroup      87483 2015-02-26 19:36 historical_tours.txt

In [29]: !hdfs dfs -ls gutenberg-output
16/01/27 21:10:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builti
n-java classes where applicable
Found 3 items
-rw-r--r--   1 jshanahan supergroup          0 2016-01-27 21:10 gutenberg-output/_SUCCESS
-rw-r--r--   1 jshanahan supergroup      18505 2016-01-27 21:10 gutenberg-output/part-00000
-rw-r--r--   1 jshanahan supergroup      18043 2016-01-27 21:10 gutenberg-output/part-00001
```

If we have multiple reducers then

Naïve Bayes: Scenario 1: 10 mappers and 1 reducer

- **Assume we have 10 mappers and 1 reducer**
  - Mapper: emits counts for words, classes
  - Reducer: aggregates to yield class conditional probs and priors
    - Smoothing
  - Save model to disk (output directory)
    - outputDir/PART-0000
    - This lives on HDFS in triplicate form
- **2nd MapReduce Job**
  - Mapper: classify each example using the learning model
  - A single Reducer aggregates classifications and calculates the error/metrics

Class conditional probabilites

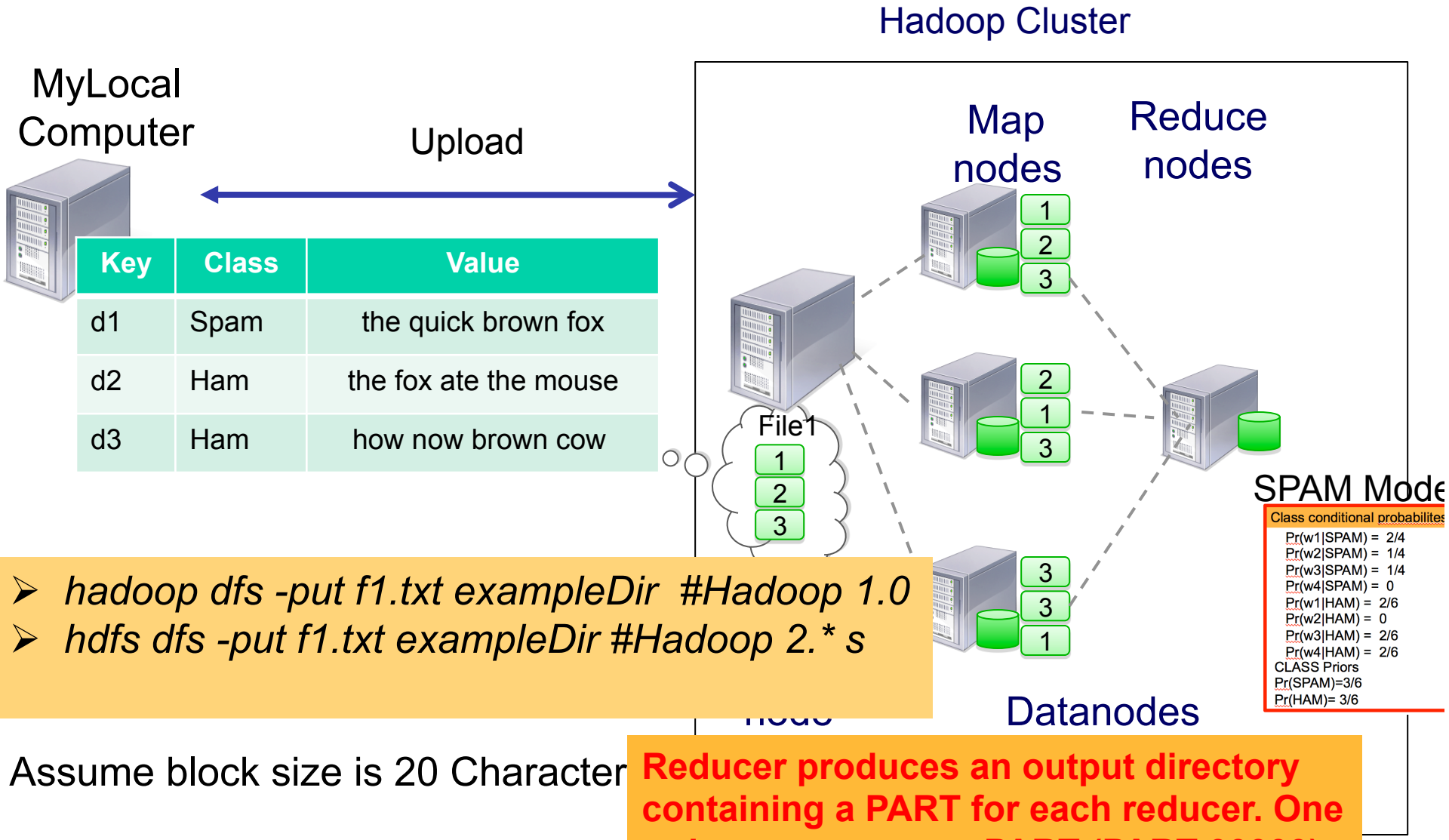
Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0

Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6

CLASS Priors
Pr(SPAM)=3/6
Pr(HAM)= 3/6

Pr(w1|SPAM) = Count(w1 in SPAM documents)/(Total WordCount in SPAM class)

Learn the NB Model:
Hadoop Cluster: 1 Name;
3 data node+ 1 reducer nodes



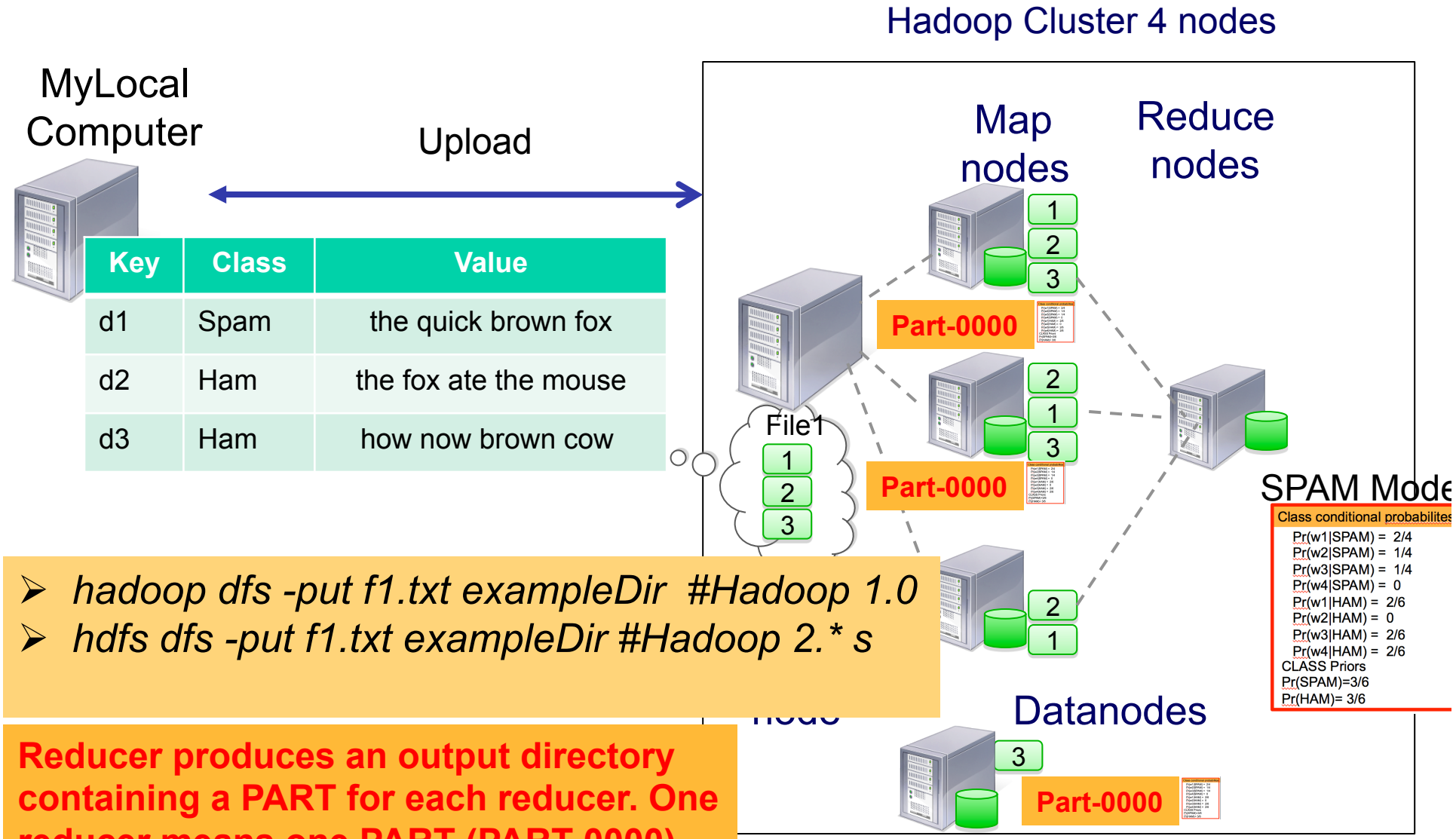
Key	Class	Value
d1	Spam	the quick brown fox
d2	Ham	the fox ate the mouse
d3	Ham	how now brown cow

- *hadoop dfs -put f1.txt exampleDir #Hadoop 1.0*
- *hdfs dfs -put f1.txt exampleDir #Hadoop 2.* s*

Assume block size is 20 Character

Reducer produces an output directory containing a PART for each reducer. One reducer means one PART (PART-00000)

Learn the NB Model:
Hadoop Cluster: 1 Name;
3 data node+ 1 reducer nodes



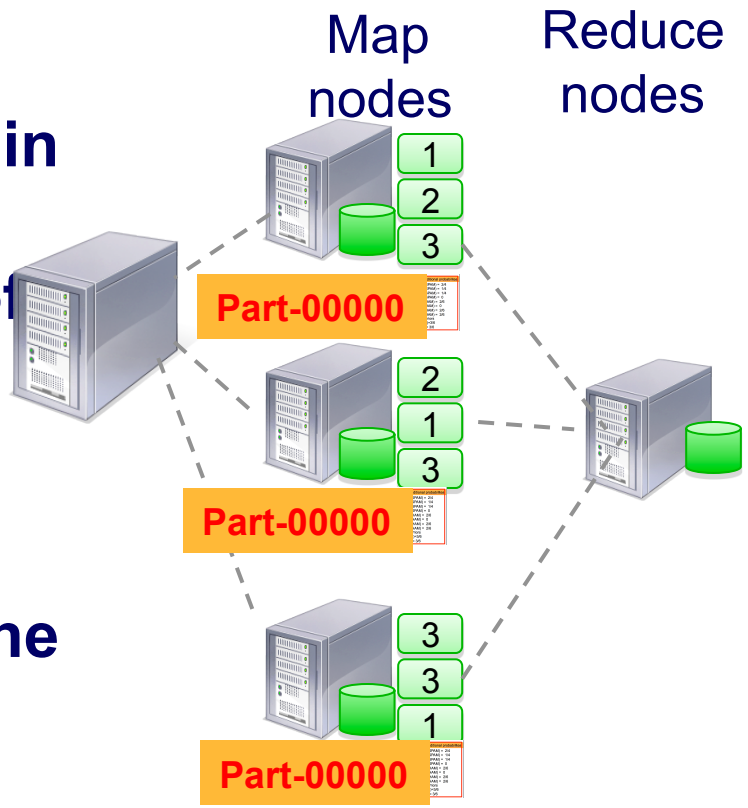
Key	Class	Value
d1	Spam	the quick brown fox
d2	Ham	the fox ate the mouse
d3	Ham	how now brown cow

- *hadoop dfs -put f1.txt exampleDir #Hadoop 1.0*
- *hdfs dfs -put f1.txt exampleDir #Hadoop 2.* s*

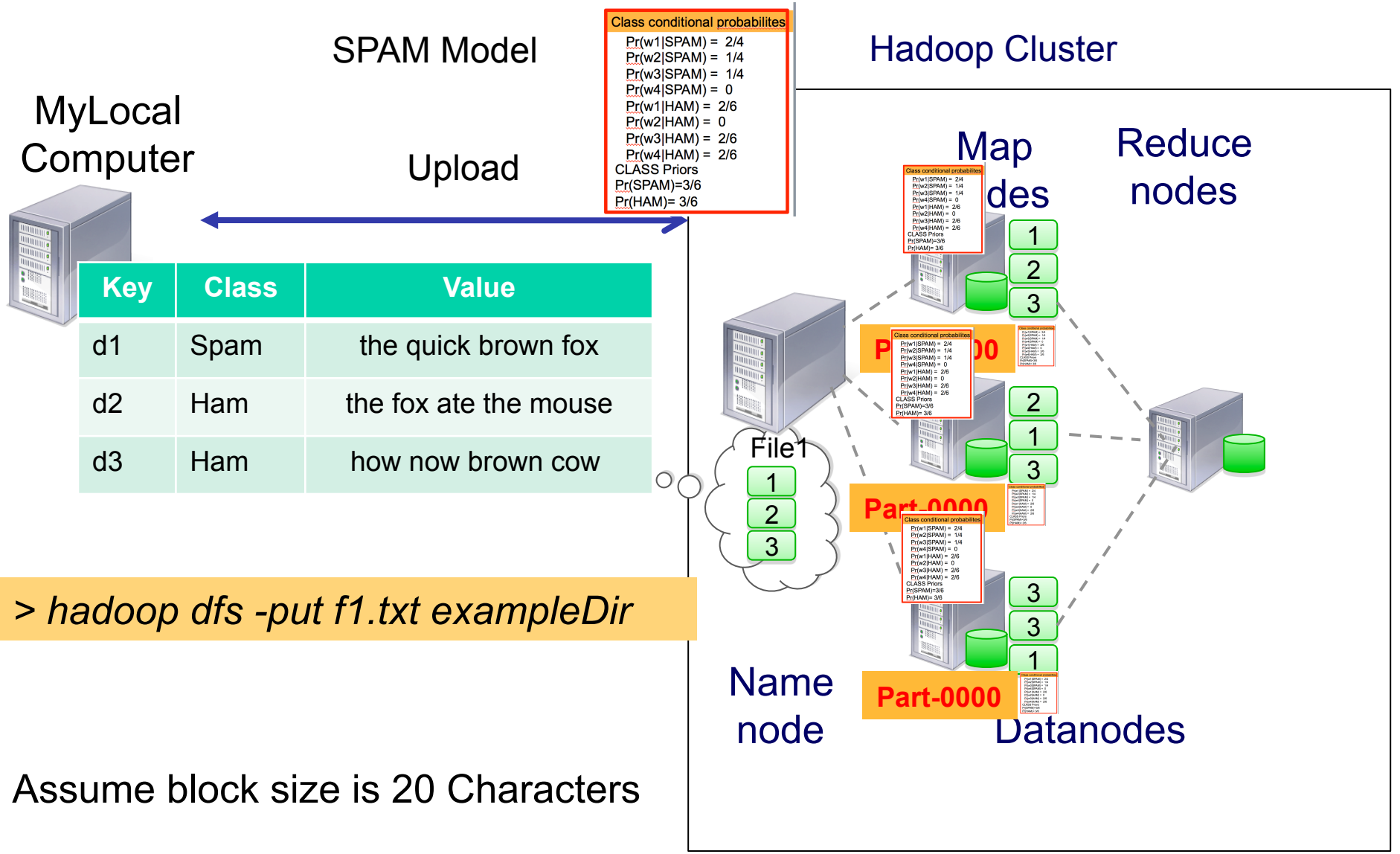
Reducer produces an output directory containing a PART for each reducer. One reducer means one PART (PART-0000)

Classification Task

- **NB Classifier: Pass model to Mapper in the 2nd job**
- **Though the model is stored in a HDFS directory it is not accessible to the mappers of the second Classifier MapReduce Job**
- **2 ways to get the model to the classifier Job (primarily the mappers):**
  - or shell command(- hdfs –cat)
  - -files;



NB Classifier: Pass model to Mapper in the 2nd job
2 ways: -files; or shell command(- hdfs –cat)



Run shell commands from python: subprocess.Popen(…)

- **Popen**
  - spawn new processes, connect to their input/output/error pipes, and obtain their return codes

Python Subprocess Module

- As you begin to create Python scripts you will likely find yourself leveraging os.system and subprocess.Popen because they let you run OS commands.
- The main difference between os.system and subprocess.Popen is that subprocess allows you to redirect STDOUT to a variable in Python.

```
>>> import subprocess
>>>
>>> com_str = 'uname -a'
>>> command = subprocess.Popen([com_str], stdout=subprocess.PIPE, shell=True)
>>> (output, error) = command.communicate()
>>> print output
Linux cell 3.11.0-20-generic #35~precise1-Ubuntu SMP Fri May 2 21:32:55 UTC 201
```

com_str = 'ls –d'

Read the output of a previous job as input directly using Hadoop Cat command

Mapper (classification)

Mapper for Classification task
Load the model directly from HDFS

```
%%writefile mapper_c.py
#!/usr/bin/python
import sys, re, string, subprocess
# read the probability from HDFS
prob = {}
cat = subprocess.Popen(["hadoop", "fs", "-cat", "prob/part-00000"], stdout=subprocess.PIPE)
for line in cat.stdout:
    word, p0, p1 = line.split()
    prob[word] = [p0, p1]

# get prior probability
prior = prob['prior_prob']
```

Shell out to hadoop cat command

Process each email in this chunk

```
# input comes from STDIN (standard input)
for line in sys.stdin:
    # use subject and body
    msg = line.split('\t', 2)
    # skip bad message
    if len(msg) < 3:
```

NB Classifier: Pass model to Mapper

SPAM Model

Class conditional probabilites
Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0
Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6
CLASS Priors
Pr(SPAM)=3/6
Pr(HAM)= 3/6

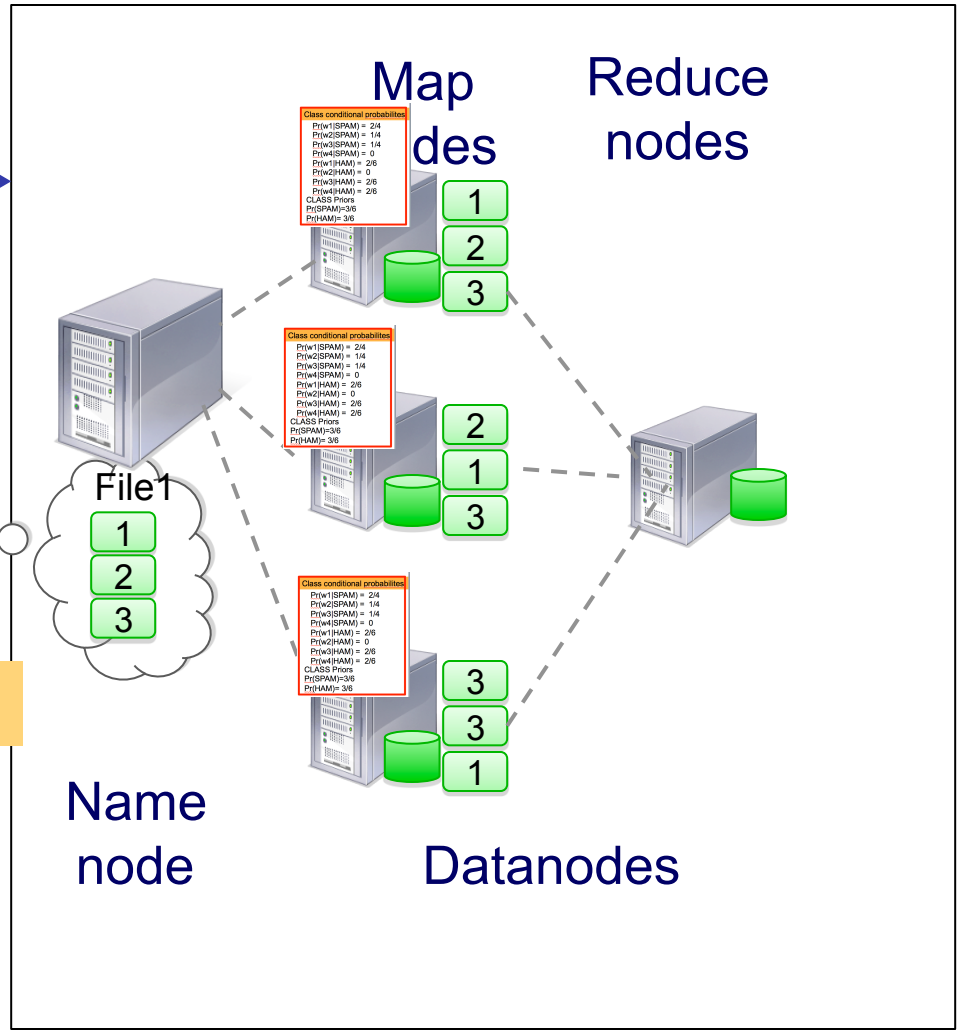
MyLocal Computer

Upload

Key	Class	Value
d1	Spam	the quick brown fox
d2	Ham	the fox ate the mouse
d3	Ham	how now brown cow

> *hadoop dfs -put f1.txt exampleDir*

Hadoop Cluster



Assume block size is 20 Characters

Copy a file from Hadoop back to your local filesystem

- .

STEP 3: DOWNLOAD FILE FROM HDFS TO LOCAL FILE SYSTEM

hadoop fs -get:

- Copies/Downloads files from HDFS to the local file system

```
# Usage:
        # hadoop fs -get <hdfs_src> <localdst>
# Example:
        hadoop fs -get /user/hadoop/dir1/popularNames.txt /home/
```

NB Classifier: Pass model to Mapper in the 2nd job
2 ways: -files; or shell command(- hdfs –cat)

SPAM Model

MyLocal Com

Class conditional probabilites
Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0
Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6
CLASS Priors
Pr(SPAM)=3/6
Pr(HAM)= 3/6

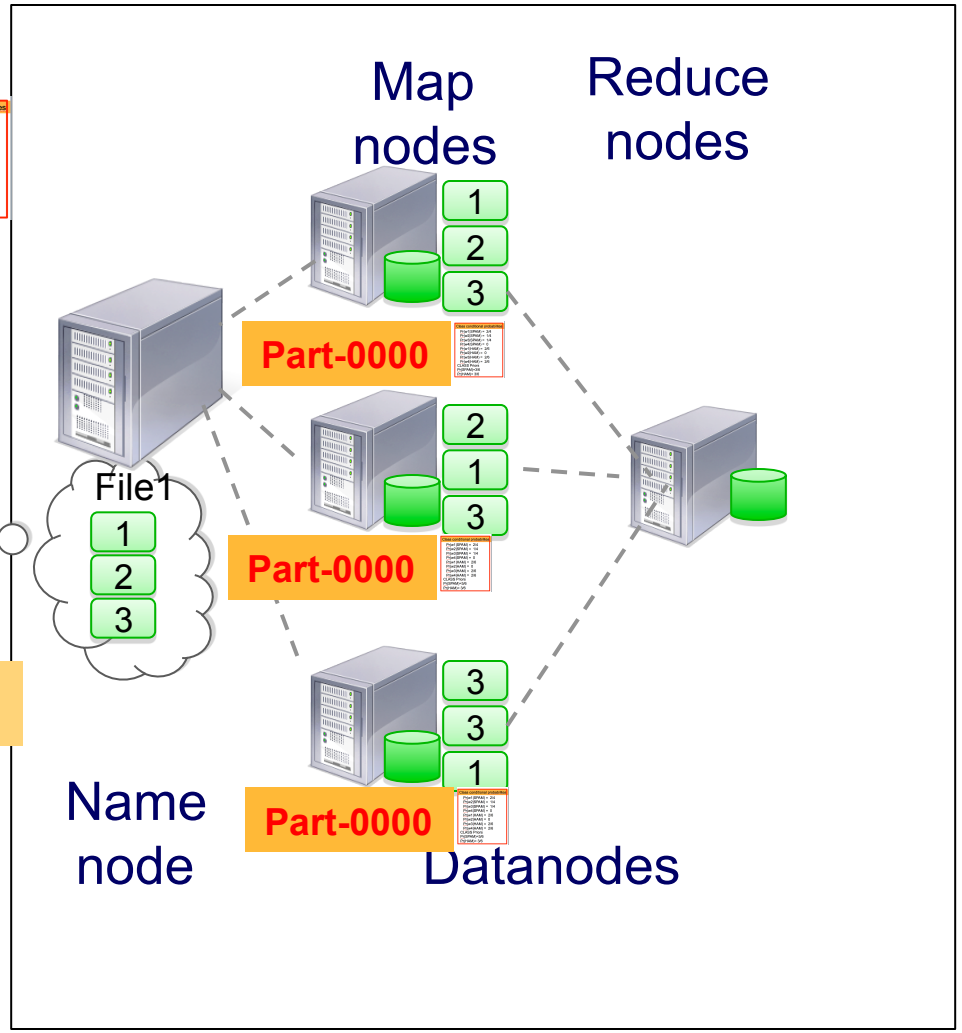
Hadoop fs –get OR use -distcp

		Value
		the quick brown fox
d2	Ham	the fox ate the mouse
d3	Ham	how now brown cow

> hadoop dfs -put f1.txt exampleDir

Assume block size is 20 Characters

Hadoop Cluster



NB Classifier: Pass model to Mapper in the 2nd job
2 ways: -files; or shell command(- hdfs –cat)

SPAM Model

MyLocal Com

Class conditional probabilites
Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0
Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6
CLASS Priors
Pr(SPAM)=3/6
Pr(HAM)= 3/6

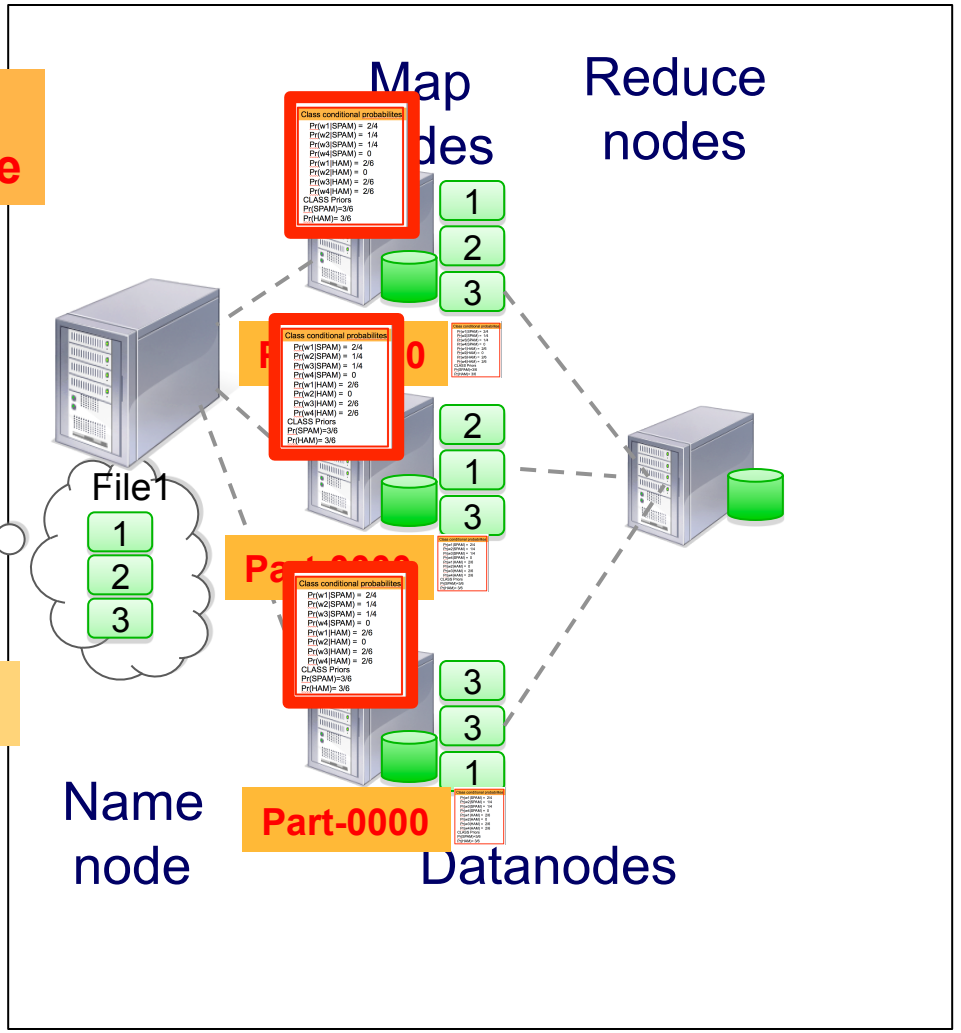
-files XXXXX
next slides for example

		Value
		the quick brown fox
d2	Ham	the fox ate the mouse
d3	Ham	how now brown cow

> hadoop dfs -put f1.txt exampleDir

Assume block size is 20 Characters

Hadoop Cluster



Distributed Cache:
read-only data/text files and more complex types such as archives and jars

James G. Shanahan[1]

[1]***NativeX and iSchool, UC Berkeley, CA***

EMAIL: James_DOT_Shanahan_AT_gmail_DOT_com

Hadoop Streeaming
Define mapper/reducer functions

```
%%writefile reducerhw2_31.py
#!/usr/bin/python
"""
Reducer code for W261 exercise 2.3
"""
__author__="Carlos Eduardo Rodriguez Castillo"
__email__="cerodriguez@berkeley.edu"

## Import relevant libraries
import sys
import re

#def reducer_initialize():

def reduce_function(record):
    record = record.strip()
    record_parameters = record.split("\t")
    accuracy_counts = record_parameters[1]
    accuracy_counts = accuracy_counts.split(",")
    accuracy_count = int(accuracy_counts[0])
    inaccuracy_count = int(accuracy_counts[1])
    HAM_log_prob = float(accuracy_counts[2])
    SPAM_log_prob = float(accuracy_counts[3])
    return (accuracy_count, inaccuracy_count, HAM_log_prob, SPAM_log_prob)

if __name__ == "__main__":
    #record_dictionary = reducer_initialize()
    total, accurate_count, inaccurate_count = 0, 0, 0
    for line in sys.stdin:
        a, i, H_log_prob, S_log_prob = reduce_function(line)
        accurate_count = accurate_count + a
        inaccurate_count = inaccurate_count + i
        total = total + 1
        ## printing the conditional probabilities for the histograms
        print "HAM_log_prob_cond_doc\t%.2f\tSPAM_log_prob_cond_doc\t%.2f" % (H_log_prob, S_log_prob)
    ## printing the misclassification rate
    print "Misclassifcation_error_rate_multinomial_Naive_Bayes_Classifier\t%.2f" % ((float(inaccurate_count)/float(total)))
```

https://www.dropbox.com/s/l3jekh6hp2fg8y7/MIDS-W261-2015-HWK-Week02-RodriguezCastillo-GreatUse-of-functions.ipynb?dl=0

Uses main: and locally defined mapper-init, mapper, reducer etc.

How to pass a command line argument to a mapper/reducer

In [183]:
```
%%writefile mapper2.2.py
#!/usr/bin/python
import sys
import re
WORD_RE = re.compile(r"[\w']+")
findword = sys.argv[1]
# make list of words entered by user for classification
findwords = re.findall(WORD_RE,findword)
findall = False
if len(findwords) == 0:
    findall = True

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into cols
    cols = line.split('\t')
    if len(cols) == 4:
        words = re.split('\s+|\,+|\.+',(cols[2] + " " + cols[3]))
        #words = re.findall(WORD_RE,(cols[2] + " " + cols[3]))
    else:
        # this is a nasty hack for now. row 60 only has the content column
        # but also contains the word assistance
        words = re.split('\s+|\,+|\.+',(cols[0]))
        #words = re.findall(WORD_RE,(cols[2] + " " + cols[3]))
    # increase counters
    for word in words:
        if word in findwords or findall == True:
            # write the results to STDOUT (standard output);
            # what we output here will be the input for the
            # Reduce step, i.e. the input for reducer.py
            #
            # tab-delimited; the trivial word count is 1
            print '%s\t%s' % (word, 1)
```

```
Overwriting mapper2.2.py
```

In [186]:
```
!hdfs dfs -rm -r /user/koza/hw22/output
!hadoop jar /usr/local/Cellar/hadoop/2.7.2/libexec/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar \
    -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator \
    -files mapper2.2.py,reducer2.2.py \
    -mapper 'mapper2.2.py assistance' \
    -reducer reducer2.2.py \
    -input /user/koza/enron/* -output /user/koza/hw22/output
```

Running your job on EMR Cluster using hadoop and S3

```
aws s3 rm s3://dz-w261-hw2/2_2_1/output                    \
hadoop jar /usr/lib/hadoop/hadoop-streaming-2.7.2-amzn-2.jar \ -D
mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator \ -
D stream.map.output.field.separator="\t" \ -D mapreduce.partition.keycomparator.options="-k2,2nr"
\ -mapper /bin/cat \ -reducer /bin/cat \ -numReduceTasks 1 \ -input s3://dz-w261-hw2/2_2/output/
part* -output s3://dz-w261-hw2/2_2_1/output                \
 hdfs dfs -cat s3://dz-w261-hw2/2_2_1/output/part-00000| head -10
```

```
!cat enronemail_1h.txt | ./hw2_2_mapper.py | sort -k1,1 |./hw2_2_reducer.py |grep "assistance"
```

```
assistance      10
```

Run job on AWS by launching an EMR Cluster

Make sure output directory doesn't exist hdfs dfs -rm -r s3n://dz-w261-hw2/hw2_2/output hadoop jar /usr/lib/hadoop/hadoop-streaming-2.7.2-amzn-2.jar \ -files s3n://dz-w261-hw2/hw2_2_mapper.py,s3n://dz-w261-hw2/hw2_2_reducer.py \ -input s3://dz-w261-hw2/enronemail_1h.txt \ -output s3://dz-w261-hw2/2_2/output \ -mapper hw2_2_mapper.py \ -reducer hw2_2_reducer.py \ -numReduceTasks 1 \

HW2.2 Results:

hdfs dfs -cat s3://dz-w261-hw2/2_2/output/combined.txt| grep "assistance"[hadoop@ip-172-31-17-191 ~]$ hdfs dfs -cat s3://dz-w261-hw2/2_2/output/part-00000| grep "assistance"
16/06/05 15:12:33 INFO s3n.S3NativeFileSystem: Opening 's3://dz-w261-hw2/2_2/output/part-00000' for reading assistance 10

HW2.2.1 Using Hadoop MapReduce and your wordcount job (from HW2.2) determine the top-10 occurring tokens (most frequent tokens)

aws s3 rm s3://dz-w261-hw2/2_2_1/output hadoop jar /usr/lib/hadoop/hadoop-streaming-2.7.2-amzn-2.jar \ -D
mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator \ -D stream.map.output.field.separator="\t" \ -D
mapreduce.partition.keycomparator.options="-k2,2nr" \ -mapper /bin/cat \ -reducer /bin/cat \ -numReduceTasks 1 \ -input s3://dz-w261-hw2/2_2/output/part* -output s3://dz-w261-hw2/2_2_1/outputhdfs dfs -cat s3://dz-w261-hw2/2_2_1/output/part-00000| head -10

HW2.2.1 Results:

[hadoop@ip-172-31-17-191 ~]$ hdfs dfs -cat s3://dz-w261-hw2/2_2_1/output/part-00000| head -10 16/06/05 15:21:05 INFO s3n.S3NativeFileSystem: Opening 's3://dz-w261-hw2/2_2_1/output/part-00000' for reading the 1247 to 964 and 670 of 566 you 445 in 418 your 395 ect 382 for 374 on 271

Job Name

- ..

```
!hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-mr1.jar \
    -D mapreduce.job.name="Ex28-TrainModel" \
    -D mapreduce.job.reduces=2 \
    -D stream.map.output.field.separator=\t \
    -D stream.num.map.output.key.fields=1 \
    -D mapreduce.partition.keypartitioner.options="-k1,1" \
    -D mapreduce.job.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator \
    -D mapreduce.partition.keycomparator.options="-k1,1 -k2,2" \
    -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner \
    -file /home/cloudera/w261/ps2/ex28/modelmap.py \
    -mapper /home/cloudera/w261/ps2/ex28/modelmap.py \
    -file /home/cloudera/w261/ps2/ex28/modelred.py \
    -reducer /home/cloudera/w261/ps2/ex28/modelred.py \
    -input /user/cloudera/w261/ps2/training-data/* \
    -output /user/cloudera/w261/ps2/ex28/model-out
```

-file is for executables only

Packaging Files With Job Submissions

You can specify any executable as the mapper and/or the reducer. The executables do not need to pre-exist on the machines in the cluster; however, if they don't, you will need to use "-file" option to tell the framework to pack your executable files as a part of job submission. For example:

```
hadoop jar hadoop-streaming-2.5.2.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper myPythonScript.py \
    -reducer /usr/bin/wc \
    -file myPythonScript.py
```

The above example specifies a user defined Python executable as the mapper. The option "-file myPythonScript.py" causes the python executable shipped to the cluster machines as a part of job submission.

In addition to executable files, you can also package other auxiliary files (such as dictionaries, configuration files, etc) that may be used by the mapper and/or the reducer. For example:

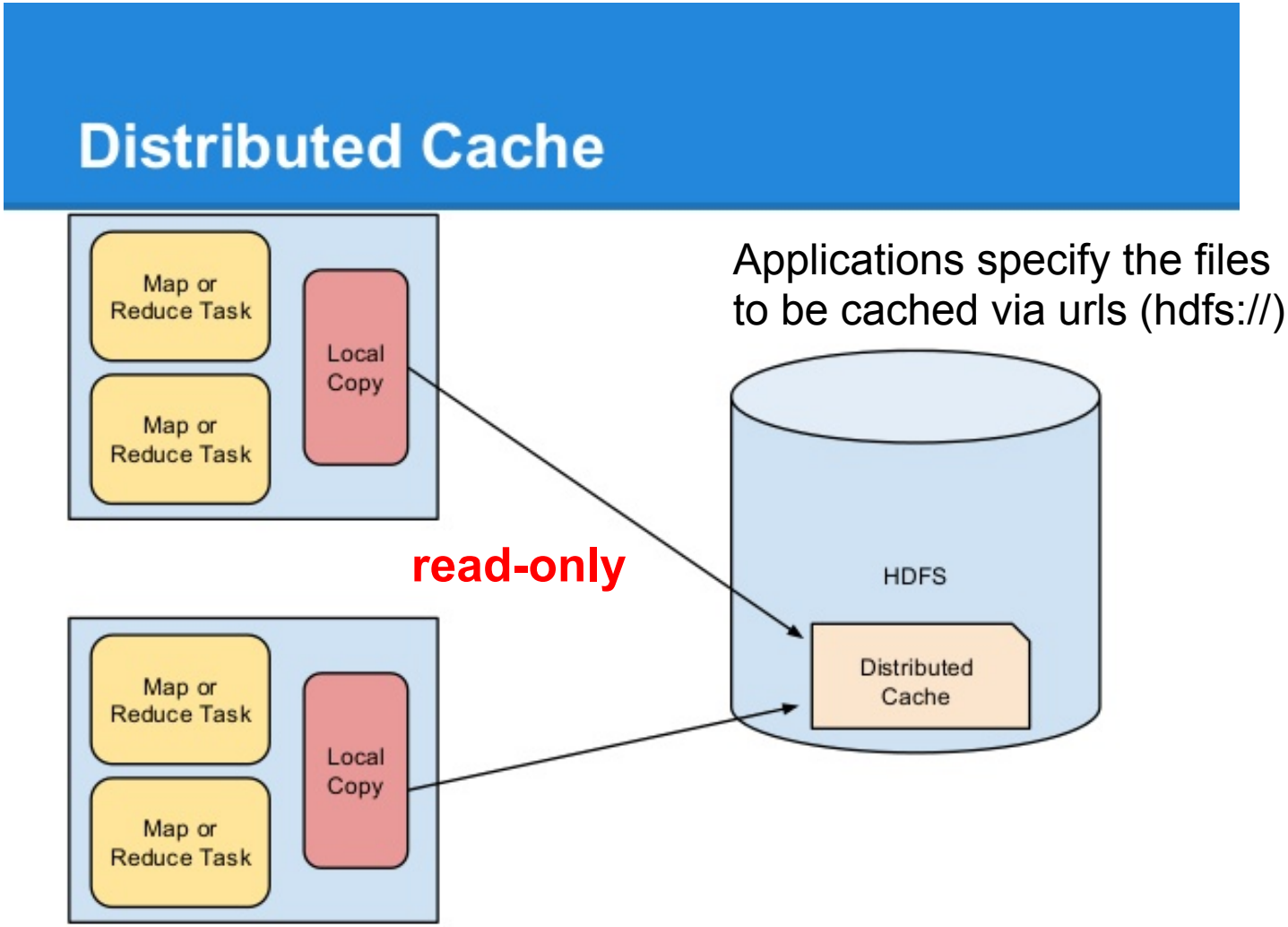
```
hadoop jar hadoop-streaming-2.5.2.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper myPythonScript.py \
    -reducer /usr/bin/wc \
    -file myPythonScript.py \
    -file myDictionary.txt
```

Distributed Cache in Hadoop

- **Distributed Cache is a facility provided by the MapReduce framework to cache files (text, archives, jars and so on) needed by applications.**

- **Distributed Cache can be used to distribute simple, read-only data/text files and more complex types such as archives and jars.**

Distributed Cache in Hadoop



Distributed Cache in Hadoop streaming

- **File is distributed cached through -files**

-files hdfs://host:fs_port/user/file.txt#filename

URL of the file File name in mapper and reducer

- **Now you can access the file in mapper and reducer**

open(filename,'r')

Does this work to access hdfs files?

Example

Hadoop streaming script

```
!hadoop jar hadoop-*streaming*.jar -files 'dictionary.txt#dictionary' \
                                   -mapper mapper.py -reducer reducer.py \
                                   -input wordcount.txt \
                                   -output wordcountDictOutput
```

Mapper

```
%%writefile mapper.py
#!/usr/bin/python
import sys
# input comes from STDIN (standard input)
f = open('dictionary', 'r')
word_dict = []
for line in f:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    for word in words:
        word_dict.append(word)
```

Example: Filtered WordCount

- http://nbviewer.ipython.org/
- http://nbviewer.jupyter.org/urls/dl.dropbox.com/s/zll36pds0z1bqtp/Hadoop%20Streaming%20WordCount-Distributedcache.ipynb

- https://www.dropbox.com/s/zll36pds0z1bqtp/Hadoop%20Streaming%20WordCount-Distributedcache.ipynb?dl=0

Wordcount for a specified dictionary of words

Data

```
%%writefile wordcount.txt
hello hi hi hallo
bonjour hola hi ciao
nihao konnichiwa ola
hola nihao hello
```

```
Overwriting wordcount.txt
```

Dictionary

```
%%writefile dictionary.txt
hello hi
```

```
Writing dictionary.txt
```

Mapper

```
%%writefile mapper.py
#!/usr/bin/python
import sys
# input comes from STDIN (standard input)
f = open('dictionary', 'r')
word_dict = []
for line in f:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    for word in words:
        word_dict.append(word)

for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        if word in word_dict:
            print '%s\t%s' % (word, 1)
```

```
Overwriting mapper.py
```

Hadoop streaming command

You can package other auxiliary files (such as dictionaries, configuration files, etc) that may be used by the mapper and/or the reducer. For example:

```
hadoop jar hadoopstreamingjarfile \
    -files 'url#filename' \
    -mapper mapperfile \
    -reducer reducerfile \
    -input inputfile \
    -output outputfile
```

hadoop streaming jar file can be found in your hadoop folder or downloaded from http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-streaming/2.6.0

```
!hadoop jar hadoop-*streaming*.jar -files 'dictionary.txt#dictionary'  -mapper mapper.py -reducer reducer.py -input wordcount.txt -output wordcountDictOutput
```

Reading in Dictionary values (key-value records) in python

Starting in Python 2.6 you can use the built-in `ast.literal_eval`:

```
>>> import ast
>>> ast.literal_eval("{'muffin' : 'lolz', 'foo' : 'kitty'}")
{'muffin': 'lolz', 'foo': 'kitty'}
```

This is safer than using `eval`. As its own docs say:

```
>>> help(ast.literal_eval)
Help on function literal_eval in module ast:

literal_eval(node_or_string)
    Safely evaluate an expression node or a string containing a Python
    expression.  The string or node provided may only consist of the following
    Python literal structures: strings, numbers, tuples, lists, dicts, booleans,
    and None.
```

OR via JSON

using `json.loads`

```
>>> import json
>>> h = '{"foo":"bar", "foo2":"bar2"}'
>>> type(h)
<type 'str'>
>>> d = json.loads(h)
>>> d
{u'foo': u'bar', u'foo2': u'bar2'}
>>> type(d)
<type 'dict'>
```

Mapper

http://nbviewer.ipython.org/urls/dl.dropbox.com/s/zll36pds0z1bqtp/Hadoop%20Streaming%20WordCount-Distributedcache.ipynb

- ...

```
%%writefile mapper.py
#!/usr/bin/python
import sys
# input comes from STDIN (standard input)
f = open('dictionary', 'r')
word_dict = []
for line in f:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    for word in words:
        word_dict.append(word)

for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        if word in word_dict:
            print '%s\t%s' % (word, 1)
```

4. Generic Command Options

Streaming supports streaming command options as well as generic command options. The general command line syntax is shown below.

Note: Be sure to place the generic options before the streaming options, otherwise the command will fail. For an example, see Making Archives Available to Tasks.

```
bin/hadoop command [genericOptions] [streamingOptions]
```

The Hadoop generic command options you can use with streaming are listed here:

Parameter	Optional/Required	Description
-conf configuration_file	Optional	Specify an application configuration file
-D property=value	Optional	Use value for given property
-fs host:port or local	Optional	Specify a namenode
-jt host:port or local	Optional	Specify a job tracker
-files	Optional	Specify comma-separated files to be copied to the Map/Reduce cluster
-libjars	Optional	Specify comma-separated jar files to include in the classpath
-archives	Optional	Specify comma-separated archives to be unarchived on the compute machines

4.1. Specifying Configuration Variables with the -D Option

You can specify additional configuration variables by using "-D <property>=<value>".

4.1.1. Specifying Directories

To change the local temp directory use:

```
-D dfs.data.dir=/tmp
```

To specify additional local temp directories use:

```
-D mapred.local.dir=/tmp/local
-D mapred.system.dir=/tmp/system
```

-files

-files

Specify comma-separated files to be copied to the Map/ Reduce cluster

Working with Large Files and Archives
The -files and -archives options allow you to make files and archives available to the tasks. The argument is a URI to the file or archive that you have already uploaded to HDFS. These files and archives are cached across jobs.

Working with Large Files and Archives

The -files and -archives options allow you to make files and archives available to the tasks. The argument is a URI to the file or archive that you have already uploaded to HDFS. These files and archives are cached across jobs. You can retrieve the host and fs_port values from the fs.default.name config variable.

Note: The -files and -archives options are generic options. Be sure to place the generic options before the command options, otherwise the command will fail. For an example, see The -archives Option. Also see Other Supported Options.

Making Files Available to Tasks

The -files option creates a symlink in the current working directory of the tasks that points to the local copy of the file.

In this example, Hadoop automatically creates a symlink named testfile.txt in the current working directory of the tasks. This symlink points to the local copy of testfile.txt.

```
-files hdfs://host:fs_port/user/testfile.txt
```

User can specify a different symlink name for -files using #.

```
-files hdfs://host:fs_port/user/testfile.txt#testfile
```

Multiple entries can be specified like this:

```
-files hdfs://host:fs_port/user/testfile1.txt,hdfs://host:fs_port/user/testfile2.txt
```

Hadoop Streaming Manual

- **All sorts of great explanations for Hadoop Streaming**
  - E.g., -file option to pass files or directories (code or data to Mapper/Reducer)

- https://hadoop.apache.org/docs/r1.2.1/streaming.html
- https://hadoop.apache.org/docs/r1.2.1/streaming.pdf

Passing a file to Hadoop

- **The way *-file* option works is that the file (EXECUTABLE file) is copied to the worker node**
- **-file works with a directory or file**
  - I have a hadoop streaming job which takes a directory and runs against the 6 files in the folder.

- **Remember that the path supplied for the -file command is the path in HDFS, so use the *ls* command to make sure that the path is correct.**

Hadoop Streaming exampe

Hadoop Streaming

Hadoop streaming is a utility that comes with the Hadoop distribution. The utility allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer. For example:

```
$HADOOP_HOME/bin/hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper /bin/cat \
    -reducer /bin/wc
```

How Streaming Works

In the above example, both the mapper and the reducer are executables that read the input from stdin (line by line) and emit the output to stdout. The utility will create a Map/Reduce job, submit the job to an appropriate cluster, and monitor the progress of the job until it completes.

When an executable is specified for mappers, each mapper task will launch the executable as a separate process when the mapper is initialized. As the mapper task runs, it converts its inputs into lines and feed the lines to the stdin of the process. In the meantime, the mapper collects the line oriented outputs from the stdout of the process and converts each line into a key/value pair, which is collected as the output of the mapper. By default, the *prefix of a line up to the first tab character* is the **key** and the rest of the line (excluding the tab character) will be the **value**. If there is no tab character in the line, then entire line is considered as key and the value is null. However, this can be customized, as discussed later.

When an executable is specified for reducers, each reducer task will launch the executable as a separate process then the reducer is initialized. As the reducer task runs, it converts its input key/values pairs into lines and feeds the lines to the stdin of the process. In the meantime, the reducer collects the line oriented outputs from the stdout of the process, converts each line into a key/value pair, which is collected as the output of the reducer. By default, the prefix of a line up to the first tab character is the key and the rest of the line (excluding the tab character) is the value. However, this can be customized, as discussed later.

This is the basis for the communication protocol between the Map/Reduce framework and the streaming mapper/reducer.

You can supply a Java class as the mapper and/or the reducer. The above example is equivalent to:

```
$HADOOP_HOME/bin/hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper org.apache.hadoop.mapred.lib.IdentityMapper \
    -reducer /bin/wc
```

User can specify `stream.non.zero.exit.is.failure` as `true` or `false` to make a streaming task that exits with a non-zero status to be `Failure` or `Success` respectively. By default, streaming tasks exiting with non-zero status are considered to be failed tasks.

Passing files to a Mapper /Reducer: -file (executable files) versus -files (data files)

3.2 Packaging Files With Job Submissions

You can specify any executable as the mapper and/or the reducer. The executables do not need to pre-exist on the machines in the cluster; however, if they don't, you will need to use "-file" option to tell the framework to pack your executable files as a part of job submission. For example:

```
$HADOOP_HOME/bin/hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper myPythonScript.py \
    -reducer /bin/wc \
    -file myPythonScript.py
```

The above example specifies a user defined Python executable as the mapper. The option "-file myPythonScript.py" causes the python executable shipped to the cluster machines as a part of job submission.

In addition to executable files, you can also package other auxiliary files (such as dictionaries, configuration files, etc) that may be used by the mapper and/or the reducer. For example:

```
$HADOOP_HOME/bin/hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper myPythonScript.py \
    -reducer /bin/wc \
    -file myPythonScript.py \
    -file myDictionary.txt
```

Passing files to a Mapper /Reducer: -file (executable files) versus -files (data files) -files

-files

Working with Large Files and Archives

The -files and -archives options allow you to make files and archives available to the tasks. The argument is a URI to the file or archive that you have already uploaded to HDFS. These files and archives are cached across jobs. You can retrieve the host and fs_port values from the fs.default.name config variable.

Note: The -files and -archives options are generic options. Be sure to place the generic options before the command options, otherwise the command will fail. For an example, see The -archives Option. Also see Other Supported Options.

Making Files Available to Tasks

The -files option creates a symlink in the current working directory of the tasks that points to the local copy of the file.

In this example, Hadoop automatically creates a symlink named testfile.txt in the current working directory of the tasks. This symlink points to the local copy of testfile.txt.

```
-files hdfs://host:fs_port/user/testfile.txt
```

User can specify a different symlink name for -files using #.

```
-files hdfs://host:fs_port/user/testfile.txt#testfile
```

Multiple entries can be specified like this:

```
-files hdfs://host:fs_port/user/testfile1.txt,hdfs://host:fs_port/user/testfile2.txt
```

In this example, Hadoop automatically creates a symlink named testfile.txt in the current working directory of the tasks. This symlink points to the local copy of testfile.txt.

Naïve Bayes: Scenario 2: 10 mappers and 3 reducer

- **Assume we have 10 mappers and 3 reducer**
  - Mapper: emits counts for words, classes
  - Reducer: aggregates to yield class conditional probs and priors
    - Smoothing

How do we solve this?

Class conditional probabilites

Pr(w1|SPAM) = 2/4
Pr(w2|SPAM) = 1/4
Pr(w3|SPAM) = 1/4
Pr(w4|SPAM) = 0
Pr(w1|HAM) = 2/6
Pr(w2|HAM) = 0
Pr(w3|HAM) = 2/6
Pr(w4|HAM) = 2/6

Pr(SPAM)=3/6
Pr(HAM)= 3/6

Pr(w1|SPAM) = Count(w1 in SPAM documents)/(Total WordCount in SPAM class)

Zero Reducers

16

I am just trying to confirm my understanding of difference between 0 reducer and identity reducer.

- 0 reducer means reduce step will be skipped and mapper output will be the final out
- Identity reducer means then shuffling/sorting will still take place?

hadoop mapreduce

7

share improve this question

asked May 17 '12 at 5:44

kee

1,740 ● 4 ● 23 ● 53

add a comment

Use comments to ask for more information or suggest improvements. Avoid answering questions in comments.

4 Answers

active oldest votes

21

You understanding is correct. I would define it as following: If you do not need sorting of map results - you set 0 reduced,and the job is called map only.
If you need to sort the mapping results, but do not need any aggregation - you choose identity reducer.
And to complete the picture we have a third case : we do need aggregation and, in this case we need reducer.

APPENDIX C

Preparing the NCDC Weather Data

Zero Reducers Example [Hadoop The definitive Guide, Oreilly Book, 4th edition 2015]

This appendix gives a runthrough of the steps taken to prepare the raw weather datafiles so they are in a form that is amenable for analysis using Hadoop. If you want to get a copy of the data to process using Hadoop, you can do so by following the instructions given at the website that accompanies this book at *http://www.hadoopbook.com/*. The rest of this appendix explains how the raw weather datafiles were processed.

The raw data is provided as a collection of *tar* files, compressed with *bzip2*. Each year of readings comes in a separate file. Here's a partial directory listing of the files:

```
1901.tar.bz2
1902.tar.bz2
1903.tar.bz2
...
2000.tar.bz2
```

Each *tar* file contains a file for each weather station's readings for the year, compressed with *gzip*. (The fact that the files in the archive are compressed makes the *bzip2* compression on the archive itself redundant.) For example:

```
% tar jxf 1901.tar.bz2
% ls 1901 | head
029070-99999-1901.gz
029500-99999-1901.gz
029600-99999-1901.gz
029720-99999-1901.gz
029810-99999-1901.gz
227070-99999-1901.gz
```

Because there are tens of thousands of weather stations, the whole dataset is made up of a large number of relatively small files. It's generally easier and more efficient to process a smaller number of relatively large files in Hadoop (see "Small files and CombineFileInputFormat" on page 228), so in this case, I concatenated the decompressed files for a whole year into a single file, named by the year. I did this using a MapReduce program, to take advantage of its parallel processing capabilities. Let's take a closer look at the program.

The program has only a map function. No reduce function is needed because the map does all the file processing in parallel with no combine stage. The processing can be done with a Unix script, so the Streaming interface to MapReduce is appropriate in this case; see Example C-1.

Example C-1. Bash script to process raw NCDC datafiles and store in HDFS

```
#!/usr/bin/env bash

# NLineInputFormat gives a single line: key is offset, value is S3 URI
read offset s3file

# Retrieve file from S3 to local disk
echo "reporter:status:Retrieving $s3file" >&2
$HADOOP_HOME/bin/hadoop fs -get $s3file .

# Un-bzip and un-tar the local file
target=`basename $s3file .tar.bz2`
mkdir -p $target
echo "reporter:status:Un-tarring $s3file to $target" >&2
tar jxf `basename $s3file` -C $target

# Un-gzip each station file and concat into one file
echo "reporter:status:Un-gzipping $target" >&2
for file in $target/*/*
do
  gunzip -c $file >> $target.all
  echo "reporter:status:Processed $file" >&2
done

# Put gzipped version into HDFS
echo "reporter:status:Gzipping $target and putting in HDFS" >&2
gzip -c $target.all | $HADOOP_HOME/bin/hadoop fs -put - gz/$target.gz
```

The input is a small text file (*ncdc_files.txt*) listing all the files to be processed (the files start out on S3, so the files are referenced using S3 URIs that Hadoop understands). Here is a sample:

```
s3n://hadoopbook/ncdc/raw/isd-1901.tar.bz2
s3n://hadoopbook/ncdc/raw/isd-1902.tar.bz2
...
s3n://hadoopbook/ncdc/raw/isd-2000.tar.bz2
```

By specifying the input format to be `NLineInputFormat`, each mapper receives one line of input, which contains the file it has to process. The processing is explained in the script, but briefly, it unpacks the *bzip2* file and then concatenates each station file into a single file for the whole year. Finally, the file is gzipped and copied into HDFS. Note the use of `hadoop fs -put -` to consume from standard input.

APPENDIX C

Preparing the NCDC Weather Data

Zero Reducers Example [Hadoop The definitive Guide, Oreilly Book, 4th edition 2015]

This app
so they a
copy of t
given at
rest of th

The raw
of readin

```
1901.
1902.
1903.
...
2000.
```

Each *tar*
with *gzip*. (The fact that the files in the archive are compressed makes the *bzip2* compression on the archive itself redundant.) For example:

```
% tar
% ls
02907
02956
02966
02972
02981
22707
```

Because there are tens of thousands of weather stations, the whole dataset is made up of a large number of relatively small files. It's generally easier and more efficient to process a smaller number of relatively large files in Hadoop (see "Small files and CombineFileInputFormat" on page 228), so in this case, I concatenated the decompressed files for a whole year into a single file, named by the year. I did this using a MapReduce program, to take advantage of its parallel processing capabilities. Let's take a closer look at the program.

The program has only a map function. No reduce function is needed because the map does all the file processing in parallel with no combine stage. The processing can be done with a Unix script, so the Streaming interface to MapReduce is appropriate in this case; see Example C-1.

Example C-1. Bash script to process raw NCDC datafiles and store in HDFS

```
#!/usr/bin/env bash

# NLineInputFormat gives a single line: key is offset, value is S3 URI
read offset s3file

# Retrieve file from S3 to local disk
echo "reporter:status:Retrieving $s3file" >&2
$HADOOP_HOME/bin/hadoop fs -get $s3file .

# Un-bzip and un-tar the local file
target=`basename $s3file .tar.bz2`
mkdir -p $target
echo "reporter:status:Un-tarring $s3file to $target" >&2
tar jxf `basename $s3file` -C $target

# Un-gzip each station file and concat into one file
echo "reporter:status:Un-gzipping $target" >&2
for file in $target/*/*
do
  gunzip -c $file >> $target.all
  echo "reporter:status:Processed $file" >&2
done

# Put gzipped version into HDFS
echo "reporter:status:Gzipping $target and putting in HDFS" >&2
gzip -c $target.all | $HADOOP_HOME/bin/hadoop fs -put - gz/$target.gz
```

The input is a small text file (*ncdc_files.txt*) listing all the files to be processed (the files start out on S3, so the files are referenced using S3 URIs that Hadoop understands). Here is a sample:

```
s3n://hadoopbook/ncdc/raw/isd-1901.tar.bz2
s3n://hadoopbook/ncdc/raw/isd-1902.tar.bz2
...
s3n://hadoopbook/ncdc/raw/isd-2000.tar.bz2
```

By specifying the input format to be `NLineInputFormat`, each mapper receives one line of input, which contains the file it has to process. The processing is explained in the script, but briefly, it unpacks the *bzip2* file and then concatenates each station file into a single file for the whole year. Finally, the file is gzipped and copied into HDFS. Note the use of `hadoop fs -put -` to consume from standard input.

The program has only a map function. No reduce function is needed because the map does all the file processing in parallel with no combine stage. The processing can be done with a Unix script, so the Streaming interface to MapReduce is appropriate in this
case

By specifying the input format to be NLineInputFormat, each mapper receives one line of input, which contains the file it has to process

Status messages are echoed to standard error with a `reporter:status` prefix so that they get interpreted as a MapReduce status update. This tells Hadoop that the script is making progress and is not hanging.

The script to run the Streaming job is as follows:

```
% hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
  -D mapred.reduce.tasks=0 \
  -D mapred.map.tasks.speculative.execution=false \
  -D mapred.task.timeout=12000000 \
  -input ncdc_files.txt \
  -inputformat org.apache.hadoop.mapred.lib.NLineInputFormat \
  -output output \
  -mapper load_ncdc_map.sh \
  -file load_ncdc_map.sh
```

I set the number of reduce tasks to zero, since this is a map-only job. I also turned off speculative execution so duplicate tasks wouldn't write the same files (although the approach discussed in "Task side-effect files" on page 209 would have worked, too). The task timeout was set to a high value so that Hadoop doesn't kill tasks that are taking a long time (for example, when unarchiving files or copying to HDFS, when no progress is reported).

Finally, the files were archived on S3 by copying them from HDFS using *distcp*.

- End of Lecture