FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING

CHRISTOPHER D. MANNING AND
HINRICH SCHÜTZE

2 *Mathematical Foundations*

THIS CHAPTER presents some introductory material on probability and information theory, while the next chapter presents some essential knowledge of linguistics. A thorough knowledge of one or more of the fields of probability and statistics, information theory, and linguistics is desirable, and perhaps even necessary, for doing original research in the field of Statistical **NLP.** We cannot provide a thorough well-motivated introduction to each of these three fields within this book, but nevertheless, we attempt to summarize enough material to allow understanding of everything that follows in the book. We do however assume knowledge of parsing, either from a computer science or computational linguistics perspective. We also assume a reasonable knowledge of mathematical symbols and techniques, perhaps roughly to the level of a first year undergraduate course, including the basics of such topics as: set theory, functions and relations, summations, polynomials, calculus, vectors and matrices, and logarithms. Mathematical notations that we use are summarized in the Table of Notations.

If you are familiar with one of the areas covered in these two chapters, then you should probably just skim the corresponding section. If you're not familiar with a topic, we think it is probably best to try to read through each section, but you will probably need to reread sections when the techniques in them are put to use. These chapters don't say much about applications – they present the preparatory theory for what follows.

2.1 Elementary Probability Theory

This section sketches the essentials of probability theory necessary to understand the rest of this book.

2.1.1 Probability spaces

PROBABILITY THEORY

Probability theory deals with predicting how likely it is that something will happen. For example, if one tosses three coins, how likely is it that they will all come up heads? Although our eventual aim is to look at language, we begin with some examples with coins and dice, since their behavior is simpler and more straightforward.

EXPERIMENT
TRIAL

The notion of the likelihood of something is formalized through the concept of an *experiment* (or trial) – the process by which an observation is made. In this technical sense, tossing three coins is an experiment. All that is crucial is that the experimental protocol is well defined. We

BASIC OUTCOMES
SAMPLE SPACE
DISCRETE
CONTINUOUS

assume a collection of *basic outcomes* (or sample points) for our experiment, the sample space Ω. Sample spaces may either be *discrete,* having at most a countably infinite number of basic outcomes, or *continuous,* having an uncountable number of basic outcomes (for example, measuring a person's height). For language applications and in this introduction, we will mainly deal with discrete sample spaces which only contain a finite

EVENT

number of basic outcomes. Let an event A be a subset of Ω. For example, in the coin experiment, the first coin being a head, and the second and third coming down tails is one basic outcome, while any result of one head and two tails is an example of an event. Note also that Ω represents the certain event, the space of all possible experimental outcomes, and 0 represents the impossible event. We say that an experimental outcome must be an event. The foundations of probability theory depend on the

σ-FIELD

set of events $\mathcal{F}$ forming a *σ-field* – a set with a maximal element Ω and arbitrary complements and unions. These requirements are trivially satisfied by making the set of events, the event *space,*

EVENT SPACE

the power set of the sample space (that is, the set of all subsets of the sample space, often written $2^{\mathcal{F}}$).

PROBABILITY
FUNCTION
PROBABILITY
DISTRIBUTION

Probabilities are numbers between O and 1, where O indicates impossibility and 1 certainty. A *probability function* (also known as a *probability distribution)* distributes a probability mass of 1 throughout the sample space Ω. Formally, a discrete probability function is any function $P: \mathcal{F} \rightarrow [0, 1]$ such that:

- $P(\Omega) = 1$

DISJOINT

- Countable additivity: For *disjoint* sets $A_j \in \mathcal{F}$ (i.e., $A_j \cap A_k = \emptyset$ for $j \neq k$)

$$P\left(\bigcup_{j=1}^{\infty} A_j\right) = \sum_{j=1}^{\infty} P(A_j) \tag{2.1}$$

We call $P(A)$ the probability of the event A. These axioms say that an event that encompasses, say, three distinct possibilities must have a probability that is the sum of the probabilities of each possibility, and that since an experiment must have some basic outcome as its result, the probability of that is 1. Using basic set theory, we can derive from these axioms a set of further properties of probability functions; see exercise 2.1.

PROBABILITY SPACE

A well-founded *probability space* consists of a sample space Ω, a a-field of events $\mathcal{F}$, and a probability function P. In Statistical NLP applications, we always seek to properly define such a probability space for our models. Otherwise, the numbers we use are merely ad hoc scaling factors, and there is no mathematical theory to help us. In practice, though, corners often have been, and continue to be, cut.

Example 1: A fair coin is tossed 3 times. What is the chance of 2 heads'?

Solution: The experimental protocol is clear. The sample space is:

$$\Omega = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$$

Each of the basic outcomes in Ω is equally likely, and thus has probability 1/8. A situation where each basic outcome is equally likely is called a uniform *distribution*. In a finite sample space with equiprobable basic outcomes, $P(A) = \frac{|A|}{|\Omega|}$ (where $|A|$ is the number of elements in a set A).

UNIFORM DISTRIBUTION

The event of interest is:

$$A = \{HHT, HTH, THH\}$$

So:

$$P(A) = \frac{|A|}{|\Omega|} = \frac{3}{8}$$

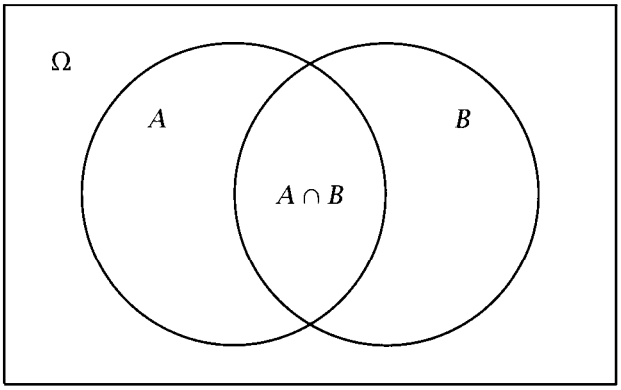

Figure 2.1 A diagram illustrating the calculation of conditional probability $P(A|B)$. Once we know that the outcome is in B, the probability of A becomes $P(A \cap B)/P(B)$.

2.1.2 Conditional probability and independence

CONDITIONAL PROBABILITY

PRIOR PROBABILITY

POSTERIOR PROBABILITY

Sometimes we have partial knowledge about the outcome of an experiment and that naturally influences what experimental outcomes are possible. We capture this knowledge through the notion of *conditional probability*. This is the updated probability of an event given some knowledge. The probability of an event before we consider our additional knowledge is called the *prior probability* of the event, while the new probability that results from using our additional knowledge is referred to as the *posterior probability* of the event. Returning to example 1 (the chance of getting 2 heads when tossing 3 coins), if the first coin has been tossed and is a head, then of the 4 remaining possible basic outcomes, 2 result in 2 heads, and so the probability of getting 2 heads now becomes $\frac{1}{2}$. The conditional probability of an event A given that an event B has occurred $(P(B) > 0)$ is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.2}$$

Even if P(B) = 0 we have that:

$$P(A \cap B) = P(B)P(A|B) = P(A)P(B|A) \tag{2.3}$$

[The multiplication rule]

We can do the conditionalization either way because set intersection is symmetric $(A \cap B = B \cap A)$. One can easily visualize this result by looking at the diagram in figure 2.1.

CHAIN RULE

The generalization of this rule to multiple events is a central result that will be used throughout this book, the *chain rule:*

$$P(A_1 \cap \ldots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \cdots P(A_n|\cap_{i=1}^{n-1} A_i) \tag{2.4}$$

▼ The chain rule is used in many places in Statistical NLP, such as working out the properties of Markov models in chapter 9.

INDEPENDENCE

Two events *A, B* are *independent* of each other if $P(A \cap B) = P(A)P(B)$. Unless P(B) = 0 this is equivalent to saying that $P(A) = P(A|B)$ (i.e., knowing that *B* is the case does not affect the probability of *A).* This equivalence follows trivially from the chain rule. Otherwise events are *dependent.* We can also say that *A* and *B* are *conditionally independent* given C when $P(A \cap B|C) = P(A|C)P(B|C)$.

DEPENDENCE
CONDITIONAL INDEPENDENCE

2.1.3 Bayes' theorem

BAYES' THEOREM

Bayes' theorem lets us swap the order of dependence between events. That is, it lets us calculate $P(B|A)$ in terms of $P(A|B)$. This is useful when the former quantity is difficult to determine. It is a central tool that we will use again and again, but it is a trivial consequence of the definition of conditional probability and the chain rule introduced in equations (2.2) and (2.3):

$$P(B|A) = \frac{P(B \cap A)}{P(A)} = \frac{P(A|B)P(B)}{P(A)} \tag{2.5}$$

NORMALIZING CONSTANT

The righthand side denominator *P(A)* can be viewed as a *normalizing* constant, something that ensures that we have a probability function. If we are simply interested in which event out of some set is most likely given *A, we* can ignore it. Since the denominator is the same in all cases, we have that:

$$\arg\max_B \frac{P(A|B)P(B)}{\mathrm{P(A)}} = \arg\max_B P(A|B)P(B) \tag{2.6}$$

However, we can also evaluate the denominator by recalling that:

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

So we have:

$$\begin{aligned} P(A) &= P(A \cap B) + P(A \cap \overline{B}) \qquad \text{[additivity]} \\ &= P(A|B)P(B) + P(A|\overline{B})P(\overline{B}) \end{aligned}$$

B and $\overline{B}$ serve to split the set A into two disjoint parts (one possibly empty), and so we can evaluate the conditional probability on each, and then sum, using additivity. More generally, if we have some group of sets B_i that ***partition A,*** that is, if $A \subseteq \cup_i B_i$ and the B_i are disjoint, then:

PARTITION

$$(2.7)\quad P(A) = \sum P(A|B_i)P(B_i)$$

This gives us the following equivalent but more elaborated version of Bayes' theorem:

Bayes' theorem: If $A \subseteq \cup_{i=1}^{n} B_i$, $P(A) > 0$, and $B_i \cap B_j = 0$ for $i \neq j$ then:

$$(2.8)\quad P(B_j|A) = \frac{P(A|B_j)P(B_j)}{P(A)} = \frac{P(A|B_j)P(B_j)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

Example 2: Suppose one is interested in a rare syntactic construction, perhaps parasitic gaps, which occurs on average once in 100,000 sentences. Joe Linguist has developed a complicated pattern matcher that attempts to identify sentences with parasitic gaps. It's pretty good, but it's not perfect: if a sentence has a parasitic gap, it will say so with probability 0.95, if it doesn't, it will wrongly say it does with probability 0.005. Suppose the test says that a sentence contains a parasitic gap. What is the probability that this is true?

Solution: Let G be the event of the sentence having a parasitic gap, and let T be the event of the test being positive. We want to determine:

$$P(G|T) = \frac{P(T|G)P(G)}{P(T|G)P(G) + P(T|\overline{G})P(\overline{G})}$$

$$= \frac{0.95 \times 0.00001}{0.95 \times 0.00001 + 0.005 \times 0.99999} \approx 0.002$$

Here we use having the construction or not as the partition in the denominator. Although Joe's test seems quite reliable, we find that using it won't help as much as one might have hoped. On average, only **1** in every 500 sentences that the test identifies will actually contain a parasitic gap. This poor result comes about because the prior probability of a sentence containing a parasitic gap is so low.

▼ Bayes' theorem is central to the noisy channel model described in section 22.4.

First die	Second die 1	2	3	4	5	6
6	7	8	9	IO	11	12
5	6	7	8	9	IO	11
4	5	6	7	8	9	IO
3	4	5	6	7	8	9
2	3	4	5	6	7	8
1	2	3	4	5	6	7

x	2	3	4	5	67	8	9	IO	1l1l	12
$p(X = x)$	1/36	1/18	1/12	1/9	5/36	15/36	1/9	1/12	1/18	1/36

Figure 2.2 A random variable X for the sum of two dice. Entries in the body of the table show the value of X given the underlying basic outcomes, while the bottom two rows show the pmf p(x).

2.1.4 Random variables

RANDOMVARIABLE

A *random variable* is simply a function $X: \Omega \to \mathbb{R}^n$ (commonly with $n = 1$), where $\mathbb{R}$ is the set of real numbers. Rather than having to work with some irregular event space which differs with every problem we look at, a random variable allows us to talk about the probabilities of numerical values that are related to the event space. We think of an abstract *stochastic process* that generates numbers with a certain probability distribution. (The word ***stochastic*** simply means 'probabilistic' or 'randomly generated,' but is especially commonly used when referring to a sequence of results assumed to be generated by some underlying probability distribution.)

STOCHASTIC PROCESS

INDICATORRANDOM VARIABLE
BERNOULLI TRIAL

A discrete random variable is a function $X: \Omega \to S$ where S is a countable subset of $\mathbb{R}$. If $X: \Omega \to \{0,1\}$, then X is called an *indicator random variable* or a ***Bernoulli trial.***

Example 3: Suppose the events are those that result from tossing two dice. Then we could define a discrete random variable X that is the sum of their faces: $S = \{2, \ldots, 12\}$, as indicated in figure 2.2.

PROBABILITY MASS FUNCTION

Because a random variable has a numeric range, we can often do mathematics more easily by working with the values of a random variable, rather than directly with events. In particular we can define the *probability mass function* (pmf) for a random variable X, which gives the proba-

bility that the random variable has different numeric values:

(2.9) pmf $p(x) = p(X = x) = P(A_x)$ where $A_x = \{\omega \in \Omega : X(\omega) = x\}$

We will write pmfs with a lowercase roman letter (even when they are variables). If a random variable X is distributed according to the pmf p(x), then we will write $X \sim p(x)$.

Note that $p(x) > 0$ at only a countable number of points (to satisfy the stochastic constraint on probabilities), say $\{x_i : i \in \mathbb{N}\}$, while $p(x) = 0$ elsewhere. For a discrete random variable, we have that:

$$\sum_i p(x_i) = \sum_i P(A_{x_i}) = P(\Omega) = 1$$

Conversely, any function satisfying these constraints can be regarded as a mass function.

▼ Random variables are used throughout the introduction to information theory in section 2.2.

2.1.5 Expectation and variance

EXPECTATION
MEAN

The *expectation* is the mean or average of a random variable.

If X is a random variable with a pmf p(x) such that $\sum_x |x|\, p(x) < \infty$ then the expectation is:

(2.10) $E(X) = \sum_x x\, p(x)$

Example 4: If rolling one die and Y is the value on its face, then:

$$E(Y) = \sum_{y=1}^{6} y\, p(y) = \frac{1}{6}\sum_{y=1}^{6} y = \frac{21}{6} = 3\frac{1}{2}$$

This is the expected average found by totaling up a large number of throws of the die, and dividing by the number of throws.

If $Y \sim p(y)$ is a random variable, any function g(Y) defines a new random variable. If $E(g(Y))$ is defined, then:

(2.11) $E(g(Y)) = \sum_Y g(y)\, p(y)$

For instance, by letting g be a linear function $g(Y) = aY + b$, we see that $E(g(Y)) = aE(Y) + b$. We also have that $E(X + Y) = E(X) + E(Y)$ and if X and Y are independent, then $E(XY) = E(X)E(Y)$.

VARIANCE

The variance of a random variable is a measure of whether the values of the random variable tend to be consistent over trials or to vary a lot. One measures it by finding out how much on average the variable's values deviate from the variable's expectation:

$$\begin{aligned} \text{Var}(X) &= E((X - E(X))^2) \\ &= E(X^2) - E^2(X) \end{aligned} \tag{2.12}$$

STANDARD DEVIATION

The commonly used ***standard deviation*** of a variable is the square root of the variance. When talking about a particular distribution or set of data, the mean is commonly denoted as μ, the variance as σ^2, and the standard deviation is hence written as σ.

Example 5: What is the expectation and variance for the random variable introduced in example 3, the sum of the numbers on two dice?

Solution: For the expectation, we can use the result in example 4, and the formula for combining expectations in (or below) equation (2.11):

$$E(X) = E(Y + Y) = E(Y) + E(Y) = 3\frac{1}{2} + 3\frac{1}{2} = 7$$

The variance is given by:

$$\text{Var}(X) = E((X - E(X))^2) = \sum_x \text{p}(x)(x - E(X))^2 = 5\frac{5}{6}$$

Because the results for rolling two dice are concentrated around 7, the variance of this distribution is less than for an 'll-sided die,' which returns a uniform distribution over the numbers 2-12. For such a uniformly distributed random variable U, we find that $\text{Var}(U) = 10$.

▼ Calculating expectations is central to Information Theory, as we will see in section 2.2. Variances are used in section 5.2.

2.1.6 Notation

In these sections, we have distinguished between P as a probability function and p as the probability mass function of a random variable. However, the notations $P(\cdot)$ and $\text{p}(\cdot)$ do not always refer to the same function. Any time that we are talking about a different probability space, then we are talking about a different function. Sometimes we will denote these

different functions with subscripts on the function to make it clear what we are talking about, but in general people just write *P* and rely on context and the names of the variables that are arguments to the function to disambiguate. It is important to realize that one equation is often referring to several different probability functions, all ambiguously referred to as *P*.

2.1.7 Joint and conditional distributions

Often we define many random variables over a sample space giving us a joint (or multivariate) probability distribution. The joint probability mass function for two discrete random variables X, Y is:

$$p(x, y) = P(X = x, Y = y)$$

MARGINAL DISTRIBUTION

Related to a joint pmf are marginal *pmfs*, which total up the probability masses for the values of each variable separately:

$$p_X(x) = \sum_Y p(x, y) \qquad p_Y(y) = \sum_X p(x, y)$$

In general the marginal mass functions do not determine the joint mass function. But if X and Y are independent, then $p(x, y) = p_X(x)\, p_Y(y)$. For example, for the probability of getting two sixes from rolling two dice, since these events are independent, we can compute that:

$$p(Y = 6, Z = 6) = p(Y = 6)\, p(Z = 6) = \frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$$

There are analogous results for joint distributions and probabilities for the intersection of events. So we can define a conditional pmf in terms of the joint distribution:

$$p_{X|Y}(x|y) = \frac{p(x, y)}{p_Y(y)} \quad \text{for y such that } p_Y(y) > 0$$

and deduce a chain rule in terms of random variables, for instance:

$$p(w, x, y, z) = P(w)P(x|w)P(y|w, x)\, p(z|w, x, y)$$

2.1.8 Determining *P*

So far we have just been assuming a probability function *P* and giving it the obvious definition for simple examples with coins and dice. But what

do we do when dealing with language? What do we say about the probability of a sentence like *The cow chewed its cud?* In general, for language events, unlike dice, P is unknown. This means we have to *estimate* P. We do this by looking at evidence about what P must be like based on a sample of data. The proportion of times a certain outcome occurs is called the *relative frequency* of the outcome. If $C(u)$ is the number of times an outcome u occurs in N trials then $\frac{C(u)}{N}$ is the relative frequency of u. The relative frequency is often denoted f_u. Empirically, if one performs a large number of trials, the relative frequency tends to stabilize around some number. That this number exists provides a basis for letting us calculate probability estimates.

ESTIMATION

RELATIVE FREQUENCY

Techniques for how this can be done are a major topic of this book, particularly covered in chapter 6. Common to most of these techniques is to estimate P by assuming that some phenomenon in language is acceptably modeled by one of the well-known families of distributions (such as the binomial or normal distribution), which have been widely studied in statistics. In particular a binomial distribution can sometimes be used as an acceptable model of linguistic events. We introduce a couple of families of distributions in the next subsection. This is referred to as a *parametric* approach and has a couple of advantages. It means we have an explicit probabilistic model of the process by which the data was generated, and determining a particular probability distribution within the family only requires the specification of a few parameters, since most of the nature of the curve is fixed in advance. Since only a few parameters need to be determined, the amount of training data required is not great, and one can calculate how much training data is sufficient to make good probability estimates.

PARAMETRIC

But, some parts of language (such as the distributions of words in newspaper articles in a particular topic category) are irregular enough that this approach can run into problems. For example, if we assume our data is binomially distributed, but in fact the data looks nothing like a binomial distribution, then our probability estimates might be wildly wrong.

For such cases, one can use methods that make no assumptions about the underlying distribution of the data, or will work reasonably well for a wide variety of different distributions. This is referred to as a *non-parametric* or *distribution-free* approach. If we simply empirically estimate P by counting a large number of random events (giving us a discrete distribution, though we might produce a continuous distribution from

NON-PARAMETRIC

DISTRIBUTION-FREE

such data by interpolation, assuming only that the estimated probability density function should be a fairly smooth curve), then this is a nonparametric method. However, empirical counts often need to be modified or smoothed to deal with the deficiencies of our limited training data, a topic discussed in chapter 6. Such smoothing techniques usually assume a certain underlying distribution, and so we are then back in the world of parametric methods. The disadvantage of nonparametric methods is that we give our system less prior information about how the data are generated, so a great deal of training data is usually needed to compensate for this.

▼ Non-parametric methods are used in automatic classification when the underlying distribution of the data is unknown. One such method, nearest neighbor classification, is introduced in section 16.4 for text categorization.

2.1.9 Standard distributions

Certain probability mass functions crop up commonly in practice. In particular, one commonly finds the same basic form of a function, but just with different constants employed. Statisticians have long studied these families of functions. They refer to the family of functions as a distribution and to the numbers that define the different members of the family as *parameters*. Parameters are constants when one is talking about a particular pmf, but variables when one is looking at the family. When writing out the arguments of a distribution, it is usual to separate the random variable arguments from the parameters with a semicolon (;). In this section, we just briefly introduce the idea of distributions with one example each of a discrete distribution (the binomial distribution), and a continuous distribution (the normal distribution).

DISTRIBUTION

PARAMETEKS

Discrete distributions: The binomial distribution

BINOMIAL DISTRIBUTION

A *binomial distribution* results when one has a series of trials with only two outcomes (i.e., Bernoulli trials), each trial being independent from all the others. Repeatedly tossing a (possibly unfair) coin is the prototypical example of something with a binomial distribution. Now when looking at linguistic corpora, it is never the case that the next sentence is truly independent of the previous one, so use of a binomial distribution is always an approximation. Nevertheless, for many purposes, the dependency be-

tween words falls off fairly quickly and we can assume independence. In any situation where one is counting whether something is present or absent, or has a certain property or not, and one is ignoring the possibility of dependencies between one trial and the next, one is at least implicitly using a binomial distribution, so this distribution actually crops up quite commonly in Statistical NLP applications. Examples include: looking through a corpus to find an estimate of the percent of sentences in English that have the word the in them or finding out how commonly a verb is used transitively by looking through a corpus for instances of a certain verb and noting whether each use is transitive or not.

The family of binomial distributions gives the number r of successes out of n trials given that the probability of success in any trial is p:

$$b(r; n, p) = \binom{n}{r} p^r (1-p)^{n-r} \quad \text{where} \quad \binom{n}{r} = \frac{n!}{(n-r)!r!} \quad 0 \leq r \leq n \tag{2.13}$$

The term $\binom{n}{r}$ counts the number of different possibilities for choosing r objects out of n, not considering the order in which they are chosen. Examples of some binomial distributions are shown in figure 2.3. The binomial distribution has an expectation of np and a variance of $np(1-p)$.

Example 6: Let R have as value the number of heads in n tosses of a (possibly weighted) coin, where the probability of a head is p.

Then we have the binomial distribution:

$$p(R = r) = b(r; n, p)$$

(The proof of this is by counting: each basic outcome with r heads and $n-r$ tails has probability $h^r(1-h)^{n-r}$, and there are $\binom{n}{r}$ of them.)

▼ The binomial distribution turns up in various places in the book, such as when counting n-grams in chapter 6, and for hypothesis testing in section 8.2.

▼ The generalization of a binomial trial to the case where each of the trials has more than two basic outcomes is called a multinomial experiment, and is modeled by the *multinomial distribution.* A zcroth order n-gram model of the type we discuss in chapter 6 is a straightforward example of a multinomial distribution.

MULTINOMIAL DISTRIBUTION

▼ Another discrete distribution that we discuss and use in this book is the Poisson distribution (section 15.3.1). Section 5.3 discusses the Bernoulli distribution, which is simply the special case of the binomial distribution where there is only one trial. That is, we calculate $b(r; 1, p)$.

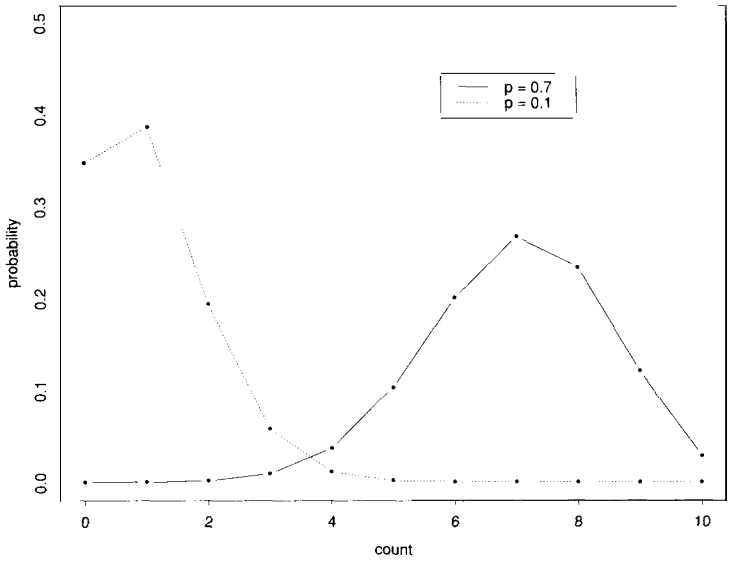

Figure 2.3 Two examples of binomial distributions: $b(r;10,0.7)$ and $b(r;10,0.1)$.

Continuous distributions: The normal distribution

So far we have looked only at discrete probability distributions and discrete random variables, but many things, such as measurements of heights and lengths, are best understood as having a continuous domain, over the real numbers $\mathbb{R}$. In this book, we do not outline the mathematics of continuous distributions. Suffice it to say that there are generally analogous results, except with points becoming intervals, and sums becoming integrals. However, we will occasionally have need to refer to continuous probability distributions, so we will give one example here: the normal distribution, which is central to all work in probability and statistics.

BELL CURVE
NORMAL DISTRIBUTION

For many things in the world, such as the heights or IQs of people, one gets a distribution that is known in the media as a bell curve, but which is referred to in statistics as a normal *distribution.* Some normal distribution curves are shown in figure 2.4. The values of the graphed functions, ***probability density functions (pdf), do not directly give the*** probabilities of the points along the x-axis (indeed, the probability of a point is always 0 for a continuous distribution). Rather the probability

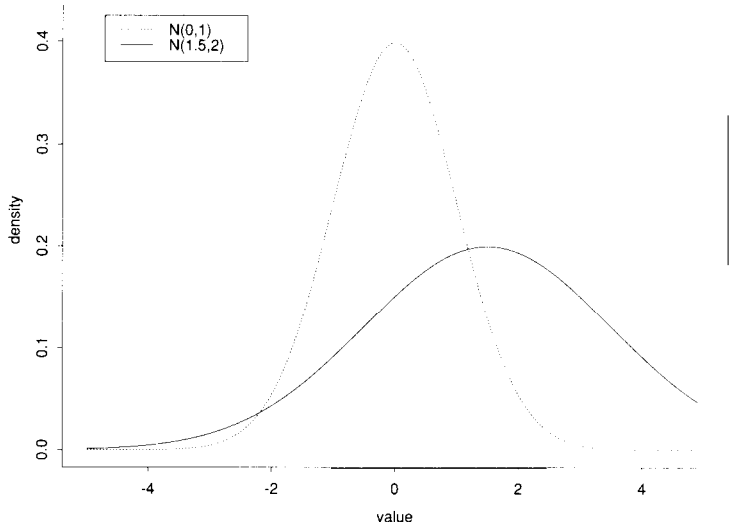

Figure 2.4 Example normal distribution curves: $\mathrm{n}(x;0,1)$ and $\mathrm{n}(x;1.5,2)$.

of a result within a certain interval on the x-axis is given by the area delimited by that region, the x-axis and the function curve.

The normal distribution has two parameters for the mean μ, and the standard deviation σ, and the curve is given by:

$$\mathrm{n}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)} \tag{2.14}$$

STANDARD NORMAL DISTRIBUTION

The curve where $\mu = 0$ and $\sigma = 1$ is referred to as the standard ***normal distribution***. A few figures for areas under this curve are given in the appendix.

While it is much better to refer to such a curve as a 'normal distribution' than as a 'bell curve,' if you really want to fit into the Statistical NLP or pattern recognition communities, you should instead learn to refer to these functions as *Gaussians*, and to remark things like, 'Maybe we could model that using 3 Gaussians' at appropriate moments.[1]

GAUSSIANS

1. Carl Friedrich Gauss was the first to use normal curves to model experimental data, using them to model the errors made by astronomers and surveyors in repeated measurements of the same quantity, but the normal curve was discovered by Abraham de Moivre.

In much of statistics, the discrete binomial distribution is approximated by the continuous normal distribution – one can see the basic similarity in the shapes of the curves by comparing figures 2.3 and 2.4. Such an approximation is acceptable when both basic outcomes have a reasonable probability of occurring or the amount of data is very large (roughly, when $np(1-p) > 5$). But, in natural language, events like occurrences of the phrase *shade tree mechanics* are so rare, that even if you have a huge amount of text, there will be a significant difference between the appropriate binomial curve and the approximating normal curve, and so use of normal approximations can be unwise.

▼ Gaussians are often used in clustering, as discussed in chapter 14. In particular, here we have only discussed the one-dimensional or univariate normal distribution, while we present there the generalization to many dimensions (the multivariate normal distribution).

▼ Other continuous distributions discussed in this book are the hyperbolic distributions discussed in section 1.4.3, and the t distribution used for hypothesis testing in section 5.3.

2.1.10 Bayesian statistics

FREQUENTIST STATISTICS

BAYESIAN STATISTICS

So far, we have presented a brief introduction to orthodox *frequentist statistics.* Not everyone is agreed on the right philosophical foundations for statistics, and the main rival is a Bayesian approach to statistics. Actually, the Bayesians even argue among themselves, but we are not going to dwell on the philosophical issues here. We want to just briefly introduce the Bayesian approach because Bayesian methods are very useful in Statistical NLP, and we will come across them in later chapters.

Bayesian updating

MAXIMUM LIKELIHOOD ESTIMATE

PRIOR BELIEF

Suppose one takes a coin and tosses it 10 times, and gets 8 heads. Then from a frequentist point of view, the result is that this coin comes down heads 8 times out of 10. This is what is called the maximum likelihood estimate, as discussed further in section 62.1. However, if one has looked the coin over, and there doesn't seem anything wrong with it, one would be very reluctant to accept this estimate. Rather, one would tend to think that the coin would come down equally head and tails over the long run, and getting 8 heads out of 10 is just the kind of thing that happens sometimes given a small sample. In other words one has a *prior belief* that

influences one's beliefs even in the face of apparent evidence against it. Bayesian statistics measure degrees of belief, and are calculated by starting with prior beliefs and updating them in the face of evidence, by use of Bayes' theorem.

For example, let μ_m be the model[2] that asserts $P(\text{head}) = m$. Let s be a particular sequence of observations yielding i heads and j tails. Then, for any $m, 0 \leq m \leq 1$:

$$P(s|\mu_m) = m^i(1-m)^j \tag{2.15}$$

From a frequentist point of view, we wish to find the MLE:

$$\arg\max_m P(s|\mu_m)$$

To do this, we can differentiate the above polynomial, and find its maximum, which fortunately gives the intuitive answer of $\frac{i}{i+j}$, or 0.8 for the case of 8 heads and 2 tails.

But now suppose that one wants to quantify one's belief that the coin is probably a regular, fair one. One can do that by assuming a prior probability distribution over how likely it is that different models μ_m are true. Since one would want most of the probability mass close to $\frac{1}{2}$, one might use something like a Gaussian distribution centered on $\frac{1}{2}$, but since polynomials are the only things we can remember how to differentiate, let us instead assume that one's prior belief is modeled by the distribution:

$$P(\mu_m) = 6m(1-m) \tag{2.16}$$

This polynomial was chosen because its distribution is centered on $\frac{1}{2}$, and, conveniently, the area under the curve between 0 and 1 is 1.

When one sees an observation sequence s one wants to know one's new belief in the fairness of the coin. One can calculate this from (2.15) and (2.16) by Bayes' theorem:

$$P(\mu_m|s) = \frac{P(s|\mu_m)P(\mu_m)}{P(s)} \tag{2.17}$$

$$= \frac{m^i(1-m)^j \times 6m(1-m)}{P(s)}$$

2. By a model we mean whatever theoretical edifices we construct to explain something in the world. A probabilistic model might comprise the specification of a distribution and certain parameter values. Thus, we are introducing some notational sloppiness in equation (2.15), since previously we were conditioning on an event, that is, a subset of the event space, and now we are conditioning on a model, but we will allow ourselves that freedom.

$$= \frac{6m^{i+1}(1-m)^{j+1}}{P(s)}$$

Now $P(s)$ is the prior probability of s. Let us assume for the moment that it does not depend on μ_m, and therefore that we can ignore it while finding the m that maximizes this equation. If we then differentiate the numerator so as find its maximum, we can determine that for the case of 8 heads and 2 tails:

$$\arg\max_m P(\mu_m|s) = \frac{3}{4}$$

Because our prior was weak (the polynomial is a quite flat curve centered over $\frac{1}{2}$), we have moved a long way in the direction of believing that the coin is biased, but the important point is that we haven't moved all the way to 0.8. If we had assumed a stronger prior, we would have moved a smaller distance from $\frac{1}{2}$. (See exercise 2.8.)

But what do we make of the denominator P(s)? Well, since we have just seen s, one might conclude that this is **1,** but that is not what it means. Rather, it is the marginal *probability* which is obtained by adding up all the $P(s|\mu_m)$ weighted by the probability of μ_m, as we saw earlier in equation (2.8). For the continuous case, we have the integral:

MARGINAL PROBABILITY

(2.18)
$$P(s) = \int_0^1 P(s|\mu_m)P(\mu_m)dm$$
$$= \int_0^1 6m^{i+1}(1-m)^{j+1}dm$$

This just happens to be an instance of the beta integral, another continuous distribution well-studied by statisticians, and so we can look up a book to find out that:

(2.19)
$$P(s) = \frac{6(i+1)!(j+1)!}{(i+j+3)!}$$

NORMALIZATION FACTOR

But the important point is that the denominator is just a *normalization factor*, which ensures that what we calculate for $P(\mu_m|s)$ in (2.17) is actually a probability function.

In the general case where data come in sequentially and we can reasonably assume independence between them, we start off with an a priori probability distribution, and when a new datum comes in, we can update our beliefs by calculating the maximum of the a posteriori distribution, what is sometimes referred to as the MAP probability. This then becomes the new prior, and the process repeats on each new datum. This process is referred to as *Bayesian updating*.

BAYESIAN UPDATING

Bayesian decision theory

But there is another thing that we can do with this new approach: use it to evaluate which model or family of models better explains some data. Suppose that we did not actually see the sequence of coin tosses but just heard the results shouted out over the fence. Now it may be the case, as we have assumed so far, that the results reported truly reflect the results of tossing a single, possibly weighted coin. This is the theory μ, which is a family of models, with a parameter representing the weighting of the coin. But an alternative theory is that at each step someone is tossing two fair coins, and calling out "tails" if *both* of them come down tails, and heads otherwise. Let us call this new theory ν. According to v, ifs is a particular observed sequence of i heads and j tails, then:

$$P(s|\nu) = \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^j \tag{2.20}$$

Note that one of these theories has a free parameter (the weighting of the coin m), while the other has no parameters. Let us assume that, a priori, both of these theories are equally likely, for instance:

$$P(\mu) = \mathrm{P(v)} = \frac{1}{2} \tag{2.21}$$

We can now attempt to work out which theory is more likely given the data we have seen. We use Bayes' theorem again, and write down:

$$P(\mu|s) = \frac{P(s|\mu)P(\mu)}{P(s)} \qquad P(\nu|s) = \frac{P(s|\nu)P(\nu)}{P(s)}$$

The potentially confusing point here is that we have made a quick change in our notation. The quantity we are now describing as $P(s|\mu)$ is the quantity that we wrote as just $P(s)$ in (2.19) – since at that time we were assuming that theory μ_m was true and we were just trying to determine m, whereas what we are now writing as $P(s)$ is the prior probability of s, not knowing whether μ is true or not. With that gotten straight, we can calculate the ***likelihood ratio*** between these two models. The $P(s)$ terms in the denominators cancel, and we can work out the rest using equations (2.19),(2.20), and (2.21):

LIKELIHOOD RATIO

$$\begin{aligned} \frac{P(\mu|s)}{P(\nu|s)} &= \frac{P(s|\mu)P(\mu)}{P(s|\nu)P(\nu)} \\ &= \frac{\frac{6(i+1)!(j+1)!}{(i+j+3)!}}{\left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^j} \end{aligned} \tag{2.22}$$

10 Results Reported			20 Results Reported		
Heads	Tails	Likelihood ratio	Heads	Tails	Likelihood ratio
0	10	4.03×10^4	0	20	1.30×10^{10}
1	9	2444.23	2	18	2.07×10^7
2	8	244.42	4	16	1.34×10^5
3	7	36.21	6	14	2307.06
4	6	7.54	8	12	87.89
5	5	2.16	10	10	6.89
6	4	0.84	12	8	1.09
7	3	0.45	14	6	0.35
8	2	0.36	16	4	0.25
9	1	0.37	18	2	0.48
10	0	0.68	20	0	3.74

Table 2.1 Likelihood ratios between two theories. The left three columns are for a sequence s of 10 pieces of data, and the right three columns for a sequence of 20 pieces of data.

If this ratio is greater than 1, we should prefer μ, and otherwise we should prefer ν (or commonly people take the log of this ratio and see if that value is greater than or less than zero).

We can calculate this ratio for different combinations of heads and tails. Table 2.1 shows likelihood values for sequences of 10 and 20 results. If there are few heads, then the likelihood ratio is greater than one, and the possibly weighted coin theory wins, since it is never strongly incompatible with any data (because of its free parameter). On the other hand, if the distribution is roughly what we'd expect according to the two fair coins theory (a lot more heads than tails) then the likelihood ratio is smaller than one, and the simpler two fair coins theory wins. As the quantity of data available becomes greater, the ratio of heads needs to be nearer $\frac{3}{4}$ in order for the two fair coins model to win. If these are the only two theories under consideration, and we choose the one that wins in such a likelihood ratio, then we have made what is called the *Bayes optimal decision.*

BAYESOPTIMAL DECISION

▼ If there are more theories, we can compare them all and decide on the most likely one in the same general manner. An example of this and more general discussion of Bayesian decision theory can be found in our discussion of word sense disambiguation in section 7.2.1.

21.11 Exercises

Exercise *2.1* [⋆]

This exercise indicates the kind of facility with set theory needed for this book, and summarizes a few useful results in probability theory. Use set theory and the axioms defining a probability function to show that:

a. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ [the addition rule]

b. P(0) = 0

c. $P(\overline{A})$ = 1 -P(A)

d. $A \subseteq B \Rightarrow P(A) \leq P(B)$

e. $P(B - A) = P(B) - P(A \cap B)$

Exercise *2.2* [⋆]

Assume the following sample space:

(2.23) Ω = {is-noun, has-plural-s, is-adjective, is-verb]

and the function $f : 2^{\Omega} \rightarrow [0, 1]$ with the following values:

x	$f(x)$
{ is-noun }	0.45
{ has-plural-s }	0.2
{ is-adjective }	0.25
{ is-verb }	0.3

Can f be extended to all of 2^{Ω} such that it is a well-formed probability distribution? If not, how would you model these data probabilistically?

Exercise 2.3 [⋆]

Compute the probability of the event 'A period occurs after a three-letter word and this period indicates an abbreviation (not an end-of-sentence marker),' assuming the following probabilities.

(2.24) P(is-abbreviation] three-letter-word) = 0.8

(2.25) P(three-letter-word) = 0.0003

Exercise 2.4 [⋆]

Are X and Y as defined in the following table independently distributed?

x	0	0	1	1
Y	0	1	0	1
p(X = x, Y = y)	0.32	0.08	0.48	0.12

Exercise 2.5 [⋆]

In example 5, we worked out the expectation of the sum of two dice in terms of the expectation of rolling one die. Show that one gets the same result if one calculates the expectation for two dice directly.

Exercise 2.6 [⋆⋆]

Consider the set of grades you have received for courses taken in the last two years. Convert them to an appropriate numerical scale. What is the appropriate distribution for modeling them?

Exercise 2.7 [⋆⋆]

Find a linguistic phenomenon that the binomial distribution is a good model for. What is your best estimate for the parameter p?

Exercise 2.8 [⋆⋆]

For $i = 8$ and j = 2, confirm that the maximum of equation (2.15) is at 0.8, and that the maximum of equation (2.17) is 0.75. Suppose our prior belief had instead been captured by the equation:

$$P(\mu_m) = 30m^2(1-m)^2$$

What then would the MAP probability be after seeing a particular sequence of 8 heads and 2 tails? (Assume the theory μ_m and a prior belief that the coin is fair.)

2.2 Essential Information Theory

The field of information theory was developed in the 1940s by Claude Shannon, with the initial exposition reported in (Shannon 1948). Shannon was interested in the problem of maximizing the amount of information that you can transmit over an imperfect communication channel such as a noisy phone line (though actually many of his concerns stemmed from codebreaking in World War II). For any source of 'information' and any 'communication channel,' Shannon wanted to be able to determine theoretical maxima for (i) data compression – which turns out to be given by the Entropy H (or more fundamentally, by the Kolmogorov complexity K), and (ii) the transmission rate – which is given by the Channel Capacity C. Until Shannon, people had assumed that necessarily, if you send your message at a higher speed, then more errors must occur during the transmission. But Shannon showed that providing that you transmit the information in your message at a slower rate than the Channel Capacity, then you can make the probability of errors in the transmission of your message as small as you would like.

2.2.1 Entropy

ALPHABET

Let p(x) be the probability mass function of a random variable X, over a discrete set of symbols (or *alphabet*) X:

$$p(x) = P(X = x), \quad x \in x$$

For example, if we toss two coins and count the number of heads, we have a random variable: $p(0) = 1/4, p(1) = 1/2, p(2) = 1/4$.

ENTROPY
SELF-INFORMATION

The *entropy* (or self-information) is the average uncertainty of a single random variable:

(2.26) **Entropy** $$H(p) = H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Entropy measures the amount of information in a random variable. It is normally measured in bits (hence the log to the base 2), but using any other base yields only a linear scaling of results. For the rest of this book, an unadorned log should be read as log to the base 2. Also, for this definition to make sense, we define 0 log 0 = 0.

Example 7: Suppose you are reporting the result of rolling an 8-sided die. Then the entropy is:

$$H(X) = -\sum_{i=1}^{8} p(i) \log p(i) = -\sum_{i=1}^{8} \frac{1}{8} \log \frac{1}{8} = -\log \frac{1}{8} = \log 8 = 3 \text{ bits}$$

This result is what we would expect. Entropy, the amount of information in a random variable, can be thought of as the average length of the message needed to transmit an outcome of that variable. If we wish to send the result of rolling an eight-sided die, the most efficient way is to simply encode the result as a 3 digit binary message:

1	2	3	4	5	6	7	8
001	010	011	100	101	110	111	000

The transmission cost of each result is 3 bits, and there is no cleverer way of encoding the results with a lower average transmission cost. In general, an optimal code sends a message of probability p(i) in $\lceil -\log p(i) \rceil$ bits.

The minus sign at the start of the formula for entropy can be moved inside the logarithm, where it becomes a reciprocal:

(2.27) $$H(X) = \sum_{x \in X} p(x) \log \frac{1}{p(x)}$$

People without any statistics background often think about a formula like this as a sum of the quantity p(x) log(1/ p(x)) for each x. While this is mathematically impeccable, it is the wrong way to think about such equations. Rather you should think of $\sum_{x \in \mathcal{X}} p(x) \ldots$ as an idiom. It says to take a weighted average of the rest of the formula (which will be a function of *x*), where the weighting depends on the probability of each x. Technically, this idiom defines an *expectation,* as we saw earlier. Indeed,

$$(2.28) \quad H(X) = E\left(\log \frac{1}{p(X)}\right)$$

Example 8: Simplified Polynesian Simplified Polynesian[3] appears to be just a random sequence of letters, with the letter frequencies as shown:

p	t	k	a	i	u
1/8	1/4	1/8	1/4	1/8	1/8

Then the per-letter entropy is:

$$\begin{aligned} H(P) &= -\sum_{i \in \{p,t,k,a,i,u\}} P(i) \log P(i) \\ &= -\left[4 \times \frac{1}{8} \log \frac{1}{8} + 2 \times \frac{1}{4} \log \frac{1}{4}\right] \\ &= 2\frac{1}{2} \text{ bits} \end{aligned}$$

This is supported by the fact that we can design a code that on average takes $2\frac{1}{2}$ bits to transmit a letter:

p	t	k	a	i	u
100	00	101	01	110	111

Note that this code has been designed so that fewer bits are used to send more frequent letters, but still so that it can be unambiguously decoded – if a code starts with a 0 then it is of length two, and if it starts with a 1 it is of length 3. There is much work in information theory on the design of such codes, but we will not further discuss them here.

TWENTY QUESTIONS

One can also think of entropy in terms of the Twenty *Questions* game. If you can ask yes/no questions like 'Is it a *t* or an a?' or 'Is it a consonant?' then on average you will need to ask $2\frac{1}{2}$ questions to identify each letter with total certainty (assuming that you ask good questions!). In

3. Polynesian languages, such as Hawai'ian, are well known for their small alphabets.

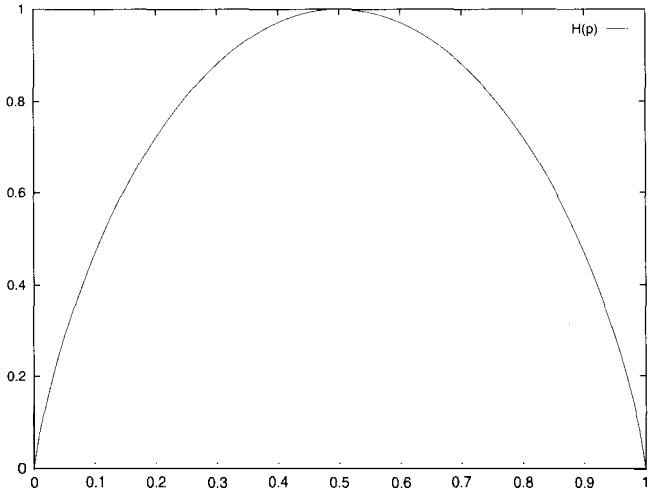

Figure 2.5 The entropy of a weighted coin. The horizontal axis shows the probability of a weighted coin to come up heads. The vertical axis shows the entropy of tossing the corresponding coin once.

other words, entropy can be interpreted as a measure of the size of the 'search space' consisting of the possible values of a random variable and its associated probabilities.

Note that: (i) $H(X) \geq 0$, (ii) $H(X) = 0$ only when the value of X is determinate, hence providing no new information, and that (iii) entropy increases with the message length. The information needed to transmit the results of tossing a possibly weighted coin depends on the probability p that it comes up heads, and on the number of tosses made. The entropy for a single toss is shown in figure 2.5. For multiple tosses, since each is independent, we would just multiply the number in the graph by the number of tosses.

2.2.2 Joint entropy and conditional entropy

The joint entropy of a pair of discrete random variables X, Y $\sim p(x, y)$ is the amount of information needed on average to specify both their values. It is defined as:

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \tag{2.29}$$

The conditional entropy of a discrete random variable Y given another X, for X, Y $\sim p(x, y)$, expresses how much extra information you still need to supply on average to communicate Y given that the other party knows X:

$$
\begin{aligned}
(2.30)\quad H(Y|X) &= \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) \\
&= \sum_{x \in \mathcal{X}} p(x) \left[- \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x) \right] \\
&= - \sum_{x \in \mathcal{X}} \sum_{\text{yty}} p(x, y) \log p(y|x)
\end{aligned}
$$

There is also a Chain rule for entropy:

$$
\begin{aligned}
(2.31)\quad H(X, Y) &= H(X) + H(Y|X) \\
H(X_1, \ldots, X_n) &= H(X_1) + H(X_2|X_1) + \ldots + H(X_n|X_1, \ldots, X_{n-1})
\end{aligned}
$$

The products in the chain rules for probabilities here become sums because of the log:

$$
\begin{aligned}
H(X, Y) &= -E_{p(x,y)} (\log p(x, y)) \\
&= -E_{p(x,y)} (\log(p(x) p(y|x))) \\
&= -E_{p(x,y)} (\log p(x) + \log p(y|x)) \\
&= -E_{p(x)} (\log p(x)) - E_{p(x,y)} (\log p(y|x)) \\
&= H(X) + H(Y|X)
\end{aligned}
$$

Example 9: Simplified Polynesian revisited An important scientific idea is the distinction between a model and reality. Simplified Polynesian isn't a random variable, but we approximated it (or modeled it) as one. But now let's learn a bit more about the language. Further fieldwork has revealed that Simplified Polynesian has syllable structure. Indeed, it turns out that all words consist of sequences of CV (consonant-vowel) syllables. This suggests a better model in terms of two random variables C for the consonant of a syllable, and V for the vowel, whose joint distribution $P(C, V)$ and marginal distributions $P(C, \cdot)$ and $P(\cdot, V)$ are as follows:

(2.32)

	p	t	k	
a	$\frac{1}{16}$	$\frac{3}{8}$	$\frac{1}{16}$	$\frac{1}{2}$
i	$\frac{1}{16}$	$\frac{3}{16}$	0	$\frac{1}{4}$
u	0	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{1}{4}$
	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	

Note that here the marginal probabilities are on a per-syllable basis, and are therefore double the probabilities of the letters on a per-letter basis, which would be:

(2.33)

p	t	k	a	i	u
1/16	3/8	1/16	1/4	1/8	1/8

We can work out the entropy of the joint distribution, in more than one way. Let us use the chain rule:[4]

$$\begin{aligned} H(C) &= 2 \times \frac{1}{8} \times 3 + \frac{3}{4}(2 - \log 3) \\ &= \frac{9}{4} - \frac{3}{4}\log 3 \text{ bits} \approx 1.061 \text{ bits} \end{aligned}$$

$$\begin{aligned} H(V|C) &= \sum_{c=p,t,k} p(C = c) H(V|C = c) \\ &= \frac{1}{8}H\left(\frac{1}{2}, \frac{1}{2}, 0\right) + \frac{3}{4}H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right) + \frac{1}{8}H\left(\frac{1}{2}, 0, \frac{1}{2}\right) \\ &= 2 \times \frac{1}{8} \times 1 + \frac{3}{4}\left[\frac{1}{2} \times 1 + 2 \times \frac{1}{4} \times 2\right] \\ &= \frac{1}{4} + \frac{3}{4} \times \frac{3}{2} \\ &= \frac{11}{8} \text{ bits} = 1.375 \text{ bits} \end{aligned}$$

$$\begin{aligned} H(C, V) &= H(C) + H(V|C) \\ &= \frac{9}{4} - \frac{3}{4}\log 3 + \frac{11}{8} \\ &= \frac{29}{8} - \frac{3}{4}\log 3 \approx 2.44 \text{ bits} \end{aligned}$$

4. Within the calculation, we use an informal, but convenient, notation of expressing a finite-valued distribution as a sequence of probabilities, which we can calculate the entropy of.

Note that those 2.44 bits are now the entropy for a whole syllable (which was 2 x $2\frac{1}{2}$ = 5 for the original Simplified Polynesian example). Our better understanding of the language means that we are now much less uncertain, and hence less surprised by what we see on average than before.

Because the amount of information contained in a message depends on the length of the message, we normally want to talk in terms of the per-letter or per-word entropy. For a message of length n, the per-letter/word entropy, also known as the *entropy rate,* is:[5]

ENTROPY RATE

$$(2.34)\quad H_{\text{rate}} = \frac{1}{n}H(X_{1n}) = -\frac{1}{n}\sum_{x_{1n}} p(x_{1n}) \log p(x_{1n})$$

If we then assume that a language is a stochastic process consisting of a sequence of tokens L = (Xi), for example a transcription of every word you utter in your life, or a corpus comprising everything that is sent down the newswire to your local paper, then we can define the entropy of a human language L as the entropy rate for that stochastic process:

$$(2.35)\quad H_{\text{rate}}(L) = \lim_{n\to\infty} \frac{1}{n}H(X_1, X_2, \ldots, X_n)$$

We take the entropy rate of a language to be the limit of the entropy rate of a sample of the language as the sample gets longer and longer.

2.2.3 Mutual information

By the chain rule for entropy,

$$H(X,Y) = \text{H(X)} + H(Y|X) = \text{H(Y)} + H(X|Y)$$

Therefore,

$$\text{H(X)} - H(X|Y) = \text{H(Y)} - H(Y|X)$$

MUTUAL INFORMATION

This difference is called the *mutual information* between X and Y. It is the reduction in uncertainty of one random variable due to knowing about another, or in other words, the amount of information one random variable contains about another. A diagram illustrating the definition of mutual information and its relationship to entropy is shown in figure 2.6 (adapted from Cover and Thomas (1991: 20)).

5. Commonly throughout this book we use two subscripts on something to indicate a subsequence. So, here, we use X_{ij} to represent the sequence of random variables (Xi,. . . , X_j) and similarly $x_{ij} = (x_i, \ldots, x_j)$. This notation is slightly unusual, but very convenient when sequences are a major part of the domain of discourse. So the reader should remember this convention and be on the lookout for it.

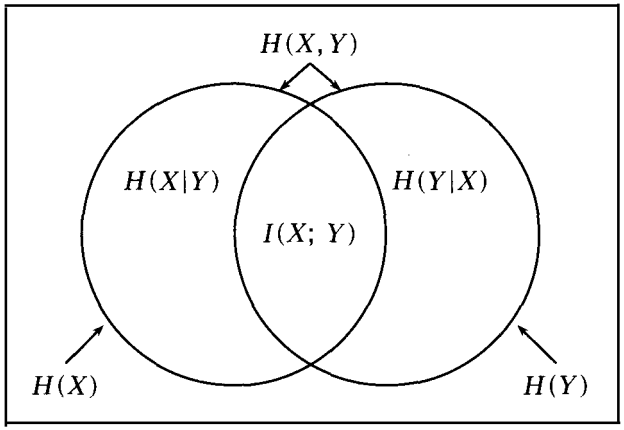

Figure 2.6 The relationship between mutual information I and entropy H.

Mutual information is a symmetric, non-negative measure of the common information in the two variables. People thus often think of mutual information as a measure of dependence between variables. However, it is actually better to think of it as a measure of independence because:

- It is 0 only when two variables are independent, but
- For two dependent variables, mutual information grows not only with the degree of dependence, but also according to the entropy of the variables.

Simple arithmetic gives us the following formulas for mutual information $I(X;Y)$:[6]

$$
\begin{aligned}
(2.36)\quad I(X;Y) &= \mathrm{H(X)} - H(X|Y) \\
&= \mathrm{H(X)} + \mathrm{H(Y)} - H(X,Y) \\
&= \sum_x \mathrm{p}(x) \log \frac{1}{\mathrm{p}(x)} + \sum_y \mathrm{p}(y) \log \frac{1}{\mathrm{p}(y)} + \sum_{x,y} \mathrm{p}(x,y) \log \mathrm{p}(x,y) \\
&= \sum_{x,y} \mathrm{p}(x,y) \log \frac{\mathrm{p}(x,y)}{\mathrm{p}(x)\,\mathrm{p}(y)}.
\end{aligned}
$$

Since $H(X|X) = 0$, note that:

$$\mathrm{H(X)} = \mathrm{H(X)} - H(X|X) = I(X;X)$$

6. Mutual information is conventionally written with a semi-colon separating the two arguments. We are unsure why.

This illustrates both why entropy is also called self-information, and how the mutual information between two totally dependent variables is not constant but depends on their entropy.

We can also derive conditional mutual information and a chain rule:

$$I(X;Y|Z) = I((X;Y)|Z) = H(X|Z) - H(X|Y,Z) \tag{2.37}$$

$$\begin{aligned} I(X_{1n};\mathrm{Y}) &= I(X_1;Y) + \ldots + I(X_n;Y|X_1,\ldots.X_{n-1}) \\ &= \sum_{i=1}^{n} I(X_i;\ Y|X_1,\ldots,X_{i-1}) \end{aligned} \tag{2.38}$$

POINTWISE MUTUAL INFORMATION

In this section we have defined the mutual information between two random variables. Sometimes people talk about the ***pointwise mutual information*** between two particular points in those distributions:

$$I(x,y) = \log \frac{\mathrm{p}(x,\mathrm{Y})}{\mathrm{p}(x)\ \mathrm{P}(\mathrm{Y})}$$

This has sometimes been used as a measure of association between elements, but there are problems with using this measure, as we will discuss in section 5.4.

▼ Mutual information has been used many times in Statistical NLP, such as for clustering words (section 14.1.3). It also turns up in word sense disambiguation (section 7.2.2).

2.2.4 The noisy channel model

Using information theory, Shannon modeled the goal of communicating down a telephone line – or in general across any channel – in the following way: The aim is to optimize in terms of throughput and accuracy the communication of messages in the presence of noise in the channel. It is assumed that the output of the channel depends probabilistically on the input. In general, there is a duality between ***compression,*** which is achieved by removing all redundancy, and transmission accuracy, which is achieved by adding controlled redundancy so that the input can be recovered even in the presence of noise. The goal is to encode the message in such a way that it occupies minimal space while still containing enough redundancy to be able to detect and correct errors. On receipt, the message is then decoded to give what was most likely the original message. This process is shown in figure 2.7.

COMPRESSION

REDUNDANCY

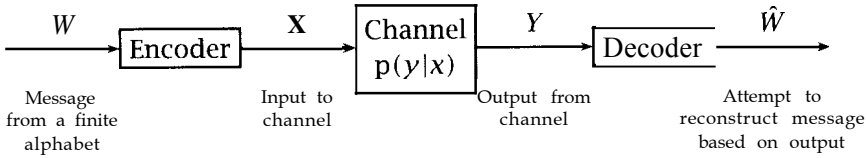

Figure 2.7 The noisy channel model.

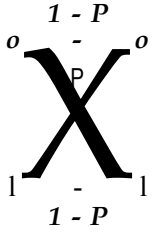

Figure 2.8 A binary symmetric channel. A 1 or a 0 in the input gets flipped on transmission with probability p.

CAPACITY

The central concept that characterizes a channel in information theory is its ***capacity.*** The channel capacity describes the rate at which one can transmit information through the channel with an arbitrarily low probability of being unable to recover the input from the output. For a memoryless channel, Shannon's second theorem states that the channel capacity can be determined in terms of mutual information as follows:

$$C = \max_{p(X)} I(X;Y) \tag{2.39}$$

According to this definition, we reach a channel's capacity if we manage to design an input code X whose distribution maximizes the mutual information between the input and the output over all possible input distributions p(X) .

As an example, consider the binary symmetric channel in figure 2.8. Each input symbol is either a 1 or a 0, and noise in the channel causes each symbol to be flipped in the output with probability ***p.*** We find that:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - H(p) \end{aligned}$$

Therefore,

$$\max_{p(X)} I(X;Y) = 1 - H(p)$$

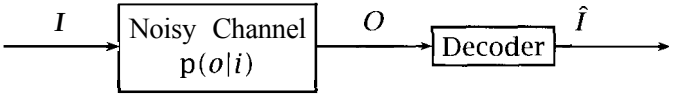

Figure 2.9 The noisy channel model in linguistics.

This last line follows because the mutual information is maximized by maximizing the entropy in the codes, which is done by making the input and hence the output distribution uniform, so their entropy is 1 bit. Since entropy is non-negative, $C \leq 1$. The channel capacity is 1 bit only if the entropy is zero, that is if $p = 0$ and the channel reliably transmits a 0 as 0 and a 1 as 1, or if $p = 1$ and it always flips bits. A completely noisy binary channel which transmits both 0s and 1s with equal probability as 0s and 1s (i.e., $p = \frac{1}{2}$) has capacity $C = 0$, since in this case there is no mutual information between X and Y. Such a channel is useless for communication.

It was one of the early triumphs of information theory that Shannon was able to show two important properties of channels. First, channel capacity is a well-defined notion. In other words, for each channel there is a smallest upper bound of $I(X;Y)$ over possible distributions p(X). Second, in many practical applications it is easy to get close to the optimal channel capacity. We can design a code appropriate for the channel that will transmit information at a rate that is optimal or very close to optimal. The concept of capacity eliminates a good part of the guesswork that was involved in designing communications systems before Shannon. One can precisely evaluate how good a code is for a communication line and design systems with optimal or near-optimal performance.

The noisy channel model is important in Statistical NLP because a simplified version of it was at the heart of the renaissance of quantitative natural language processing in the 1970s. In the first large quantitative project after the early quantitative NLP work in the 1950s and 60s, researchers at IBM's T. J. Watson research center cast both speech recognition and machine translation as a noisy channel problem.

Doing linguistics via the noisy channel model, we do not get to control the encoding phase. We simply want to decode the output to give the most likely input, and so we work with the channel shown in figure 2.9. Many problems in NLP can be construed as an attempt to determine the most likely input given a certain output. We can determine

Application	Input	output	$p(i)$	$p(o\|i)$
Machine Translation	L_1 word sequences	L_2 word sequences	$p(L_1)$ *in a* language model	translation model
Optical Character Recognition (OCR)	actual text	text with mistakes	prob of language text	model of OCR errors
Part Of Speech (POS) tagging	POS tag sequences	English words	prob of POS sequences	$p(w\|t)$
Speech recognition	word sequences	speech signal	prob of word sequences	acoustic model

Table 2.2 Statistical NLP problems as decoding problems.

this as follows, by using Bayes' theorem, and then noting that the output probability is a constant:

$$\hat{I} = \arg\max_i p(i|o) = \arg\max_i \frac{p(i)\,p(o|i)}{p(o)} = \arg\max_i p(i)\,p(o|i) \tag{2.40}$$

LANGUAGE MODEL

Here we have two probability distributions to consider: p(i) is the language ***model,*** the distribution of sequences of 'words' in the input language, and $p(o|i)$ is the channel probability.

CHANNEL PROBABILITY

As an example, suppose we want to translate a text from English to French. The noisy channel model for translation assumes that the true text is in French, but that, unfortunately, when it was transmitted to us, it went through a noisy communication channel and came out as English. So the word cow we see in the text was really ***vuche,*** garbled by the noisy channel to cow. All we need to do in order to translate is to recover the original French – or to ***decode*** the English to get the French.[7]

DECODE

The validity of the noisy channel model for translation is still giving rise to many a heated debate among NLP researchers, but there is no doubt that it is an elegant mathematical framework that has inspired a significant amount of important research. We will discuss the model in more detail in chapter 13. Other problems in Statistical NLP can also be seen as instantiations of the decoding problem. A selection is shown in table 2.2.

7. The French reader may be sympathetic with the view that English is really a form of garbled French that makes the language of *clarté* unnecessarily ambiguous!

2.2.5 Relative entropy or Kullback-Leibler divergence

RELATIVE ENTROPY

For two probability mass functions, p(x), q(x) their relative *entropy* is given by:

$$D(\mathrm{p} \,\|\, \mathrm{q}) = \sum_{x \in \mathcal{X}} \mathrm{p}(x) \log \frac{\mathrm{p}(x)}{\mathrm{q}(x)} \tag{2.41}$$

where again we define $0 \log \frac{0}{\mathrm{q}} = 0$ and otherwise $\mathrm{p} \log \frac{\mathrm{p}}{0} = \infty$. The relative entropy, also known as the *Kullback-Leibler* divergence, is a measure of how different two probability distributions (over the same event space) are. Expressed as an expectation, we have:

KULLBACK-LEIBLER DIVERGENCE

$$D(\mathrm{p} \,\|\, \mathrm{q}) = E_{\mathrm{p}} \left(\log \frac{\mathrm{p}(X)}{\mathrm{q}(X)} \right) \tag{2.42}$$

Thus, the KL divergence between p and q is the average number of bits that are wasted by encoding events from a distribution p with a code based on a not-quite-right distribution q.

This quantity is always non-negative, and $D(\mathrm{p} \| \mathrm{q}) = 0$ iff p = q. For these reasons, some authors use the name 'KL distance,' but note that relative entropy is not a metric (in the sense in which the term is used in mathematics): it is not symmetric in p and q (see exercise 2.12), and it does not satisfy the *triangle inequality*.[8] Hence we will use the name 'KL divergence,' but nevertheless, informally, people often think about the relative entropy as the 'distance' between two probability distributions: it gives us a measure of how close two pmfs are.

TRIANGLE INEQUALITY

Mutual information is actually just a measure of how far a joint distribution is from independence:

$$I(X; Y) = D(\mathrm{p}(x, y) \| \mathrm{p}(x)\, \mathrm{p}(y)) \tag{2.43}$$

We can also derive conditional relative entropy and a chain rule for relative entropy (Cover and Thomas 1991: 23):

$$D(\mathrm{p}(y|x) \,\|\, \mathrm{q}(y|x)) = \sum_{x} \mathrm{p}(x) \sum_{y} \mathrm{p}(y|x) \log \frac{\mathrm{p}(y|x)}{\mathrm{q}(y|x)} \tag{2.44}$$

$$D(\mathrm{p}(x, y) \| \mathrm{q}(x, y)) = D(\mathrm{p}(x) \| \mathrm{q}(x)) + D(\mathrm{p}(y|x) \| \mathrm{q}(y|x)) \tag{2.45}$$

8. The triangle inequality is that for any three points x, y, z:

$$d(x, y) \leq d(x, z) + d(z, y)$$

▼ KL divergence is used for measuring selectional preferences in section 8.4.

2.2.6 The relation to language: Cross entropy

So far we have examined the notion of entropy, and seen roughly how it is a guide to determining efficient codes for sending messages, but how does this relate to understanding language? The secret to this is to return to the idea that entropy is a measure of our uncertainty. The more we know about something, the lower the entropy will be because we are less surprised by the outcome of a trial.

We can illustrate this with the examples used above. Consider again Simplified Polynesian from examples 8 and 9. This language has 6 letters. The simplest code is to use 3 bits for each letter of the language. This is equivalent to assuming that a good model of the language (where our 'model' is simply a probability distribution) is a uniform model. However, we noticed that not all the letters occurred equally often, and, noting these frequencies, produced a zeroth order model of the language. This had a lower entropy of 2.5 bits per letter (and we showed how this observation could be used to produce a more efficient code for transmitting the language). Thereafter, we noticed the syllable structure of the language, and developed an even better model that incorporated that syllable structure into it. The resulting model had an even lower entropy of 1.22 bits per letter. The essential point here is that if a model captures more of the structure of a language, then the entropy of the model should be lower. In other words, we can use entropy as a measure of the quality of our models.

Alternately, we can think of entropy as a matter of how surprised we will be. Suppose that we are trying to predict the next word in a Simplified Polynesian text. That is, we are examining $P(w|h)$, where w is the next word and h is the history of words seen so far. A measure of our surprise on seeing the next word can be derived in terms of the conditional probability assigned to w by our model m of the distribution of Simplified Polynesian words. Surprise can be measured by what we might term the *pointwise entropy* $H(w|h) = -log_2 \mathrm{m}(w|h)$. If the predictor is certain that word w follows a given history h and it is correct, then the information supplied to the predictor on seeing w is $-\log_2 1 = 0$. In other words, the predictor does not experience any surprise at all. On the other hand, if the model thinks that w cannot follow h, then $\mathrm{m}(w|h) = 0$ and

SURPRISE

POINTWISE ENTROPY

so the information supplied to the predictor is infinite ($-\log 0 = \infty$). In this case our model is infinitely surprised, which is normally a very bad thing. Usually our models will predict a probability between these two extremes for each event and so the model will gain some information, or alternatively, be somewhat surprised, when it sees the next word, and the goal is to keep that level of surprise as low as possible. Summing over the surprise of the predictor at each word gives an expression for our total surprise:

$$\begin{aligned} H_{\text{total}} &= -\sum_{j=1}^{n} \log_2 \mathrm{m}(w_j | w_1, w_2, \ldots, w_{j-1}) \\ &= -\log_2 \mathrm{m}(w_1, w_2, \ldots, w_n) \end{aligned}$$

The second line above follows from the chain rule. Normally, we would want to normalize this measure by the length of the text so our notion of surprise is not dependent on the size of the text. This normalized measure gives the average surprise of the predictor per word.

So far this discussion has been rather informal, but we can formalize it through the notion of relative entropy. Suppose that we have some empirical phenomenon, in Statistical NLP usually utterances in a certain language. Assuming some mapping to numbers, we can represent it via a random variable X. Then we assume that there is some probability distribution over the utterances – for instance, you hear ***Thank you*** much more often than ***On you. So we*** take $X \sim p(x)$.

Now, unfortunately we do not know what $p(\cdot)$ is for empirical phenomena. But by looking at instances, for example by looking at a corpus of utterances, we can estimate roughly what p seems to be like. In other words, we can produce a model m of the real distribution, based on our best estimates. In making this model, what we want to do is to minimize $D(\mathrm{p} \,\|\, \mathrm{m})$ – to have as accurate a probabilistic model as possible. Unfortunately, we normally cannot calculate this relative entropy – again, because we do not know what p is. However, there is a related quantity, the cross entropy, which we fortunately can get a handle on.

CROSS ENTROPY

The ***cross entropy*** between a random variable X with true probability distribution p(x) and another pmf q (normally a model of p) is given by:

$$\begin{aligned} H(X, \mathrm{q}) &= H(X) + D(\mathrm{p} \,\|\, \mathrm{q}) \\ &= -\sum_{x} \mathrm{p}(x) \log \mathrm{q}(x) \end{aligned} \tag{2.46}$$

$$= E_p\left(\log \frac{1}{q(x)}\right) \tag{2.47}$$

(Proof of this is left to the reader as exercise 2.13.)

Just as we defined the entropy of a language in section 2.2.2, we can define the cross entropy of a language $L = (Xi) \sim p(x)$ according to a model m by:

$$H(L, m) = -\lim_{n\to\infty} \frac{1}{n} \sum_{x_{1n}} p(x_{1n}) \log m(x_{1n}) \tag{2.48}$$

We do not seem to be making much progress, because it still seems that we cannot calculate this quantity without knowing p. But if we make certain assumptions that the language is 'nice,' then the cross entropy for the language can be calculated as:

$$H(L, m) = -\lim_{n\to\infty} \frac{1}{n} \log m(x_{1n}) \tag{2.49}$$

Using this second form, we can calculate the cross entropy based only on knowing our probability model and having a large body of utterances available. That is, we do not actually attempt to calculate the limit, but approximate it by calculating for a sufficiently large n:

$$H(L, m) \approx -\frac{1}{n} \log m(x_{1n}) \tag{2.50}$$

This measure is just the figure for our average surprise. Our goal will be to try to minimize this number. Because H(X) is fixed (if unknown), this is equivalent to minimizing the relative entropy, which is a measure of how much our probability distribution departs from actual language use. The only additional requirement is that the text that we use to test the model must be an independent test set, and not part of the training corpus that we used to estimate the parameters of the model. Cross entropy is inversely related to the average probability a model assigns to words in test data. Lower model cross entropy normally leads to better performance in applications, but it need not do so if it is just a matter of improving the magnitude of probability estimates, but not their relative ordering. (See section 6.2.3 for more practical details on calculating the cross entropy of models.)

But what justifies going from equation (2.48) to equation (2.49)? The formula for language cross entropy has an expectation embedded within it:

$$H(L, m) = \lim_{n\to\infty} \frac{1}{n} E\left(\log \frac{1}{m(X_{1n})}\right) \tag{2.51}$$

Recall that the expectation is a weighted average over all possible sequences. But in the above formula we are using a limit and looking at longer and longer sequences of language use. Intuitively, the idea is then that if we have seen a huge amount of the language, what we have seen is 'typical.' We no longer need to average over all samples of the language; the value for the entropy rate given by this particular sample will be roughly right.

The formal version of this is to say that if we assume that $L = (\mathrm{Xi})$ is a stationary ergodic process, then we can prove the above result. This is a consequence of the Shannon-McMillan-Breiman theorem, also known as the Asymptotic Equipartition Property:

Theorem: If H_{rate} is the entropy rate of a finite-valued stationary ergodic process (X_n), then:

$$-\frac{1}{n}\log \mathrm{p}(X_1, \ldots, X_n) \rightarrow H, \quad \text{with probability 1}$$

We will not prove this theorem; see Cover and Thomas (1991: ch. 3, 15).

ERGODIC An *ergodic* process is one that, roughly, cannot get into different substates that it will not escape from. An example of a non-ergodic process is one that in the beginning chooses one of two states: one in which it generates 0 forever, one in which it generates 1 forever. If a process is not ergodic, then even looking at one very long sequence will not necessarily tell us what its typical behavior is (for example, what is likely to happen when it gets restarted).

STATIONARY A stationary process is one that does not change over time. This is clearly wrong for language: new expressions regularly enter the language while others die out. And so, it is not exactly correct to use this result to allow the calculation of a value for cross entropy for language applications. Nevertheless, for a snapshot of text from a certain period (such as one year's newswire), we can assume that the language is near enough to unchanging, and so this is an acceptable approximation to truth. At any rate, this is the method regularly used.

2.2.7 The entropy of English

As noted above, English in general is not a stationary ergodic process. But we can nevertheless model it with various stochastic approximations. In

n-GRAM MODELS particular, we can model English with what are known as ***n-gram models***

MARKOVCHAINS

MARKOVASSUMPTION

or *Markov chains.* These models, which we discuss in detail in chapters 6 and 9, are ones where we assume a limited memory. We assume that the probability of the next word depends only on the previous k words in the input. This gives a k^{th} order *Markov* approximation:

$$P(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = P(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_{n-k} = x_{n-k})$$

If we are working on a character basis, for example, we are trying to guess what the next character in a text will be given the preceding k characters. Because of the redundancy of English, this is normally fairly easy. For instance, a generation of students have proved this by being able to make do with photocopies of articles that are missing the last character or two of every line.

By adding up counts of letters, letter digraphs (that is, sequences of two letters), and so on in English, one can produce upper bounds for the entropy of English." We assume some such simplified model of English and compute its cross entropy against a text and this gives us an upper bound for the true entropy of English – since $D(\mathrm{p}||\mathrm{m}) \geq 0$, $H(X, \mathrm{m}) \geq \mathrm{H(X)}$. Shannon did this, assuming that English consisted of just 27 symbols (the 26 letters of the alphabet and SPACE – he ignored case distinctions and punctuation). The estimates he derived were:

(2.52)

Model	Cross entropy (bits)	
zeroth order	4.76	(uniform model, so log 2 7)
first order	4.03	
second order	2.8	
Shannon's experiment	1.3 (1.34)	(Cover and Thomas 1991: 140)

The first three lines show that as the order of the model increases, that is, as information about the frequencies of letters (first order) and digraphs (second order) is used, our model of English improves and the calculated cross entropy drops. Shannon wanted a tighter upper bound on the entropy of English, and derived one by human experiments – finding out how good at guessing the next letter in a text a human being was. This gave a much lower entropy bound for English. (A later experiment with

9. More strictly, one produces an estimate for the text on which the counts are based, and these counts are good for 'English only to the extent that the text used is representative of English as a whole. Working at the character level, this is not too severe a problem, but it becomes quite important when working at the word level, as discussed in chapter 4.

more subjects on the same text that Shannon used produced the figure in parentheses, 1.34.)

Of course, the real entropy of English must be lower still: there are doubtless patterns in people's speech that humans do not pick up on (although maybe not that many!). But at present, the statistical language models that we can construct are much worse than human beings, and so the current goal is to produce models that are as good as English speakers at knowing which English utterances sound normal or common and which sound abnormal or marked.

▼ We return to n-gram models in chapter 6.

2.2.8 Perplexity

PERPLEXITY In the speech recognition community, people tend to refer to ***perplexity*** rather than cross entropy. The relationship between the two is simple:

$$\text{perplexity}(x_{1n}, \mathrm{m}) = 2^{H(x_{1n},\mathrm{m})} \tag{2.53}$$

$$= \mathrm{m}(x_{1n})^{\frac{1}{n}} \tag{2.54}$$

We suspect that speech recognition people prefer to report the larger non-logarithmic numbers given by perplexity mainly because it is much easier to impress funding bodies by saying that "we've managed to reduce perplexity from 950 to only 540" than by saying that "we've reduced cross entropy from 9.9 to 9.1 bits." However, perplexity does also have an intuitive reading: a perplexity of k means that you are as surprised on average as you would have been if you had had to guess between k equiprobable choices at each step.

2.2.9 Exercises

Exercise 2.9 [⋆]

Take a (short) piece of text and compute the relative frequencies of the letters in the text. Assume these are the true probabilities. What is the entropy of this distribution?

Exercise 2.10 [⋆]

Take another piece of text and compute a second probability distribution over letters by the same method. What is the KL divergence between the two distributions? (You will need to 'smooth' the second distribution and replace any zero with a small quantity E.)

Exercise 2.11 [⋆]

Cast the problem of word sense disambiguation as a noisy channel model, in analogy to the examples in table 2.2. Word sense disambiguation is the problem of determining which sense of an ambiguous word is used (e.g., 'industrial plant' vs. 'living plant' for *plant)* and will be covered in chapter 7.

Exercise 2.12 [⋆]

Show that the KL divergence is not symmetric by finding an example of two distributions p and q for which $D(p\|q) \neq D(q\|p)$.

Exercise 2.13 [⋆]

Prove the equality shown in the first two lines of (2.46).

Exercise 2.14 [⋆]

We arrived at the simplified way of computing cross entropy in equation (2.49) under the premise that the process we are dealing with is ergodic and stationary. List some characteristics of natural languages that show that these two properties are only approximately true of English.

Exercise 2.15 [⋆⋆]

Reproduce Shannon's experiment. Write a program that shows you a text one letter at a time. Run it on a text you have not seen. Can you confirm Shannon's estimate of the entropy of English?

Exercise 2.16 [⋆⋆]

Repeat the last exercise for one text that is 'easy' (e.g., a newsgroup posting) and one text that is 'hard' (e.g., a scientific article from a field you don't know well). Do you get different estimates? If the estimates are different, what difficulties does the experiment raise for interpreting the different estimates of the entropy of English?

2.3 Further Reading

Aho et al. (1986: ch. 4) cover parsing in computer science, and Allen (1995: ch. 3) covers parsing in computational linguistics. Most of the mathematics we use is covered in Part I of (Cormen et al. 1990), but not vector spaces and matrices, for which one should consult an introduction to linear algebra such as (Strang 1988).

Many books give good introductions to basic probability theory. A few good ones, listed in approximate order of increasing difficulty are (Moore and McCabe 1989; Freedman et al. 1998; Siegel and Castellan 1988; DeGroot 1975). Krenn and Samuelsson (1997) is particularly recommended as a much more thorough introduction to statistics aimed at a Statistical

NLP audience. Unfortunately most introduction to statistics textbooks follow a very fixed syllabus which is dominated by hypothesis testing as applied in experimental sciences such as biology and psychology. Often these concerns are rather distant from the issues of most relevance to Statistical NLP, and it can be helpful to also look at books covering quantitative methods for machine learning, such as (Mitchell 1997).

The coverage of information theory here barely scratches the surface of that field. Cover and Thomas (1991) provide a thorough introduction.

Brown et al. (1992b) present an estimate of 1.75 bits per character for the entropy of English based on predicting the next word, trained on an enormous corpus of English text.