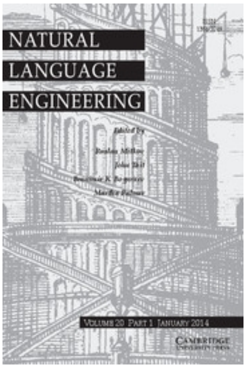

A hierarchical Dirichlet language model

David J. C. MacKay and Linda C. Bauman Peto

Natural Language Engineering / Volume 1 / Issue 03 / September 1995, pp 289 - 308
DOI: 10.1017/S1351324900000218, Published online: 12 September 2008

Link to this article: http://journals.cambridge.org/abstract_S1351324900000218

How to cite this article:
David J. C. MacKay and Linda C. Bauman Peto (1995). A hierarchical Dirichlet language model. Natural Language Engineering, 1, pp 289-308 doi:10.1017/S1351324900000218

A hierarchical Dirichlet language model

DAVID J. C. MACKAY
Cavendish Laboratory
Cambridge CB3 OHE, UK
email: mackay@mrao.cam.ac.uk

LINDA C. BAUMAN PETO†
Department of Computer Science
University of Toronto, Canada
email: peto@cs.toronto.edu

(*Received 16 February 1995*)

Abstract

We discuss a hierarchical probabilistic model whose predictions are similar to those of the popular language modelling procedure known as 'smoothing'. A number of interesting differences from smoothing emerge. The insights gained from a probabilistic view of this problem point towards new directions for language modelling. The ideas of this paper are also applicable to other problems such as the modelling of triphomes in speech, and DNA and protein sequences in molecular biology. The new algorithm is compared with smoothing on a two million word corpus. The methods prove to be about equally accurate, with the hierarchical model using fewer computational resources.

1 Introduction

Speech recognition and automatic translation both depend upon a language model that assigns probabilities to word sequences. The automatic translation system implemented at IBM used a crude 'trigram' model of language with impressive results (Brown, DellaPietra, DellaPietra and Mercer 1993). Similar language models are also used in speech recognition systems (Bahl, Jelinek and Mercer 1983; Jelinek and Mercer 1980). Trigram models are often implemented using a particular kludge involving 'smoothing' their predictions with the predictions of better-determined bigram and monogram models, the smoothing coefficients being determined by 'deleted interpolation' (Jelinek and Mercer 1980; Bahl, Brown, de Souza, Mercer and Nahamoo 1991). Another generally used language model employs a similar procedure known as 'backing off' (Katz 1987).

Text compression is a similar prediction task in which character sequences are to be predicted (adaptively, or otherwise). In text compression, the smoothing technique

† Present address: c/o P.O. Box 3398, Cambridge, Ontario, N3H 4T3 Canada.

is known as 'blending' and is used to combine the predictions obtained using contexts of different orders (Bell, Cleary and Witten 1990).

This paper's aim is to reverse-engineer the underlying model which gives a probabilistic meaning to smoothing, allowing it to be better understood, objectively tuned and sensibly modified. The objective is not to create a rival language model but rather to demonstrate the Bayesian approach to language modelling and show that it is feasible. For simplicity, this paper will pretend that the language model is simply a bigram model, since the key issues can be addressed by studying the smoothing of bigram statistics. This paper assumes throughout that a bigram model, i.e. a Markov process, is an appropriate language model, and discusses optimal inference subject to that assumption.

1.1 The bigram language model with smoothing

To develop a predictive model for language, a string of T words $D = w_1, w_2, \dots w_T$, is observed and the marginal and conditional frequencies are observed. We define the marginal count F_i to be the number of times that word i occurs, and the conditional count $F_{i|j}$ to be the number of times that word j is immediately followed by word i. (We are ignoring the option of grouping words by a common root and other complications not central to the concept of smoothing.) Given these counts, 'estimators' of the marginal probability of word i and of the conditional probability of word i following word j are $f_i = (F_i + \alpha/W)/(T + \alpha)$ and $f_{i|j} = (F_{i|j} + \beta/W)/(F_j + \beta)$, where the 'initial counts' α/W and β/W are commonly set to 0, 1/2 or 1 (Bell *et al.* 1990). The subscripts i and j run from 1 to W, the total number of distinct words in the language. If the initial counts are set to 0 we obtain the maximum likelihood estimators $f_i = F_i/T$ and $f_{i|j} = F_{i|j}/F_j$, which assign zero frequency to all words and word pairs that did not occur in the data. The practical aim of language modelling is to predict what word w_t will be given w_{t-1} and given all the available information and data. This prediction is described by a 'predictive' probability over w_t, $\hat{P}(w_t|w_{t-1})$. One might be inclined to use the observed conditional frequency $f_{w_t|w_{t-1}}$ as an estimator for this predictive probability, if the statistics were adequate. But typically (and especially in the case of trigram modelling), this conditional frequency estimator has large variance, because there are so many possible couplets ij that only a small fraction of them have been observed in the data. So the following kludge is adopted:

$$\hat{P}(w_t|w_{t-1}) = \lambda f_{w_t} + (1 - \lambda) f_{w_t|w_{t-1}}. \tag{1}$$

Thus the noisy bigram statistics are 'smoothed' by the better determined monogram model's predictions. A cross-validation procedure called 'deleted interpolation' is used to set λ (Jelinek and Mercer 1980; Bahl *et al.* 1991). This involves dividing the data into a number of blocks, computing predictions for each block using the other blocks as training data, and adjusting λ to optimize predictive performance. It has been found that better predictions can be obtained if contexts w_1 with similar values of f_{w_1} are grouped together, with a separate λ for each group determined by deleted interpolation.

In text compression, 'blending' combines together the predictions of different models in a manner similar to equation (1). The parameters equivalent to λ are not adapted, but are fixed by the *a priori* choice of an 'escape mechanism'. According to (Bell *et al.* 1990), "there can be no theoretical justification for choosing any particular escape mechanism". We would agree that it is not possible to make language models without making *a priori* assumptions; but we argue that it is possible within a *hierarchical* model effectively to determine the smoothing parameters *a posteriori* from the data.

1.2 Any rational predictive procedure can be made Bayesian

The smoothing procedure sounds sensible, but slightly *ad hoc*. Since rational inference can always be mapped onto probabilities (Cox 1946), the aim of this paper is to discover what implicit probabilistic model the above procedure can be related to. The smoothing formula and deleted interpolation were originally conceived as a way of combining together the predictions of different models. But in this paper we will define a single hierarchical model with a non-trivial Dirichlet prior which gives predictive distributions similar to (1), including adaptive expressions for the weighting coefficients equivalent to λ. However, various interesting differences will emerge, highlighting problems with equation (1).

2 An explicit model using Dirichlet priors

The heart of a bigram model is a conditional distribution $P(w_t = i|w_{t-1} = j)$, described by $W(W-1)$ independent parameters, where W is the number of words in the language [W possible conditioning terms on the right-hand side, for each of which a probability distribution with $(W-1)$ independent parameters is specified]. These parameters will be denoted by Q, with $P(w_t = i|w_{t-1} = j) \equiv q_{i|j}$. Q is a $W \times W$ transition probability matrix. A single row of Q, the probability vector for transitions from state j, is denoted by $\mathbf{q}_{|j}$. (Alternative ways of parameterizing the model might be defined using, for example, the marginal word probabilities $P(w_t)$ and the joint probabilities $P(w_t, w_{t-1})$. However, the conditional probability parameterization Q is chosen because it is the natural representation of a Markov process; the marginal distribution $P(w_t)$ is not independent, but is a deterministic function of the conditional probability matrix Q: namely, $P(w_t)$ is the principal eigenvector of Q.) The parameters Q are never perfectly known, and our uncertainty about their values can be represented by a probability distribution over possible Qs.

2.1 The inferences we will make

A model $\mathcal{H}$ is a specification of the model parameters, the way that the probability of the data depends on those parameters, and a prior probability distribution on those parameters. Given a model $\mathcal{H}$, there are two inferences we will be interested in making. Both these inferences can be made mechanically using the rules of probability theory:

A: Infer the parameters given the data
We do this by Bayes' theorem, which gives the probability of the parameters Q given the data D in terms of the likelihood function $P(D|Q,\mathcal{H})$ and the prior distribution $P(Q|\mathcal{H})$:

$$P(Q|D,\mathcal{H}) = \frac{P(D|Q,\mathcal{H})P(Q|\mathcal{H})}{P(D|\mathcal{H})}. \tag{2}$$

The normalizing constant is given by integrating the numerator over Q:

$$P(D|\mathcal{H}) = \int P(D|Q,\mathcal{H})P(Q|\mathcal{H})\, d^kQ, \tag{3}$$

where k is the dimensionality of Q.

B: Predict the next word in a given context
To obtain the probability of w_t given w_{t-1} and the data D, we use the sum rule of probability $P(A|C) = \int P(A|B,C)P(B|C)\, dB$ to marginalize over the unknown parameters Q:

$$\begin{aligned} P(w_t|w_{t-1},D,\mathcal{H}) &= \int P(w_t|w_{t-1},Q,D,\mathcal{H})P(Q|D,\mathcal{H})\, d^kQ && (4)\\ &= \int q_{w_t|w_{t-1}} P(Q|D,\mathcal{H})\, d^kQ. && (5) \end{aligned}$$

The distribution inside the integral, $P(Q|D,\mathcal{H})$, depends upon the likelihood function and the prior, as shown in equation (2).

2.2 The likelihood function

The likelihood function $P(D|Q,\mathcal{H})$ can be written down immediately, independent of the assumptions $\mathcal{H}$ which define the rest of the model. We make the simplifying assumption that the first word of the data set is given *a priori*, and is not to be predicted by the model. The probability of the string of words is then the probability of the second word given the first, times the probability of the third word given the second, and so forth:

$$P(D|Q,\mathcal{H}) = \prod_t q_{w_t|w_{t-1}}. \tag{6}$$

We can rewrite this product by counting how often each variable $q_{i|j}$ appears in the product. This is given by the conditional count $F_{i|j}$. Thus

$$P(D|Q,\mathcal{H}) = \prod_j \prod_i q_{i|j}^{F_{i|j}}. \tag{7}$$

So given the assumed bigram model, the conditional counts $F_{i|j}$ contain all the relevant information that the data convey about Q.

2.3 What prior?

Thus having defined the parameterization of the model, Q, the only question that remains before the two inferences above are fully defined is 'what is the prior over

Q?' In particular, this paper examines the question, *what prior* $P(Q|\mathcal{H})$ *would give us predictive distributions of the 'smoothed' form (1)?*

2.4 A convenient family of priors: Dirichlet distributions

The Dirichlet distribution (Antoniak 1974) for a probability vector $\mathbf{p}$ with I components is parameterized by a measure $\mathbf{u}$ (a vector with all coefficients $u_i > 0$) which we will write here as $\mathbf{u} = \alpha\mathbf{m}$, where $\mathbf{m}$ is a normalized measure over the I components ($\sum m_i = 1$), and α is a positive scalar:

$$P(\mathbf{p}|\alpha\mathbf{m}) = \frac{1}{Z(\alpha\mathbf{m})} \prod_{i=1}^{I} p^{\alpha m_i - 1} \delta\left(\textstyle\sum_i p_i - 1\right) \equiv \text{Dirichlet}^{(I)}(\mathbf{p}|\alpha\mathbf{m}). \tag{8}$$

The function $\delta(x)$ is the Dirac delta function which simply restricts the distribution to the simplex such that $\mathbf{p}$ is normalized, i.e. $\sum_i p_i = 1$. The normalizing constant of the Dirichlet distribution is:

$$Z(\alpha\mathbf{m}) = \prod_i \Gamma(\alpha m_i) / \Gamma(\alpha). \tag{9}$$

The vector $\mathbf{m}$ is the mean of the probability distribution:

$$\int \text{Dirichlet}^{(I)}(\mathbf{p}|\alpha\mathbf{m})\, \mathbf{p}\, d^I\mathbf{p} = \mathbf{m}. \tag{10}$$

The role of α can be characterized in two ways. First, the parameter α measures the sharpness of the distribution; it measures how different we expect typical samples $\mathbf{p}$ from the distribution to be from the mean $\mathbf{m}$. A large value of α produces a distribution over $\mathbf{p}$ which is sharply peaked around $\mathbf{m}$. The effect of α can be visualized by drawing a typical sample from the distribution Dirichlet$^{(I)}(\mathbf{p}|\alpha\mathbf{m})$, with $\mathbf{m}$ set to the uniform vector $m_i = 1/I$, and making a Zipf plot, that is, a ranked plot of the values of the components p_i. It is traditional to plot both p_i (vertical axis) and the rank (horizontal axis) on logarithmic scales so that power law relationships appear as straight lines. Figure 1 shows these plots for a single sample from ensembles with $I = 100$ and $I = 1000$ and with α from 0.1 to 1000. For large α, the plot is shallow with many components having similar values. For small α, typically one component p_i receives an overwhelming share of the probability, and of the probability that remains to be shared among the other components, another component $p_{i'}$ receives a similarly large share. In the limit as α goes to zero, the plot tends to an increasingly steep power law.

Second, we can characterize the role of α in terms of the predictive distribution that results when we observe samples from $\mathbf{p}$ and obtain counts $\mathbf{F} = (F_1, F_2, \ldots F_I)$ of the possible outcomes. The posterior probability of $\mathbf{p}$ is, conveniently, another Dirichlet distribution:

$$P(\mathbf{p}|\mathbf{F}, \alpha\mathbf{m}) = \frac{P(\mathbf{F}|\mathbf{p})P(\mathbf{p}|\alpha\mathbf{m})}{P(\mathbf{F}|\alpha\mathbf{m})} \tag{11}$$

$$= \frac{\prod_i p_i^{F_i} \prod_i p_i^{\alpha m_i - 1} \delta(\sum_i p_i - 1)/Z(\alpha\mathbf{m})}{P(\mathbf{F}|\alpha\mathbf{m})} \tag{12}$$

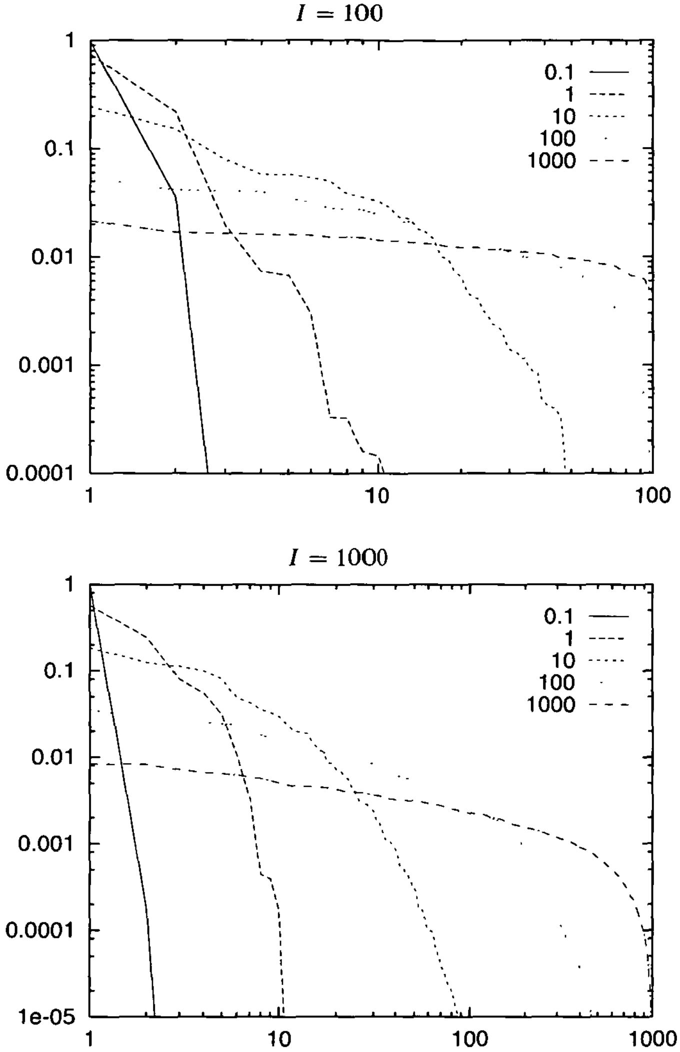

Fig. 1. Zipf plots for random samples from Dirichlet distributions with various values of $\alpha = 0.1 \ldots 1000$. For each given I and α, I samples from a standard gamma distribution were generated with shape parameter α/I and normalized to give a sample $\mathbf{p}$ from the Dirichlet distribution. The Zipf plot shows the probabilities p_i, ranked by magnitude, versus their rank.

$$= \frac{\prod_i p_i^{F_i + \alpha m_i - 1} \delta(\sum_i p_i - 1)}{P(\mathbf{F}|\alpha\mathbf{m}) Z(\alpha\mathbf{m})} \tag{13}$$

$$= \text{Dirichlet}^{(I)}(\mathbf{p}|\mathbf{F} + \alpha\mathbf{m}). \tag{14}$$

The predictive distribution given the data $\mathbf{F}$ is then:

$$P(i|\mathbf{F}, \alpha\mathbf{m}) = \int \text{Dirichlet}^{(I)}(\mathbf{p}|\mathbf{F} + \alpha\mathbf{m})\, \mathbf{p}\ d^I\mathbf{p} = \frac{F_i + \alpha m_i}{\sum_{i'} F_{i'} + \alpha m_{i'}}. \tag{15}$$

Notice that the term αm_i appears as an effective initial count in bin i. The value of α defines the number of samples from $\mathbf{p}$ that are required in order that the data dominate over the prior in subsequent predictions. If $\alpha \gg \sum_i F_i$ then $P(i|\mathbf{F}, \alpha\mathbf{m}) \simeq m_i$; if $\alpha \ll \sum_i F_i$ then $P(i|\mathbf{F}, \alpha\mathbf{m}) \simeq F_i/(\sum_{i'} F_{i'})$.

Finally, we note from equations (9) and (14) that the 'evidence' for $\alpha\mathbf{m}$, $P(\mathbf{F}|\alpha\mathbf{m})$, is:

$$(16) \qquad P(\mathbf{F}|\alpha\mathbf{m}) = \frac{Z(\mathbf{F} + \alpha\mathbf{m})}{Z(\alpha\mathbf{m})} = \frac{\prod_i \Gamma(F_i + \alpha m_i)}{\Gamma(\sum_i F_i + \alpha)} \frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha m_i)}.$$

The important role of the evidence (also known as the marginalized likelihood) will become clear shortly. Additional useful formulae and approximations are found in Appendix A.

2.5 *Definition of the hierarchical model $\mathcal{H}_{\mathrm{D}}$*

We now define the prior of a hierarchical model that we denote $\mathcal{H}_D$ (D for Dirichlet). It is called a hierarchical model because as well as containing unknown parameters Q which place a probability distribution on data, it contains unknown 'hyperparameters' which define a probability distribution over the parameters Q.

To obtain predictions similar to those of the smoothing equation (1), we must assign a *coupled* prior to the parameters Q, that is, a prior under which learning the probability vector for one context $\mathbf{q}_j$ gives us information about what the probability vectors $\mathbf{q}_{j'}$ in other contexts might be. We introduce an *unknown* measure on the words, $\mathbf{u} = \alpha\mathbf{m}$, and define a separable prior, given $\alpha\mathbf{m}$, on the vectors $\mathbf{q}_{|j}$ that make up Q:

$$(17) \qquad P(Q|\alpha\mathbf{m}, \mathcal{H}_D) = \prod_j \text{Dirichlet}^{(I)}(\mathbf{q}_{|j}|\alpha\mathbf{m}).$$

We produce a dependence between the vectors $\mathbf{q}_{|j}$ by putting an uninformative prior $P(\alpha\mathbf{m})$ on the measure $\alpha\mathbf{m}$ (to be precise, a flat prior on $\mathbf{m}$ and a broad gamma prior over α). The prior on Q defined by this hierarchical model is then:

$$(18) \qquad P(Q|\mathcal{H}_D) = \int \prod_j \left[\text{Dirichlet}^{(I)}(\mathbf{q}_{|j}|\alpha\mathbf{m})\right] P(\alpha\mathbf{m}) d^I \alpha\mathbf{m}.$$

When we use this hierarchical model we can effectively find out from the data what the measure should be, as we will show in section 3. If we have additional prior knowledge about the language such that we expect specific structure in the measure, then we could define a more informative prior $P(\alpha\mathbf{m})$ which should further improve the model's predictive performance. In this paper, we aim simply to demonstrate a minimal data-driven Bayesian model, where the emphasis is on getting information from the data, rather than adding detailed human knowledge. The hierarchical model that we have described puts a qualitative prior over the parameters Q (qualitative in that the form of a Dirichlet distribution is specified, but without specifying quantitative values for the hyperparameters $\alpha\mathbf{m}$); this is effectively turned into a quantitative prior by consulting the data. This general approach is sometimes called 'empirical Bayes'. Our method is distinguished from many empirical Bayes prescriptions in that (a) we use Bayesian inference to control the hyperparameters;

(b) we motivate this procedure as an approximation to the ideal predictive Bayesian approach.

2.6 Inference and prediction using the hierarchical Dirichlet model

It is convenient to distinguish two levels of inference. We are interested in the plausible values of (at level 1) the parameters, $Q = \{q_{i|j}\}$ and (at level 2) the 'hyperparameters' $\alpha\mathbf{m}$. We use the results of section 2.4.

2.6.1 Level 1 inference

At level 1, we assume we know $\mathbf{m}$ and α. It is then easy to infer a posterior distribution for Q, and get a predictive distribution. By Bayes' theorem the posterior distribution is

$$(19)\qquad P(Q|D,\alpha\mathbf{m},\mathcal{H}_D) = \frac{P(D|Q,\mathcal{H}_D)P(Q|\alpha\mathbf{m},\mathcal{H}_D)}{P(D|\alpha\mathbf{m},\mathcal{H}_D)}.$$

This distribution is separable into a product over contexts j, because both the prior $P(Q|\alpha\mathbf{m},\mathcal{H}_D)$ and the likelihood $P(D|Q,\mathcal{H}_D)$ are separable.

$$(20)\qquad P(Q|D,\alpha\mathbf{m},\mathcal{H}_D) = \prod_j P(\mathbf{q}_{|j}|D,\alpha\mathbf{m},\mathcal{H}_D).$$

The posterior distribution of each conditional probability vector is simply another Dirichlet distribution:

$$(21)\quad P(\mathbf{q}_{|j}|D,\alpha\mathbf{m},\mathcal{H}_D) \propto \prod_i q_{i|j}^{F_{i|j}+\alpha m_i - 1}\delta(\textstyle\sum_i q_{i|j}-1) = \text{Dirichlet}^{(I)}(\mathbf{q}_{|j}|\mathbf{F}+\alpha\mathbf{m}).$$

This posterior can be used for prediction:

$$(22)\qquad P(i|j,D,\alpha\mathbf{m},\mathcal{H}_D) = \frac{F_{i|j}+\alpha m_i}{\sum_{i'} F_{i'|j}+\alpha m_{i'}}.$$

Notice that $\mathbf{m}$ is taking precisely the role of the marginal statistics in equation (1). To make this explicit, the predictive distribution can be written:

$$(23)\qquad P(i|j,D,\alpha\mathbf{m},\mathcal{H}_D) = \lambda_j m_i + (1-\lambda_j)f_{i|j},$$

where $f_{i|j} = F_{i|j}/F_j$ and

$$(24)\qquad \lambda_j = \frac{\alpha}{F_j+\alpha}.$$

Note that, in contrast to λ in equation (1), this quantity λ_j is not constant. It varies inversely with the frequency of the given context j. Practitioners of deleted interpolation have, as mentioned in the introduction, found it useful to divide the contexts j into different groups, according to their frequency F_j, with a separate λ for each group. Each λ has to be optimized using deleted interpolation. Here, simply by turning the handle of Bayesian inference, we have produced a smoothing prescription which, we anticipate, eliminates this need to group contexts by their frequency. The appropriate variation of λ with F_j is automatically present in (24). (Not that this is a new idea: the 'blending' method in text compression (Bell *et al.* 1990) uses the same variation with F_j.)

2.6.2 Level 2 inference

At the second level of inference, we infer the hyperparameters given the data. The posterior distribution of $\alpha\mathbf{m}$ is, by Bayes' theorem:

$$(25) \qquad P(\alpha\mathbf{m}|D, \mathcal{H}_D) = \frac{P(D|\alpha\mathbf{m}, \mathcal{H}_D)P(\alpha\mathbf{m}|\mathcal{H}_D)}{P(D|\mathcal{H}_D)}.$$

The data-dependent term $P(D|\alpha\mathbf{m}, \mathcal{H}_D)$ is the normalizing constant from the first level of inference (19). We call it the *evidence* for $\alpha\mathbf{m}$. We will proceed by finding the maximum $[\alpha\mathbf{m}]^{\mathrm{MP}}$ of the posterior distribution $P(\alpha\mathbf{m}|D, \mathcal{H}_D)$. The ideal Bayesian method would put a proper prior on the hyperparameters and marginalize over them when making predictions:

$$(26) \qquad P(i|j, D, \mathcal{H}_D) = \int P(\alpha\mathbf{m}|D, \mathcal{H}_D)P(i|j, D, \alpha\mathbf{m}, \mathcal{H}_D)\, d^W(\alpha\mathbf{m}).$$

However, if (as we expect) the posterior distribution $P(\alpha\mathbf{m}|D, \mathcal{H}_D)$ is sharply peaked in $\alpha\mathbf{m}$ so that it is effectively a delta function in (26), relative to $P(i|j, D, \alpha\mathbf{m}, \mathcal{H}_D)$, then we may approximate:

$$(27) \qquad P(i|j, D, \mathcal{H}_D) \simeq P(i|j, D, [\alpha\mathbf{m}]^{\mathrm{MP}}, \mathcal{H}_D).$$

So instead of marginalizing over the hyperparameters, we optimize them; the optimization is computationally more convenient, and often gives predictive distributions that are indistinguishable from the true predictive distribution (MacKay 1995c). We are assuming a noninformative prior $P(\alpha\mathbf{m}|\mathcal{H}_D)$, so the posterior probability maximum $[\alpha\mathbf{m}]^{\mathrm{MP}}$ is found by maximizing the evidence $P(D|\alpha\mathbf{m}, \mathcal{H}_D)$. If the accuracy of this approximation is doubted in any specific case, then the correct marginalization over the hyperparameters can be performed by, for example, Monte Carlo methods (see, for example, Neal (1992, 1993) and (West 1992)). We note in passing that the mode of a posterior probability distribution does not have any fundamental status in Bayesian inference, and its location can be changed arbitrarily by a non-linear reparameterization. The maximum of the evidence, on the other hand, is invariant under reparameterization.

Now, the question is, will $\mathbf{m}^{\mathrm{MP}}$ turn out equal to the marginal statistics f_i? If it did, then this Bayesian procedure would reproduce the predictions of smoothing. The answer is, no, the optimal measure is different. This will be discussed first using a toy example to persuade the reader that equation (1) is unsatisfactory. The mathematics of the Bayesian optimization of $\alpha\mathbf{m}$ will then be worked out in detail.

Example: A data set for which equation (1) is evidently unsatisfactory.

Imagine, you see, that the language, you see, has, you see, a frequently occurring couplet, 'you see', you see, in which the second word of the couplet, see, follows the first word, you, with very high probability, you see. Then the marginal statistics, you see, are going to become hugely dominated, you see, by the words you and see, with equal frequency, you see.

Now given this data set, what is the conditional probability of each word if and when a novel context occurs? In particular, what are the probabilities of the words 'you' and 'see'? Where the Dirichlet model (23) would assign probabilities m_i^{MP}, the smoothing formula (1) would assign probabilities proportional to f_i. So using the smoothing formula, the predictions $\hat{P}(\text{you}|\text{novel})$ and $\hat{P}(\text{see}|\text{novel})$ would come out equal, since 'you' and 'see' have both occurred equally often (11 times) so far. But is this intuitively reasonable? 'You' evidently has a relatively high probability in any context, whereas 'see' only has a high frequency because it has a high probability of following 'you'. Thus intuitively $P(\text{you}|\text{novel})$ should be greater than $P(\text{see}|\text{novel})$. We would like the probability of a word to relate not to its raw frequency, but rather to *the number of contexts in which it has occurred.* We will see shortly that $\mathbf{m}^{\text{MP}}$ does exactly this.

It should be emphasized that this failure of the smoothing formula is not because of any inadequacy of the bigram model; a Markov process can easily capture the couplet in the data set above. (In text compression, the method known as 'update exclusion' (Bell *et al.* 1990) avoids the problem described above.)

3 Inferring Dirichlet hyperparameters

3.1 The dice factory

An analogy may be useful to describe the inferences we will now make. Imagine that a factory produces biased I-sided dice. We might model the probability vector $\mathbf{q}$ of a single die as coming from a Dirichlet prior with unknown hyperparameters $\mathbf{u} = \alpha\mathbf{m}$ that characterize the factory. The data are the outcomes of rolls of J dice labelled by j. Each die j is rolled a number of times F_j, and we are told the counts of the outcomes, $F_{i|j}$, which give us imperfect information about the parameters $Q = \{q_{i|j}\}$. Given these measurements, our task is to infer the hyperparameters $\mathbf{u} = \alpha\mathbf{m}$ of the factory, in order to make better predictions about future rolls of individual dice.

This problem is identical to the language modelling problem, where the number of dice J and the number of classes I are both equal to the number of words W. We can imagine the language being generated by a dice rolling procedure in which the outcome of roll t determines which die is rolled at time $t + 1$. Other inference problems, in genome modelling for example, can also be related to the inference of models for dice factories.

3.2 The evidence for $\alpha\mathbf{m}$

The posterior probability of $\alpha\mathbf{m}$ is proportional to $P(D|\alpha\mathbf{m}) = \prod_j P(\mathbf{F}_{|j}|\alpha\mathbf{m})$, which we obtain from equation (16):

$$P(D|\alpha\mathbf{m}) = \prod_j \left(\frac{\prod_i \Gamma(F_{i|j} + \alpha m_i)}{\Gamma(F_j + \alpha)} \frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha m_i)} \right). \quad (28)$$

We now work in terms of $u_i = \alpha m_i$. To find the most probable $\mathbf{u} = \alpha\mathbf{m}$, we differentiate, using digamma functions defined by $\Psi(x) \equiv \partial \log \Gamma(x)/\partial x$. (The motivation for evaluating the gradient is that the optimization of a continuous function of many variables u_i is easiest if the gradient of the function is calculated.)

$$(29)\quad \frac{\partial}{\partial u_i}\log P(D|\mathbf{u}) = \sum_j \left[\Psi(F_{i|j}+u_i) - \Psi(F_j + \textstyle\sum_{i'} u_{i'}) + \Psi(\textstyle\sum_{i'} u_{i'}) - \Psi(u_i)\right].$$

This gradient may be fed into any optimization program to find $\mathbf{u}^{\mathrm{MP}}$. A conjugate gradients algorithm (Press, Flannery, Teukolsky and Vetterling 1988), for example, easily finds the optimum. However, we can obtain further insight and derive an explicit optimization algorithm by making some approximations.

3.3 Inferring $\mathbf{u} = \alpha\mathbf{m}$*—approximations for* $u_i < 1$ *and* $\alpha > 1$

We now assume that $\alpha > 1$ and $u_i < 1$ to derive an algorithm specialized for the parameter regimes expected in language modelling. We expect α to be greater than 1 because α corresponds to the rough number of data points needed to overwhelm the Dirichlet prior. How many times F_j do we expect we need to see context j for us to have learnt the principal properties of $\mathbf{q}_j$? If we have only seen a context one or two times, then we intuitively expect our prior knowledge of the high frequency of the word 'the', for example, to still be important. But once we have seen a context a few tens or hundreds of times we expect that the observed counts will differ significantly from the default distribution. And in preliminary experiments we did find that the most probable α ranged from about 1.4 to about 60. Now since the m_i's sum to one, a typical m_i will be 1/(size of vocabulary), therefore $u_i = \alpha m_i$ can be expected to be less than 1.

We use the relationship $\Psi(x+1) = \Psi(x) + \frac{1}{x}$ to combine the first and fourth terms in equation (29):

$$(30)\quad \Psi(F_{i|j}+u_i) - \Psi(u_i) = \frac{1}{F_{i|j}-1+u_i} + \frac{1}{F_{i|j}-2+u_i} + \cdots + \frac{1}{2+u_i} + \frac{1}{1+u_i} + \frac{1}{u_i}.$$

The number of terms in this sum is $F_{i|j}$. Assuming u_i is smaller than 1 we can approximate this sum, for $F_{i|j} \geq 1$, by

$$(31)\qquad 1/u_i + \sum_{f=2}^{F_{i|j}} 1/(f-1) - u_i \sum_{f=2}^{F_{i|j}} 1/(f-1)^2 + O(u_i^2).$$

Approximating the other terms $\Psi(\alpha) - \Psi(F_j + \alpha)$ with equation (39), we obtain the following prescription for the maximum evidence hyperparameters $\mathbf{u}^{\mathrm{MP}}$. For each F and i, let N_{Fi} be the number of contexts j such that $F_{i|j} \geq F$, and let $F_i^{\max}$ be the largest F such that $N_{Fi} > 0$. Denote the number of entries in row 'i' of $F_{i|j}$ that are non-zero, N_{1i}, by V_i. Compute the quantities:

$$(32)\qquad G_i = \sum_{f=2}^{F_i^{\max}} N_{fi}/(f-1)$$

$$(33)\qquad H_i = \sum_{f=2}^{F_i^{\max}} N_{fi}/(f-1)^2$$

and define:

$$K(\alpha) = \sum_j \log\left[\frac{F_j+\alpha}{\alpha}\right] + \frac{1}{2}\sum_j\left[\frac{F_j}{\alpha(F_j+\alpha)}\right], \tag{34}$$

then the optimal hyperparameters $\mathbf{u}$ satisfy the implicit equation:

$$u_i^{\mathrm{MP}} = \frac{2V_i}{K(\alpha^{\mathrm{MP}}) - G_i + \sqrt{(K(\alpha^{\mathrm{MP}}) - G_i)^2 + 4H_iV_i}}. \tag{35}$$

This defines a one-dimensional problem: to find the α such that the u_i given by (35) satisfy $\sum_i u_i = \alpha$. This optimal α can be found by a bracketing procedure or by a reestimation procedure in which we alternately use (35) to set u_i given α and then set $\alpha := \sum_i u_i$. We use this algorithm in section 4.

3.4 Comments thus far

- A predictive algorithm similar to 'smoothing' has been derived within a fully probabilistic model.
- The smoothing vector is not the marginal distribution, as used in the traditional language model. Rather, we see from the numerator in equation (35) that u_i is directly related to V_i, the number of contexts in which word i has occurred, thereby satisfying the desideratum raised by the toy example of the previous section.
- The weight λ of the smoothing vector scales automatically with the number of counts. There is no need to separate words into separate categories depending on their raw frequency.
- This framework does not involve cross-validation; all the data is devoted to every aspect of the modelling process.
- This perspective reveals the crudity of the implicit model underlying smoothing: all the conditional probability vectors in the matrix Q are modelled as coming from a single Dirichlet distribution. This distribution is characterized only by a mean probability vector $\mathbf{m}$ and a single scalar measure of spread about this mean, α. It seems plausible, even if a bigram model is assumed, that a more complex distribution for $\mathbf{q}_{|j}$ might give a better model. For example, as discussed later in section 5, we might believe that contexts come in equivalence classes or types – this would motivate a mixture model for the vectors $\mathbf{q}_{|j}$.

4 Application to a small corpus

We conducted an experiment to compare deleted interpolation with the new method empirically (Peto 1994). We used each algorithm to construct an alternative model from the training corpus. We then compared the predictive accuracy of the algorithms by evaluating the perplexity of the test data under each of the competing models: the better the model, the smaller the perplexity.

Perplexity is defined as $2^{H(Q;\hat{P})}$, where $H(Q;\hat{P})$ is the cross-entropy between the unknown 'true' model Q and the assumed model $\hat{P}$. (For the case of two distributions over alternatives i, $H(Q;\hat{P}) = \sum_i Q_i \log_2 \hat{P}_i$.) For the bigram models we use, and a large enough test corpus, the perplexity of the test corpus can be approximated by

$$\text{(36)} \qquad \text{Perplexity} \simeq \left[\prod_{t=2}^{T} \hat{P}(w_t|w_{t-1})\right]^{-1/T},$$

where T is the number of words in the corpus (see Brown *et al.* 1992).

The training and test corpora were taken from the English portion of Gale and Church's (1991) sentence-aligned version of the Canadian *Hansard*, the proceedings of the Canadian Parliament. This text had already been separated into sentences and stripped of titles, formatting codes and speaker identifiers. We removed sentence numbers and added sentence-begin and sentence-end markers. In keeping with common practice for experiments of this type, we split off punctuation and suffixes beginning with apostrophes from the words they followed, making them separate tokens. In order to reduce the total number of types in the vocabulary, we also replaced each number by the special token "#".

The resulting sentences were distributed into nine blocks of about 1.7 Mbytes each, with consecutive sentences going to different blocks. This interleaving of the sentences performs the important function of homogenizing the data: otherwise significant differences in token frequencies could result from different portions of the corpus as different topics were being discussed. The first six blocks were used for training data (about 2 million words), and the test data were extracted from the remaining three blocks.

We prepared three different test samples from the test data. Because the algorithms being compared only assign probabilities to bigrams composed of tokens that appear in the training data, they have no way of dealing with previously unseen tokens (we chose not to address the important zero-frequency problem in this study). Therefore, we removed all sentences that contained a token that did not occur in the training data. This left 14,393 sentences (about 260,000 tokens) in Sample 1. Next, recognizing that the *Hansard* contains many conventional phrases and sentences that might skew the results of the experiment, we removed from Sample 1 all sentences that were duplicated in either the test data or the training data. This left 12,000 sentences (about 243,000 tokens) in Sample 2. Finally, to test whether the sample was large enough for the approximation of perplexity in equation (36) to hold, we pseudo-randomly chose half the sentences in Sample 2 to become Sample 3 (6000 sentences, about 116,000 tokens).

The two algorithms have different numbers of parameters to be optimized. For the deleted interpolation method, the number of λs is chosen subjectively. We ran the deleted interpolation method with 3, 15 and 150 λs to judge the effect of this choice. In the hierarchical model presented in this paper, there is one hyperparameter u_i for each type in the training data vocabulary.

The experiment was conducted as follows. First, raw frequencies and relative frequencies of tokens and bigrams were obtained from the training data as a whole. Next, the most probable values for the parameters of each model were solved for iteratively. For the Dirichlet model, this meant solving the simultaneous equations

Table 1. *Perplexities of the three test data samples under the different models. T = number of tokens in sample.*

		Algorithm			
		Deleted interpolation			Dirichlet
Sample	$T/1000$	3 λs	15 λs	150 λs	
1	260		79.60		79.90
2	243	89.57	88.47	88.91	89.06
3	116		91.82		92.28

given by equation (35) to obtain $\mathbf{u}^{\mathrm{MP}}$. For the smoothing method, separate frequencies were first calculated for each block, and then the λ's were obtained using deleted interpolation (Jelinek and Mercer 1980). The optimization was halted when on average each parameter of the model had converged to eight decimal places. The optimized parameter values for each model were then used to compute predictive probabilities $\hat{P}(i|j)$ for each bigram in the test data. Finally, the perplexity of each of the three test data samples was evaluated using each of the models, and the results were compared.

The perplexity of each test sample under each model is given in Table 1. For all three samples, the perplexities under the deleted interpolation model and under the Dirichlet model are nearly the same.

For Sample 2, three deleted interpolation models having different numbers of λs were tested. The effect of altering the number of λs was very small. When 150 λs were used, we found that the values of λ decreased with the frequency F_j roughly as expected from equation (24).

Finally, the perplexity results for the smaller Sample 3 are close to the corresponding results for Sample 2. This suggests that Sample 2 is large enough to provide a meaningful comparison of models. The fact that the perplexity results for Sample 1 are lower than those of Sample 2 probably reflects the high degree of regularity of the extra (conventional) data more than the small increase in test data size.

With regard to resource use, the new algorithm has an advantage. The number of iterations required for each algorithm to converge was comparable. However, a single iteration of our Dirichlet model requires time linear in the size of the vocabulary, while an iteration of deleted interpolation requires time linear in the size of the training corpus. The larger the training corpus, the more significant would be this advantage of the Dirichlet model. Also, deleted interpolation requires more memory because it keeps separate count and frequency data for each block of the training corpus. In our implementation there were six such blocks.

We have not made a direct comparison with the 'backing-off' algorithm because Katz's (1987) results indicate that backing-off is indistinguishable in performance from deleted interpolation on a similar bigram modelling task.

5 Discussion

The exercise of creating a Bayesian version of the 'smoothing' procedure has given several benefits. (1) The Dirichlet model is not identical to smoothing; the differences are intuitively reasonable. (2) The Dirichlet model does away with cross-validation and therefore makes full use of the data while requiring fewer computational resources.

We would like to distinguish the general Bayesian method from the particular hierarchical Bayesian model discussed in this paper, and the computational approximations used to implement it. We emphatically do not view the presented algorithm as *the* Bayesian answer to language modelling, nor do we claim that this particular algorithm will necessarily be superior to deleted interpolation in any given application. There are many possible Bayesian language models, and the one we have studied is virtually the simplest possible. We now discuss other possible models.

5.1 Generalizations

Language modelling has here been viewed as the modelling of a set of probability vectors $\mathbf{q}_{|j}$ drawn from a coupled density over the simplex (the simplex is the space of probability vectors $\mathbf{q}$, satisfying $q_i \geq 0$ and $\sum_i q_i = 1$), with one probability vector $\mathbf{q}_{|j}$ for each context j. In this paper's model, the context j is simply the previous word, and the density over the simplex is a single Dirichlet distribution parameterized by $\alpha\mathbf{m}$.

The two simplest modifications to this model are to change the functional form of the density over the simplex, and to change the definition of a context.

An alternative density over probabilities to the Dirichlet distribution is the **entropic prior** (Skilling 1989; Gull 1989),

$$P(\mathbf{p}|\alpha\mathbf{m}) = \frac{1}{Z}\frac{1}{\prod_i p_i^{1/2}} \exp\left(\alpha \sum_i p_i \log \frac{m_i}{p_i}\right) \delta\left(\sum p_i - 1\right). \tag{37}$$

The entropic prior, like the Dirichlet prior, characterizes a language by a single mean and spread of a distribution of conditional probabilities $\mathbf{q}_{|j}$ for all contexts j. Recent work at IBM on 'maximum entropy language modelling' (S. & V. Della Pietra, personal communication) might be interpreted in terms of an entropic prior. This interpretation could then be used to obtain a Bayesian prescription for α and $\mathbf{m}$, as this paper has done for the Dirichlet model.

A more interesting model might assert that there are different *types* of context, such that for all contexts of the same type, the conditional probabilities $\mathbf{q}$ are similar. If we do not know *a priori* what the type of each context is, then this model is a **mixture model**. A mixture model $\mathcal{H}_M$ defines a density over $\mathbf{q}$ as a weighted combination of C independently parameterized simple distributions, where each mixture component $c = 1 \ldots C$ might be a Dirichlet or entropic distribution. Various algorithms can be used to implement mixture models: both Monte Carlo methods (Neal 1992) and Gaussian approximations (Hanson, Stutz and Cheeseman 1991).

Mixture models are applied to the modelling of amino acid probabilities in Mackay (1995d).

Alternatively, a model might define the context to be the last two words, with the type of the context being defined by the most recent word. With a coupled prior for the context hyperparameters, this model would give predictions similar to those of the smoothed trigram language model.

The mixture model is also able to capture the same clustered structure as the hierarchical trigram model, but has the potential advantage that it can discover other relationships between the contexts; for example, if it happens to be the case that the last word but one is sometimes more important than the last word in characterizing $\mathbf{q}_{|j}$, then the more flexible mixture model can capture this structure in the data.

Finally, we might believe that the type of a context is more naturally described with a **componential** structure (G. Hinton, personal communication; see also Williams and Hinton 1991). Imagine, for example, that any context is either legalistic or not; and that in any context, either a verb is likely, or is unlikely. A traditional mixture model would have to use four mixture components to capture these two sources of variation; and in general we would need a number of mixture components exponential in the dimensionality of the context space; whereas we might believe that the number of parameters needed to describe the probability distribution ought only to be linear in that number of dimensions. This motivates the development of componential models (a type of latent variable model), in which the type of a context is represented with several continuous or discrete dimensions. A componential model is described and applied to the modelling of amino acid probabilities in Mackay (1995d). It has been generalized to the modelling of joint distributions of multiple amino acids in MacKay (1995a, 1995b).

5.2 Relationship to previous 'empirical Bayes' approaches

An approach similar in spirit to the one advocated in this paper has been described by Nadas (1984). His 'empirical Bayes' approach also interprets the smoothing formula (1) in terms of a prior whose hyperparameters are determined from the data.

In contrast to the present paper, however, Nadas at several points 'chooses' estimators in an arbitrary way (in the fully Bayesian approach there are no choices, only mechanistic inferences). Another weakness of Nadas' paper is that the prior that is considered is a technically inappropriate prior that neglects normalization of the probability vectors $\mathbf{q}_{|j}$.

The technique of smoothing is also used in modelling with classification trees, and this literature contains a similar paper in which an 'empirical Bayes' approach is used (Buntine 1992). As above, this approach is compromised by the invocation of ad hoc estimators, instead of the derivation of inferences. An estimator for $\mathbf{m}$ is given that is not, in fact, the most probable $\mathbf{m}$. No objective procedure for setting α is given.

A fully Bayesian approach to the hyperparameter α has been given by West (1992), along with a Monte Carlo algorithm for Gibbs sampling of this hyperparameter.

The advantages of a fully Bayesian attitude to data modelling are, firstly, that one is forced to make all one's assumptions explicit; and secondly, that once the model is defined, all inferences and predictions are mechanically defined by the rules of probability theory.

Acknowledgements

DJCM thanks Peter Brown, Radford Neal, Geoff Hinton, Phil Woodland, David Robinson, Martin Oldfield, Steve Gull, John Bridle and Graeme Mitchison for helpful discussions, and the Isaac Newton Institute for hospitality.

LCBP thanks Bill Gale, Radford Neal and Peter Brown for helpful discussions.

Appendix A: The Gamma function and Digamma function

The Gamma function is defined by $\Gamma(x) \equiv \int_0^\infty du\, u^{x-1}e^{-u}$, for $x > 0$. In general, $\Gamma(x+1) = x\Gamma(x)$, and for integer arguments, $\Gamma(x+1) = x!$. The digamma function is defined by $\Psi(x) \equiv \frac{d}{dx}\log\Gamma(x)$.

For large x (for practical purposes, $0.1 \leq x \leq \infty$), the following approximations are useful:

$$\log\Gamma(x) \simeq \left(x - \tfrac{1}{2}\right)\log(x) - x + \tfrac{1}{2}\log 2\pi + O(1/x) \tag{38}$$

$$\Psi(x) = \frac{d}{dx}\log\Gamma(x) \simeq \log(x) - \frac{1}{2x} + O(1/x^2). \tag{39}$$

And for small x (for practical purposes, $0 \leq x \leq 0.5$):

$$\log\Gamma(x) \simeq \log\frac{1}{x} - \gamma_e x + O(x^2) \tag{40}$$

$$\Psi(x) \simeq -\frac{1}{x} - \gamma_e + O(x), \tag{41}$$

where γ_e is Euler's constant. The digamma function satisfies the following recurrence relation exactly:

$$\Psi(x+1) = \Psi(x) + \frac{1}{x}. \tag{42}$$

Formula for a more general algorithm

The algorithm presented in this paper is based on series expansions of $\Psi(u)$ and is not valid for all u. The following formula, although it is not part of a series expansion, gives an approximation to the difference $\Psi(F+u) - \Psi(u)$ that is accurate to within 2% for all u and all positive integers F:

$$\Psi(F+u) - \Psi(u) \simeq \frac{1}{u} + \log\frac{F+u-1/2}{u+1/2}. \tag{43}$$

This approximation is useful for gradient-based optimization of Dirichlet distributions (MacKay 1995d).

References

Antoniak, C. E. (1974) Mixtures of Dirichlet processes with applications to nonparametric problems. *Annals of Statistics* **2**: 1152–1174.

Bahl, L. R., Brown, P., de Souza, P., Mercer, R. L. and Nahamoo, D. (1991) A fast algorithm for deleted interpolation. *Proceedings of Eurospeech '91 Genoa*, pp. 1209–1212.

Bahl, L. R., Jelinek, F. and Mercer, R. L. (1983) A maximum likelihood approach to continuous speech recognition. *IEEE Trans PAMI* **5** (2): 179–190.

Bell, T. C., Cleary, J. G. and Witten, I. H. (1990) *Text compression*. Englewood Cliffs, NJ: Prentice Hall.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C. and Mercer, R. L. (1992) An estimate of an upper bound for the entropy of English. *Computational Linguistics* **18** (1): 31–40.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J. and Mercer, R. L. (1993) The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* **19** (2): 263–311.

Buntine, W. (1992) Learning classification trees. *Statistics and Computing* **2**: 63–73.

Cox, R. (1946) Probability, frequency, and reasonable expectation. *Am. J. Physics* **14**: 1–13.

Gale, W. and Church, K. (1991) A program for aligning sentences in bilingual corpora. *Proceedings of 29th Annual Meeting of the ACL*, pp. 177–184.

Gull, S. F. (1989) Developments in maximum entropy data analysis. In *Maximum Entropy and Bayesian Methods, Cambridge 1988*, J. Skilling (ed.), pp. 53–71. Dordrecht: Kluwer.

Hanson, R., Stutz, J. and Cheeseman, P. (1991) Bayesian classification with correlation and inheritance. *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia*, volume 2, pp. 692–698. San Mateo, CA: Morgan Kaufmann.

Jelinek, F. and Mercer, R. L. (1980) Interpolated estimation of Markov source parameters from sparse data. In *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal (eds.), pp. 381–402. Amsterdam: North–Holland.

Katz, S. M. (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing* **35** (3): 400–401.

MacKay, D. J. C. (1995a) Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research, Section A* **354** (1): 73–80.

MacKay, D. J. C. (1995b) Density networks and protein modelling. In *Maximum Entropy and Bayesian Methods, Cambridge 1994*, J. Skilling and S. Sibisi (eds.), Dordrecht: Kluwer.

MacKay, D. J. C. (1995c) Hyperparameters: Optimize, or integrate out? In *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, G. Heidbreder (ed.), Dordrecht: Kluwer.

MacKay, D. J. C. (1995d) Models for dice factories and amino acid probability vectors. In preparation.

Nadas, A. (1984) Estimation of probabilities in the language model of the IBM speech recognition system. *IEEE Trans ASSP* **32** (4): 859–861.

Neal, R. M. (1992) Bayesian mixture modelling. In *Maximum Entropy and Bayesian Methods, Seattle 1991*, C. Smith, G. Erickson and P. Neudorfer (eds.), pp. 197–211. Dordrecht: Kluwer.

Neal, R. M. (1993) Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG–TR–93–1, Dept. of Computer Science, University of Toronto.

Peto, L. B. (1994) A comparison of two smoothing methods for word bigram models. Technical Report CSRI-304, Computer Systems Research Institute, University of Toronto.

Press, W., Flannery, B., Teukolsky, S. A. and Vetterlıng, W. T. (1988) *Numerical Recıpes in C*. Cambridge: Cambridge University Press.

Skilling, J. (1989) Classic maximum entropy. In *Maximum Entropy and Bayesian Methods, Cambridge 1988*, J. Skilling (ed.), Dordrecht: Kluwer.

West, M. (1992) Hyperparameter estimation in Dirichlet process mixture models. Working paper 92-A03, Duke Inst. of Stats. and Decision Sciences.

Williams, C. K. I. and Hinton, G. E. (1991) Mean field networks that learn to discriminate temporally distorted strings. In *Connectionist Models: Proceedings of the 1990 Summer School*, D. S. Touretzky, J. L. Elman and T. J. Sejnowski (eds.). San Mateo, CA: Morgan Kaufmann.