Fitting maximum-entropy models on large sample spaces

Edward Schofield

A DISSERTATION

29th June 2006; revised 29th January 2007

PhD thesis submitted to the Department of Computing, Imperial College London.

Internal supervisor:

Dr. Stefan Rüger
Department of Computing
Imperial College London
United Kingdom

External supervisor:

Prof. Gernot Kubin
SPSC Laboratory
Graz University of Technology
Austria

Abstract

This thesis investigates the iterative application of Monte Carlo methods to the problem of parameter estimation for models of maximum entropy, minimum divergence, and maximum likelihood among the class of exponential-family densities. It describes a suite of tools for applying such models to large domains in which exact computation is not practically possible.

The first result is a derivation of estimators for the Lagrange dual of the entropy and its gradient using importance sampling from a measure on the same probability space or its image under the transformation induced by the canonical sufficient statistic. This yields two benefits. One is the flexibility to choose an auxiliary distribution for sampling that reduces the standard error of the estimates for a given sample size. The other is the opportunity to re-weight a fixed sample iteratively, which can cut the computational burden for each iteration.

The second result is the derivation of matrix–vector expressions for these estimators. Importance-sampling estimates of the entropy dual and its gradient can be computed efficiently from a fixed sample; the computation is dominated by two matrix–vector products involving the same matrix of sample statistics.

The third result is an experimental study of the application of these estimators to the problem of estimating whole-sentence language models. The use of importance sampling in conjunction with sample-path optimization is feasible whenever the auxiliary distribution does not too severely under-represent any linguistic features under constraint. Parameter estimation is rapid, requiring a few minutes with a 2006-vintage computer to fit models under hundreds of thousands of constraints. The procedure is most effective when used to minimize divergence (relative entropy) from existing baseline models, such as n-grams estimated by traditional means, rather than to maximize entropy under constraints on the probabilities of rare n-grams.

Acknowledgments

I wish to thank:

- Gernot Kubin, for countless hours of dedicated and thoughtful criticism, and for inspiration;
- Stefan Rüger, for his guidance and encouragement;
- David Thornley, Tomas Nordström, Andreas Türk, and Desmond Lun, for their valuable suggestions on the draft;
- Christoph Mecklenbräuker and Himanshu Tyagi, for patiently sharing their mathematical expertise;
- Markus Kommenda, Horst Rode, and my colleagues at the Telecommunications Research Center Vienna (FTW), for providing a supportive research environment;
- FTW, K-*plus*, Mobilkom Austria, Kapsch CarrierCom, and the Marie Curie Fellowship program of the European Commission, for funding my research;
- Travis Oliphant and John Hunter, for their superb tools for scientific computing;
- Chris and Suvina Town, Tomas and Gisela Nordström, and Joachim and Margarethe Wehinger, for their strong friendship;
- María Hermoso-Cristobal, for not taking me seriously;
- Heike Sauer and my dear family, for their love.

Contents

1 Introduction

In 1992, Geyer and Thompson were praised by a reviewer from the Royal Statistical Society for demonstrating the advantages of "using complex models and approximate methods of computation instead of the more usual combination of exact calculation and oversimple and, hence, inappropriate models." A trend has been developing in statistical inference, undoubtedly encouraged by the growth and spread of inexpensive computing power, towards to use of approximate methods of computation. The complexity of the models used and the size of the problems approached are growing in a kind of arms race with the available infrastructure. This thesis brings some new weapons to the armory built on maximum entropy and Monte Carlo methods.

The principle of maximum entropy states that, of all models not ruled out by constraints, the model of maximum entropy is the most appropriate for inductive reasoning. This has a solid axiomatic justification and intuitive appeal as a "reasonable and systematic way of throwing up our hands" in the presence of uncertainty [Benes, 1965]. The principle can be re-formulated as a tool for modelling rich structural relationships in data by representing prior information as constraints on the set of feasible models and adopting the entropy-maximizing model from among this set. This is a highly flexible structure that has proved itself useful in a wide range of modelling tasks, from spectral analysis for oil exploration [Burg, 1967, 1975] to characterizing DNA sequences [Bugaenko et al., 1998].

One cost imposed by this flexibility is the need to use iterative methods to fit the resulting models. Most applications of maximum entropy to date have involved small discrete spaces, for which the computational burden is usually manageable. This thesis defines a sample space as 'small' if integration over the space is practical without Monte Carlo simulation. The focus of the thesis is, instead, on 'large' spaces

that do require Monte Carlo methods of integration.

The greatest advantage of Monte Carlo integration is that it scales well to large numbers of dimensions in continuous spaces (in sharp contrast, for example, to numerical methods of 'quadrature'). Its greatest disadvantage is the high computational burden it can impose for achieving even moderate precision. My first goal for this thesis is to show how to construct Monte Carlo estimators of sufficiently low variance that fitting exponential-family models on large sample spaces becomes feasible. My second goal is to show how this task can be performed efficiently.

1.1 Outline

This thesis is structured as follows:

Chapter 2 describes the history of maximum-entropy inference, its applications to problems in science and engineering, and its benefits and drawbacks. It also reviews the exponential-family form of maximum-entropy models and the theory of parameter estimation for small sample spaces.

Chapter 3 describes Monte Carlo integration for exponential-family models whose parameters are fixed. It describes various forms of Monte Carlo estimators of the expected canonical statistic, with a focus on importance sampling and conditioning as means to reduce the variance of the Monte Carlo estimators to manageable levels.

Chapter 4 describes parameter estimation for large exponential-family models as a stochastic optimization problem and reviews some relevant literature in stochastic optimization. It formulates the Monte Carlo estimators from Chapter 3 in terms of matrix–vector operations involving logarithms of sums of exponentials. This allows efficient iterative estimation of the entropy dual and gradient without numerical overflow.

Chapter 5 tests the applicability of the theory to fitting whole-sentence language

models. It shows that a combination of importance sampling and sample path optimization allows highly efficient parameter estimation. It also shows that rare features can make parameter estimation problematic, and that reformulating the task in terms of relative-entropy minimization can help to sidestep this problem.

Chapter 6 tests the theory on the continuous modelling problem of estimating truncated Gaussian models in several dimensions. It is also a short illustration of how the scope of the estimation framework of Chapters 3 and 4 is all models, discrete or continuous, of exponential-family form.

Chapter 7 draws some conclusions on the scalability of maximum-entropy inference to large sample spaces using the ideas developed in the thesis and summarizes some recommendations for making the model-fitting process feasible and efficient.

1.2 Contributions

The following results are, to the best of my knowledge, novel—published for the first time here or in one of my papers listed below:

1. a proof that the derivative of the *integration* importance sampling estimator equals the *ratio* importance sampling estimator of the derivative (Section 3.3.1);

2. an investigation of sampling directly in the space induced by a sufficient statistic as a means of estimating its expectation with respect to a measure in the original space (Section 3.4.3), and a theorem giving consistent estimators for this in the discrete case;

3. a formulation of estimators of the entropy dual and its gradient using importance sampling and conditioning in terms of matrix–vector operations on a matrix of sufficient statistics (Section 4.4.1);

4. an alternative formulation of these matrix–vector expressions in logarithmic space for numerical stability in computer implementations (Section 4.4.2);

5. an efficient method of solving stochastic optimization problems by iteratively re-weighting importance sampling estimators in a sample-path optimization procedure (Sections 4.3 and 5.4);

6. evidence that whole-sentence models can offer significant improvement over baseline n-gram models, and a hypothesis for why significant improvements with whole-sentence models have not previously been found (Sections 5.4 and 5.5);

7. the insight that sampling from low-order n-gram language models can be performed in constant time (independent of the vocabulary size) (Section B).

An additional, non-scientific contribution to the research field is the publication of Open Source software for fitting exponential-family models on large sample spaces efficiently. This is now part of SciPy, an Open Source toolkit for scientific and engineering computing (see the Appendix). This should make the experimental results easily reproducible and reduce the barrier to adoption of these ideas into various domains in academia and industry.

I have also published some of these results in the following papers: [Schofield and Kubin, 2002; Schofield, 2003; Schofield and Zheng, 2003; Schofield, 2004].

1.3 Table of notation

The notation used in this thesis for maximum-entropy modelling is based on Jaakkola [1999], Jaakkola and Jordan [2000], Rosenfeld et al. [2001], and Malouf [2002]; the notation for importance sampling is based on the PhD thesis of Hesterberg [1988].

$\equiv$ is defined as

$\mathcal{X}$	the sample space, where $X \in \mathcal{X}$ are sentences, images, ...
ν	measure on the space $\mathcal{X}$; here either the Lebesgue measure or the counting measure
$f(x)$	vector of statistics $f_i \colon \mathcal{X} \to \mathbb{R}$, $i = 1, \ldots, m$ representing features of the observation x
$f(x) \cdot \theta$	the inner product of $f(x)$ and θ
P; $p = p_\theta$	the probability measure of the model and its corresponding density or mass function, parameterized by θ
$\dot{p}(x)$	the unnormalized model density, equal to $Zp(x) = \exp(f(x) \cdot \theta)$
Q; q	an *auxiliary* or *instrumental* probability measure from which we generate variates for importance sampling, and its corresponding density or mass function. Defined either on the sample space $\mathcal{X}$ or the feature space $\mathcal{F} = \{f(x) : x \in \mathcal{X}\}$
$\mu \equiv \mathrm{E} f(X)$	the expectation of $f(X)$ with respect to the model P
$\hat{\mu}$	an estimator of μ
$L(\theta)$	Lagrangian dual of the entropy function, subject to the moment constraints $\{\mathrm{E} f_i(X) = b_i\}_{i=1}^m$; proportional to the negative log likelihood function of the observations x under an exponential-family density p

$W(x)$	the weight function for importance sampling, defined as $p(x)/q(x)$
$\dot{W}(x)$	the unnormalized weight function, defined as $\dot{p}(x)/q(x)$
x_j, w_j	the j^{th} Monte Carlo replication of the random variables X and W
$Z = Z(\theta)$	the partition function (*Zustandssumme*) and normalization term for the model P
RV; IID	random variable; independent and identically distributed
PDF; PMF	probability density / mass function
SA; SPO	stochastic approximation; sample path optimization
CG; LMVM	conjugate gradients and limited-memory variable metric methods of optimization

2 The principle of maximum entropy

2.1 Background and motivation

The principle of maximum entropy is a principle for solving problems in statistical modelling and inference known as generalized inverse problems. Such problems arise when inferring the true state of nature that some mechanism has transformed into outputs we observe. Such outputs are, in many cases, an incomplete and perhaps noisy representation of the truth, so the problem of inversion is under-specified, meaning that several different states of nature could have accounted for the observed data. The modelling problem is to infer which are possible and to choose the most plausible from among these.

The principle instructs us to choose the model with maximal uncertainty, as measured by the entropy (usually that of Shannon [1948]), as the least-committal model among all those consistent with our prior knowledge. It is a 'Bayesian' method of inference in the sense that it acknowledges the existence of prior knowledge and states explicitly how to harness it. The principle was born in statistical mechanics with the work of Boltzmann [1887] and Gibbs [1902], and was later championed for more general scientific inference in a series of papers by Jaynes [1957, 1979, 1982, 1985, 2003]. In the words of Jaynes [1985]:

> The only way known to set up a probability distribution that honestly represents a state of incomplete knowledge is to maximize the entropy, subject to all the information we have. Any other distribution would necessarily either assume information that we do not have, or contradict information that we do have. . . .

> [The] principle of maximum entropy is not an Oracle telling which predictions *must* be right; it is a rule for inductive reasoning that tells us which predictions *are most strongly indicated by our present information*."

Jaynes' most important contribution was to see the maximum-entropy principle as something of greater generality—as a principle of reasoning rather than a principle of statistical mechanics—and to free it (Gibbs' 'canonical ensemble' method [Gibbs, 1902]) from its supposed dependence on Hamilton's equations of motion and ergodic theorems.

The theorems of Shannon and Weaver [1949] also helped Jaynes and others [e.g. Burg, 1967] to fill a gap in Gibbs' argument. Since then others have given various more axiomatic justifications of the maximum-entropy principle, including Shore and Johnson [1980]; Paris and Vencovská [1990]; Jones and Byrne [1990] and Csiszár [1991, 1996].

2.2 Relation to other methods of inference

Maximum-likelihood inference in classical statistics [e.g. Fisher, 1959] requires assumptions about the distributional form for a model, after which one seeks to find a vector of parameters that maximizes the likelihood of observing the data under these assumptions. The likelihood approach is most useful when one has frequency data but no other prior information about the quantities being estimated.

N-gram language models (to be discussed in Chapter 5) clearly demonstrate some of the limitations of maximum-likelihood estimation for practical inference problems. One limitation is that the co-occurrence frequencies of word n-grams estimated from corpora of text are difficult to reconcile with the significant body of prior knowledge that computational linguistics has accrued about language. Another limitation is that n-gram language models are dependent upon 'smoothing' heuristics[1] to redistribute probability mass from those events observed infrequently to those not observed at all, which might otherwise (incorrectly) be inferred as having probability zero.

[1] as are maximum-entropy models inferred under the mean-constraint rule (see Section 2.7)

Maximum-entropy inference, in contrast, does not assign zero probability to any situation unless the prior information rules it out. The maximum-entropy approach to modelling is, instead, to encode prior information as constraints on the set of permissible models, to let these constraints indicate a parametric form for the model, and to estimate the parameters that make the fewest additional assumptions about the process. The maximum-entropy approach is most useful when one has relevant prior information (such as a set of empirical data) with no appreciable noise (described further in Section 2.7).

Maximum likelihood and maximum entropy represent opposite extremes of reasoning, each appropriate to a distinct class of problems.[2] Neither is entirely adequate when one has both noise and prior information. Bayesian inference can be regarded as a generalization of both, with each method as a limiting case—the maximum-entropy approach in the limit of no noise, the maximum-likelihood approach in the limit of no prior information other than an assumed parametric form. The relationship is discussed further by Williams [1980], Jaynes [1985, Sec. 3], Skyrms [1985], and Uffink [1996].

2.3 Applications

The first non-thermodynamic application of Jaynes' principle was in spectral analysis for oil exploration [Burg, 1967, 1975]. Other applications since have included image reconstruction [Gull and Daniell, 1978; Barnett and MacKay, 1995], such as in radio astronomy [Press et al., 1992; Bridle et al., 1998] and tomography [Herman and Lent, 1976; Golan and Gzyl, 2002]; spectrum and chirp analysis [Jaynes, 1983, 1985]; modelling turbulence in fluids and plasmas [Montgomery, 1996]; modelling sequences of proteins [MacKay, 1996] and DNA sequences [Bugaenko et al., 1998]; several branches of natural language processing, including machine translation [Berger et al., 1996], language modelling [Rosenfeld, 1996; Rosenfeld et al., 2001], information retrieval, and parsing [Abney, 1997; Johnson et al., 1999]; and mobile communications

[2] Under certain circumstances they can, however, lead to the same conclusions (see Section 2.6).

[Debbah and Müller, 2003]. A more extensive, though somewhat dated, list is given by Shore and Johnson [1980].

2.4 Constraints and model form

The most useful formulation of maximum-entropy inference in applications to date, and the one this thesis adopts, encodes prior information in terms of linear constraints on generalized moments, of the form

$$\mathrm{E}f_i(X) = b_i \qquad \text{for } i = 1, \ldots, m, \tag{2.1}$$

where $b_i \in \mathbb{R}$ and the f_i are arbitrary statistics (features) of the data. Here and throughout the thesis the expectation operator is denoted in terms of the Lebesgue–Stieltjes integral[3] as

$$\mathrm{E}f_i(X) \equiv \int_{\mathcal{X}} f_i(x)\,\mathrm{d}P,$$

where P is a measure representing the model with respect to some measure ν. This expression subsumes both the continuous case, taking ν as the Lebesgue measure, and the discrete case, taking ν as the counting measure. In the discrete case the integral collapses to the sum $\sum f_i(x)p(x)$ over the elements x in the sample space $\mathcal{X}$.

The model P that has maximal entropy subject to these constraints can be found by introducing Lagrange multipliers θ_i, one for each constraint, and maximizing

$$\theta_0 \left[\int_{\mathcal{X}} \mathrm{d}P - 1\right] + \sum_i \theta_i \left[\int_{\mathcal{X}} f_i(x)\,\mathrm{d}P - b_i\right] + H(P) \tag{2.2}$$

where H is the differential Shannon entropy [Cover and Thomas, 1991]

$$H(P) = -\int_{\mathcal{X}} \log p(x)\,\mathrm{d}P. \tag{2.3}$$

The probability density or mass function p (hereafter just 'density') that maximizes

[3] a generalization of the Riemann and Lebesgue integration to integrals with respect to measures. See [Carter and Van Brunt, 2000] for a readable introduction.

(2.2) has the form

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(f(x) \cdot \theta), \tag{2.4}$$

[see Cover and Thomas, 1991; Papoulis, 1991], the *generalized exponential family* [e.g. Lindsey, 1996, Ch. 2], where $Z(\theta)$ is a normalization term given by

$$Z(\theta) = \int_{\mathcal{X}} \exp(f(x) \cdot \theta) \, \mathrm{d}\nu(x). \tag{2.5}$$

$Z(\theta)$ is the *Zustandssumme* or *partition function* in statistical mechanics.

The logarithm of the term $Z(\theta)$ has several useful properties [Csiszár, 1996]. The first and second-order partial derivatives of $\log Z$ are given by

$$\frac{\partial}{\partial \theta_i} \log Z(\theta) = \mathrm{E} f_i(X) \tag{2.6}$$

$$\frac{\partial^2}{\partial \theta_i^2} \log Z(\theta) = \operatorname{Var} f_i(X) \tag{2.7}$$

$$\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log Z(\theta) = \operatorname{Cov}\{f_i(X), f_j(X)\}, \tag{2.8}$$

where all three expectation operators are taken with respect to the model P. Since the covariance matrix is positive semidefinite, $\log Z(\theta)$ is a jointly convex function of the parameters $\{\theta_i\}$.

A slight generalization of this is Kullback's principle of *minimum discrimination information* [Kullback, 1959], also called the principle of *minimum relative entropy*. The relative entropy, or Kullback–Leibler divergence, between the model P and some prior P_0 is defined [see Cover and Thomas, 1991] as

Relative entropy is sometimes defined (confusingly) as the *negative* KL divergence

$$D(P \| P_0) = \int_{\mathcal{X}} \log \left(\frac{p(x)}{p_0(x)} \right) \, \mathrm{d}\nu.$$

The principle of minimum relative entropy is equivalent to the principle of maximum entropy if P_0 is the uniform distribution, whereupon minimizing $D(P \| P_0)$ reduces to minimizing the negative entropy $-H(P)$. The form of the model density for a

general prior P_0 with density p_0 becomes

$$p_\theta(x) = \frac{1}{Z(\theta; p_0)} p_0(x) \exp(f(x) \cdot \theta), \tag{2.9}$$

[see Cover and Thomas, 1991], where $Z(\theta; p_0)$ is now given by

$$Z(\theta; p_0) = \int_{\mathcal{X}} p_0(x) \exp(f(x) \cdot \theta) \, \mathrm{d}\nu(x). \tag{2.10}$$

2.5 Parameter estimation

The previous section outlined how the theory of Lagrangian duality provides a means to identify the parametric form of an entropy-maximizing distribution as that of the exponential family (2.9). This section outlines an efficient method for choosing the parameters θ to satisfy the moment constraints (2.1) based on the theory of *convex duality*.

The theory of convex duality implies that every convex function g that is sufficiently regular [Boyd and Vandenberghe, 2004] can be expressed as

$$g(\theta) = \max_\alpha \{\alpha \cdot \theta - g^*(\alpha)\}$$

in terms of its convex dual function $g^*(\alpha)$, which is also convex and satisfies

$$g^*(\alpha) = \max_\theta \{\alpha \cdot \theta - g(\theta)\}.$$

It can be shown [Jaakkola, 1999] that the negative entropy $-H(P; b)$ is the convex dual of the log partition function $\log Z(\theta)$:

$$-H(P; b) = \max_\theta \{b \cdot \theta - \log Z(\theta)\}.$$

Now notice that this variational problem has an explicit solution in terms of b. Since

$\log Z(\theta)$ is convex, the expression

$$L(\theta) \equiv \log Z(\theta) - b \cdot \theta \tag{2.11}$$

is convex, and the minimum of $L(\theta)$ occurs where its partial derivatives vanish. From (2.6) these partial derivatives are given by

$$\frac{\partial L}{\partial \theta_i} = \mathrm{E} f_i(X) - b_i, \tag{2.12}$$

so the minimum of $L(\theta)$ occurs precisely when the constraints (2.1) are satisfied. This verifies the *strong duality* of the primal entropy-maximization problem—the maximum entropy subject to the constraints (2.1) coincides with the minimum of the dual problem

$$\text{Minimize} \quad L(\theta) = \log Z(\theta) - b \cdot \theta \tag{2.13}$$

at a saddle point.

Hereafter I denote the function $L(\theta)$ the *entropy dual* or just *dual*. The dual of the relative entropy $D(P\|P_0)$ is likewise given by (2.11), with an implicit dependence of $Z(\theta)$ on the prior P_0.

Note that (2.13) is an unconstrained problem, which makes it considerably simpler to solve than the primal problem. The convexity of the dual function L also assures us that any local minimum is its unique global minimum. This provides an efficient method to set the parameters θ using any of several general-purpose nonlinear optimization methods, such as the simplex method [Nelder and Mead, 1965], methods of steepest descent and conjugate gradients [Chong and Zak, 2001], quasi-Newton methods [Boyd and Vandenberghe, 2004], interior-point methods [Saunders, 2003], or simple root-finding methods for the gradient. For comparisons of these specifically for fitting exponential-family models, see [Gull and Skilling, 1983; Malouf, 2002; Wallach, 2002; Minka, 2003; Saunders, 2003]. Section 2.6 also discusses iterative scaling methods for this task.

2.6 Other formulations

This formulation of the maximum-entropy parameter-estimation problem is similar to that of Cover and Thomas [1991] and Jaynes [2003]. It is also similar to the formulation described in Berger et al. [1996] and commonly used in applications of natural language processing, but with three differences. First, this formulation is in terms of unconditional (joint) models, for clarity of presentation, whereas that of Berger et al. [1996] is in terms of conditional models, which are less general but more appropriate for classification tasks. Second, the formulation and results in this thesis apply to either continuous or discrete models, except where otherwise stated. Third, this formulation relaxes the requirement that a set of empirical observations be available as training data for the desired feature targets b_i in (2.1).

Relaxing this third requirement is beneficial, since the notion of prior information is more general than that of frequency data observable in some training set or experiment. Consider, for example, language models crafted for a particular subset of language for an interactive voice response (IVR) system. A corpus of example sentences may not be available at all, but constraints can still be imposed by the requirements for the system—for example, as a set $\mathcal{A}$ of permissible grammars. In this case, no frequency data would be available, but prior information would still be available as constraints of the form $P(\mathcal{A}) = \mathrm{E}f(X) = 1$, where X is a sentence and f is an indicator function $1\{X \text{ parses under some grammar } g \in \mathcal{A}\}$.

A related benefit from relaxing this requirement is increased flexibility for handling the smoothing problem inherent in estimating discrete event probabilities from frequency data. As noted in Section 2.2, pure maximum-entropy inference assumes negligible noise in the data from which the target statistics b_i are derived. In the presence of noise it may be advantageous to adjust the desired b_i to account for the low frequency of certain events in the data as an alternative to (or step towards) a more thorough Bayesian analysis. One proposal is to relax the constraints (2.1) to inequalities, as with the 'fat constraints' of Newman [1977] [see also Khudanpur, 1995]. More recently Chen and Rosenfeld [1999] have tried imposing penalty terms

on large parameter values, inspired by the mature body of literature on smoothing in language modelling. For further discussion on the relationship between model constraints and empirical data in maximum-entropy inference, see [Uffink, 1996].

Several authors, including Geyer and Thompson [1992] and Berger et al. [1996], have presented a different formulation of the problem (2.13), not in terms of maximum-entropy inference, but in terms of maximum-likelihood estimation (MLE) among the class of exponential-family models (2.4). In the notation of this thesis, this problem is to

$$\text{Maximize} \quad l(\theta) = \log \prod_{j=1}^{n} p_\theta(x_j) = \sum_{j=1}^{n} \log p_\theta(x_j) \tag{2.14}$$

where $l(\theta) \equiv l(\theta \mid x) \equiv p_\theta(x)$ is the log likelihood of the model parameters θ given the observed (training) data $x = (x_1, \ldots, x_n)$. Substituting the model density from (2.4) gives

$$l(\theta) = -n \log Z(\theta) + \sum_{j=1}^{n} f(x_j) \cdot \theta. \tag{2.15}$$

It is instructive to compare this to the entropy dual L in (2.11). If the target vector b in (2.11) is set to the mean $n^{-1} \sum_{j=1}^{n} f(x_j)$ of the statistic f over the observed data, then $l = -nL$, so the optimization problems (2.13) and (2.14) are equivalent. This is a consequence of maximum-entropy and maximum-likelihood estimates being convex duals [see e.g. Altun and Smola, 2006]. Paraphrasing a theorem proved by Csiszár [1996]:

> If p is a density of exponential form (2.4) that is feasible for the constraints (2.1), then it is unique, and is also the MLE $\text{lik}\,(\theta)$ over the set of all models with exponential form (2.4).

Several iterative algorithms have been proposed specifically to solve the MLE problem (2.14). The earliest proposals in natural language processing were derivatives of the Generalized Iterative Scaling (GIS) algorithm of Darroch and Ratchliff [1972], such as Improved Iterative Scaling (IIS) [Della Pietra et al., 1997]. These are now considered obsolete, since more recent evidence suggests that general-purpose nonlinear

optimization algorithms can be considerably more efficient for fitting maximum-entropy models [Malouf, 2002] and other exponential-family models, including those arising in Markov random fields [Wallach, 2002] and logistic regression [Minka, 2003].

Although the dual-minimization and MLE problems are equivalent, the former formulation is more compact. It is immediately clear from (2.11), unlike from (2.14), that neither the empirical data $\{x_j\}$ nor their individual statistics $\{f_i(x_j)\}$ need be stored or used for parameter estimation. This observation is critical for efficient parameter estimation for problems like language modelling, which typically use text corpora containing hundreds of millions of words. There is a clear computational benefit to computing the empirical (or target) expectations $\{b_i\}$ of the feature statistics once, and then discarding the empirical data and their statistics, rather than summing over the empirical data at each iteration. Section 3.4.3 discusses the *sufficiency* of the feature statistics for the parameters, which will allow correspondingly large computational benefits when iteratively computing or estimating the entropy dual and its gradient in Chapter 4.

2.7 Benefits and drawbacks

One benefit to adopting the maximum-entropy framework for statistical inference is the ease of using it to synthesize disparate sources of partial evidence. No assumption must be made of statistical independence between the chosen features to make the model mathematically tractable. Instead, the constraints imposed on the features are relatively weak—on expectations—so they can be satisfied simultaneously. This allows a modeller or experimenter to focus more on *what* information to use in the modelling than on *how* to use it, with a convenient separation between the domain-specific task of modelling and the raw computational task of parameter estimation.

A second benefit is that the algorithms for parameter estimation (and the software implementation) are, at least for small sample spaces, independent of the task and of the chosen features or constraints. Again, this is not the case in general

with maximum-likelihood estimation (MLE), for which practitioners use a variety of algorithms, depending on the nature of the likelihood function. The essential difference is that the entropy dual (2.11) is always a convex surface in the parameter space, irrespective of the features; and efficient, general algorithms have already been found for convex optimization [see e.g. Boyd and Vandenberghe, 2004]. The likelihood function, in contrast, is only rarely convex, and the problem of non-convex global optimization is far from solved. The special case of MLE among exponential-family densities is one such rare exception, for which the (negative) likelihood function is convex, as the previous section showed.

The primary drawback of the maximum-entropy framework is that it solves only part of the problem of statistical modelling. Jaynes' papers described no particular means of selecting features, assuming rather that the constraints to be imposed represent solid task-dependent information. In its original form, then, maximum entropy was not a formalism for machine learning, but a principle for scientific inference. The missing link is a principle for using data to derive, or guide the choice of, constraints to impose upon the model, which the maximum-entropy framework takes as its starting point. Inferring or discovering these is a thorny problem at the foundation of statistical inference. Some assumption is still necessary about the relationship between the random process and the frequency data it produces.

As the previous section described, practitioners of maximum entropy sometimes assume the true feature targets to be equal to the mean values observed empirically. Uffink [1996] and others (e.g. Shimony [1985]) have criticized this practice as inadequate, arguing that this introduces an inconsistency between maximum-entropy inference and Bayesian conditionalization. This also reintroduces the problem of over-fitting a model to a finite sample of training data, which may have been a concern motivating the use of maximum entropy in the first place. One can improve efficiency somewhat by smoothing, but a degree of caution is still necessary. Certainly it is highly misleading to claim, as Ratnaparkhi did in [1998, Ch. 1], that

> researchers can use and re-use a single implementation of the maximum

> entropy framework for a wide variety of tasks, and expect it to perform at state-of-the-art accuracies.

One can indeed use and re-use a single implementation for a variety of tasks, but one cannot expect it to yield accurate models unless both the features to constrain and their target values are chosen judiciously. But with constraints that are unrepresentative of the underlying process, or too many constraints, or too few, or target values that have been estimated unreliably, the models will be poor.

This thesis will describe how some of these benefits of maximum-entropy modelling do not carry over to large spaces requiring simulation. Estimators of the entropy dual will not generally share the convexity, or even smoothness, of the true entropy dual. Nor will the separation between the modelling and parameter estimation problems be quite so complete when using simulation, since rare features, in particular, can be problematic to simulate and may need special handling. The remaining chapters will investigate these factors in detail.

2.8 Looking forward

The remainder of this thesis will address the question of how to reduce the computational burden of fitting exponential-family models of form (2.4) on large sample spaces. Here and throughout the thesis, a 'large' sample space is defined as one that is impossible or impractical to integrate over. For such sample spaces, none of the entropy (2.3), entropy dual (2.11), its derivatives (like (2.12)), nor the normalization term (2.5) can be computed exactly. The space of sentences in a natural language is one example of a large discrete space, which Chapter 5 will consider in more detail.

3 Monte Carlo estimation of model features

One barrier to fitting maximum-entropy models on a large sample space is the requirement to integrate over the space to determine how well the constraints are being met. This chapter describes how Monte Carlo methods can be used to estimate the integrals described in the previous chapter when these cannot be determined analytically. The next chapter will then show how the estimators can be used at each iteration of an appropriate iterative algorithm to fit exponential-family models on large discrete or continuous spaces with many thousands of constraints.

This chapter begins by introducing Monte Carlo integration in the context of estimating exponential-family models. It then describes why generating variates from the model itself is unnecessary and, in general, inefficient in comparison with methods of variance reduction such as importance sampling and conditioning.

One contribution of this chapter is an investigation of different importance sampling estimators for the entropy dual and its gradient. We show that the gradient of the *integration estimator* of the entropy dual is equal to the *ratio estimator* of the gradient, which gives us assurance when using estimators of both the dual and its gradient in an exact (determinstic) optimization algorithm.

Another contribution of this chapter is in establishing that, under certain conditions, it is unnecessary to generate variates from the underlying sample space in order to estimate the feature expectations. It is possible, instead, to sample directly in the space induced by the features using an appropriate change of probability measure. This can allow significant computational savings when estimating exponential-form models, especially for hard-to-compute (complex) or hard-to-simulate (rare) features.

3.1 Introduction

Recall that, if a sample space $\mathcal{X}$ is too large to integrate over in practice, neither the entropy dual (2.11) nor its gradient (2.12) can be exactly computed, which poses a problem for estimating maximum-entropy and other exponential-family models. This holds for sample spaces of strings of even a modest length, because the size of such spaces grows exponentially with that length—an example that Chapter 5 discusses in more detail. This also holds for continuous sample spaces, such as the truncated Gaussian models presented in Section 2.8.

Several options are available for evaluating integrals on continuous spaces. Numerical methods of 'quadrature' are practical when the domain of the function being integrated is of low dimensionality. Above a small number of dimensions, these are usually impractical, since their complexity usually scales exponentially with the dimension [Novak and Ritter, 1997]. Some other approaches (especially in many dimensions) include discretizing or quantizing the continuous function by a tractable number of points in each dimension; approximating it by a number of moments; and approximating it by a parametric form that is analytically tractable [see Poland and Shachter, 1993].

The focus of this chapter is on Monte Carlo methods. These offer several benefits. They scale well to high-dimensional spaces, are robust to discontinuities in the integrand, and are general enough not to impose requirements on the integrand being analytically tractable. Unlike the methods mentioned above, they are also applicable to discrete spaces. They also readily yield estimates of the error introduced by simulation, are simple to implement, and are simple to parallelize over multiple processors. They are, therefore, a useful means of harnessing modern computing power to solve complex modelling problems.

3.2 Monte Carlo integration

Consider the generalized moments from Section 2.4, defined as

$$\mu_i \equiv \mathrm{E} f_i(X) = \int_{\mathcal{X}} f_i(x)\,\mathrm{d}P \tag{3.1}$$

for any given model $P \equiv P_\theta$, and assume the parameters θ here to be fixed. Recall that this notation encompasses either discrete or continuous models. The basic Monte Carlo method for estimating the moments $\mu_i = \mathrm{E} f_i(X)$ is to generate IID variates $X_1, \ldots, X_n$ from P and to estimate μ_i by the sample mean

$$\hat{\mu}_i = n^{-1} \sum_{j=1}^{n} f_i(X_j). \tag{3.2}$$

The estimator $\hat{\mu}_i$ is unbiased and, as n becomes increasingly large, converges almost surely to μ_i by Kolmogorov's strong law of large numbers [Robert and Casella, 1998; Andrieu et al., 2003]. The rate of convergence depends on the estimator's variance $\mathrm{Var}\,\hat{\mu}_i$. This is given by

$$\mathrm{Var}\,\hat{\mu}_i = \mathrm{Var}\left(n^{-1} \sum_{j=1}^{n} f_i(X_j) \right) = n^{-1} \mathrm{Var}\, f_i(X_1), \tag{3.3}$$

where the second equality holds because the X_j are, by assumption, IID. Note that the variance of the estimator decreases linearly with the sample size n. In practical terms, an extra digit of accuracy requires 100 times as many replications. This is the one serious drawback of Monte Carlo methods of integration: obtaining high accuracy can be time-consuming.

3.2.1 Reliability

When estimating the expectations μ_i during an iterative parameter estimation algorithm, as we will do in Chapter 4, it can be useful to have confidence intervals for a given estimation method and sample size. One might also wish to impose stricter requirements on the accuracy of the estimates $\hat{\mu}_i$ at later iterations than initially, when only the general step direction is necessary. Here we consider the construction of confidence intervals for the estimates.

As before, we assume that the sample $\{X_j\}$ is IID. Then, by the central limit

theorem (CLT), the distribution of

$$\frac{\hat{\mu}_i - \mu_i}{\sqrt{\operatorname{Var} \hat{\mu}_i}}$$

tends almost surely to the standard normal distribution $\mathrm{N}(0, 1)$. See, for example, Papoulis [1991]. Now suppose we decide we want $\mu_i \in [\hat{\mu}_i - \epsilon, \hat{\mu}_i + \epsilon]$ with $100(1-\alpha)\%$ confidence. The sample size n required to achieve this must then satisfy

$$\varepsilon > z_{\alpha/2} \sqrt{\operatorname{Var} \hat{\mu}_i(X)} = z_{\alpha/2} \sigma_i / \sqrt{n},$$

where $z_{\alpha/2} = \Phi^{-1}(\alpha/2)$ is the $\alpha/2$ quantile of the standard normal distribution, and where we define $\sigma_i^2 \equiv \operatorname{Var} f_i(X_1)$. This holds whenever

$$n > \sigma_i^2 z_{\alpha/2}^2 / \varepsilon^2.$$

We cannot compute σ_i^2, but we can approximate it as the sample variance s_i^2, in which case a more accurate requirement on the sample size is

$$n > s_i^2 t_{\alpha/2, n-1}^2 / \varepsilon^2,$$

where $t_{\alpha/2,n-1}$ is the $\alpha/2$ quantile of Student's t distribution with $n-1$ degrees of freedom. For the large values of n common in practical Monte Carlo simulations, the difference between this and the CLT approximation is likely to be negligible [Hesterberg, 1988].

If we wish to estimate μ_i with a given relative tolerance, such as to within $\pm 5\%$, we require the relative half-width of, say, a 99% confidence interval for μ_i to be less than 0.05, implying

$$n > s_i^2 \left(\frac{2.576}{0.05 \hat{\mu}_i} \right)^2.$$

3.2.2 Generating variates from the model

The basic Monte Carlo estimator (3.2) takes, as its starting point, a sample $X_1, \ldots, X_n$ from the model P. Generating variates from a given distribution is, in general, more

difficult than evaluating it. This section presents two methods to generate such a sample. Both these methods require an *auxiliary* or *instrumental* distribution Q, defined on the same space and with the same support as P, from which it is easier to sample.

Accept-reject algorithm

The accept-reject sampling method, attributed to Von Neumann, uses the idea of filtering candidates generated from an auxiliary distribution to simulate observations from a target distribution. This is useful when sampling directly from the target distribution would be problematic.

Algorithm 1: The rejection sampler

Input: model density p, auxiliary density q and constant $M < \infty$ such that $p(x) \leq Mq(x)$

Output: a single variate X distributed according to p

1 **repeat**

2 Generate $X \sim q$ and $U \sim \mathrm{U}_{(0,1)}$

3 **until** $U \leq p(X)/Mq(X)$

4 **return** X

Knuth [1997] gives a proof that the variates X yielded by Algorithm 1 are distributed according to the density p, provided the bound M is finite. This condition can be satisfied by choosing an auxiliary with heavier tails than the model, in the sense of the ratio q/p being bounded. To generate each variate X, the expected number of draws from the auxiliary distribution is M, so this method is most efficient when the ratio between the model density p and the auxiliary density q is small. One attractive property of this algorithm for simulation from exponential-family densities (as well as posterior Bayes densities) is that knowledge of normalizing constants is unnecessary. This method does, however, require knowledge of the bound M, which can be difficult to find. Underestimating M yields variables from a density other than p (proportional instead to $\min\{p, Mq\}$) [Tierney, 1994]. Overestimating M reduces efficiency by increasing the rejection rate unnecessarily.

Caffo et al. [2001] have shown more recently that a 'practically perfect' sample can be obtained by estimating the bound M by the empirical supremum of the observed values of $p(X)/q(X)$. The modified sample inherits important properties of the original sample, in particular its satisfaction of the strong law of large numbers and central limit theorem, implying that, at least for Monte Carlo purposes, variates generated by the empirical supremum (ESUP) sampler (given as Algorithm 2) may be treated as a random sample from the model P. With this modification rejection sampling becomes very simple to use in practice.

Algorithm 2: The empirical supremum (ESUP) rejection sampler

Input: model density p and auxiliary density q such that the ratio $p(x) \leq q(x)$ is bounded for all $x \in \mathcal{X}$
Output: a single variate $X \sim p$

1 $M \leftarrow 0$
2 **repeat**
3 Generate $X \sim q$ and $U \sim \mathrm{U}_{(0,1)}$
4 $\tilde{M} \leftarrow \max\{\tilde{M}, p(X)/q(X)\}$
5 **until** $U \leq p(X)/\tilde{M}q(X)$
6 **return** X

Metropolis–Hastings algorithm

The Metropolis–Hastings algorithm [Metropolis et al., 1953; Hastings, 1970] is perhaps the most popular of a class of sampling methods known as Markov Chain Monte Carlo (MCMC). The primary application of MCMC methods is to simulate observations from a target distribution that is simple to evaluate but more difficult to generate observations from directly. The idea is to construct a Markov chain whose invariant distribution is (or can be transformed into) the distribution of interest; this allows the generation of (dependent) variates from a distribution that approximates the target distribution increasingly closely. Algorithm 3 describes *independence* sampling, an instance of the Metropolis–Hastings method.

The ratio in line 4 ensures that the independence sampler, like the rejection sampler,

Algorithm 3: The Metropolis–Hastings 'independence' sampler

Input: the desired sample size n, model density p and an auxiliary density q such that $p(x) > 0 \implies q(x) > 0$

Output: variates $X_1, \ldots, X_n \sim p$

1 Generate $X_0 \sim q$
2 **foreach** j in $\{0, \ldots, n-1\}$ **do**
3 Generate $X \sim q$ and $U \sim \mathrm{U}_{(0,1)}$
4 **if** $U < p(X)q(X_j) / \left(p(X_j)q(X)\right)$ **then**
5 $X_{j+1} \leftarrow X$
6 **else**
7 $X_{j+1} \leftarrow X_j$
8 **end**
9 **end**
10 **return** $\{X_1, \ldots, X_n\}$

requires knowledge of the model density p only up to a normalizing constant. This makes it, too, attractive for sampling from exponential-family models, which are generally expensive to normalize.

Notice in line 7 that the independence sampler will re-draw the previous observation if the candidate is rejected. This results in the sample it produces *not* being statistically independent, despite the name. It also ensures that a sample of size n will be produced in only n loops, potentially much faster than the rejection sampler, but with redundancy that increases the variance of estimates relative to an IID sample of the same size.

Rosenfeld et al. [2001] tested the independence sampler and other sampling methods for fitting exponential-form models to natural-language sentences. In their tests both were substantially more efficient than Gibbs sampling for estimating the expectations of various (primarily lexical) features. Chapter 5 will describe some similar tests with language models, comparing the basic Monte Carlo estimates (3.2), derived from the two sampling methods presented here, against importance sampling, discussed next.

3.3 Importance sampling

So far this chapter has described how one way to estimate the features expectations (3.1) is to generate a sample from the model P and use the basic Monte Carlo estimator (3.2). This estimator is suboptimal for two reasons. First, sampling from the model itself can be computationally expensive, whether directly or by resampling variates from another candidate distribution. Second, estimators with lower variances usually exist, implying that the same accuracy can be achieved with a smaller sample size.

The efficiency of a simulation estimator involves two factors: the sample size necessary and the computational effort required to generate it. Sometimes it is possible to increase sampling efficiency by *increasing* the variance; some examples of this are given by Fishman and Kulkarni [1992] and Glynn and Whitt [1992]. But for most purposes reducing the variance is possible without a large increase in the computational cost of generating variates, in which case reducing the variance improves simulation efficiency.

Several methods for reducing variance have been studied and applied widely in practice. Most such methods achieve their variance reduction by harnessing known properties of the quantity being estimated. In Figure 3.1 the area of the unshaded triangle in (b) is known from basic geometry to be 1/2; sampling in that region is therefore unnecessary. This is an example of importance sampling, a powerful and general method of reducing variance that this section (and Chapter 4) will show to be particularly efficient when estimating exponential-family models. Other variance reduction methods include antithetic variables, control variables (sometimes considered a special case of importance sampling), conditioning, and stratification. For readable surveys of these methods, see [L'Ecuyer, 1994] and [Robert and Casella, 1998]; for a description of their relationship to MCMC, see [Andrieu et al., 2003].

The need for variance reduction is particularly acute when one wishes to estimate the probability of a rare event, as is common in performance and reliability analysis. An example given by Heidelberger [1995] is when choosing the buffer size of a switch

in a communications system to make the packet loss probability very small, perhaps 10^{-9}. In this case a sample size of 6.64×10^{11} is required to estimate the packet loss probability to within $\pm 10\%$ at a 99% confidence level. As the probability of interest becomes smaller, the required sample size (and simulation time) grows ever larger. Chapter 5 will demonstrate that rare features can also make maximum-entropy models difficult to estimate.

Example

Figure 3.1 is a simple illustration of variance reduction for Monte Carlo estimation of π. The estimators in both subfigures are based on integrating under the unit circle (whose area is defined to be π) in the first quadrant. The estimator $\hat{\pi}_1$ in (a) is the proportion of sample points (x, y) drawn from independent uniform distributions $\mathrm{U}_{0,1}$ that fall in the circle:

$$\hat{\pi}_1 = 4\left(1\left\{X^2 + Y^2 < 1\right\}\right)$$

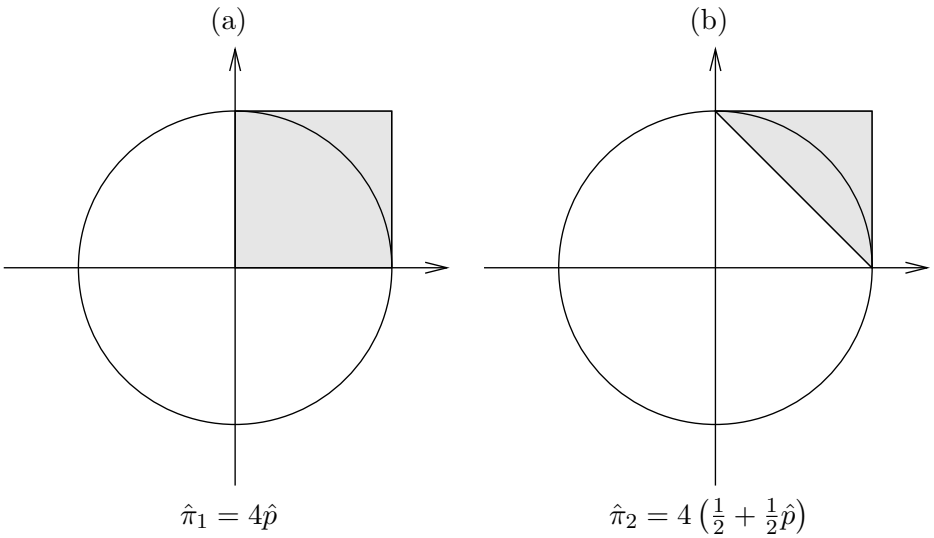


Figure 3.1: Two consistent estimators for π, where $\hat{p}$ is the estimated proportion of the shaded region inside the unit circle given by $\hat{p} = n^{-1}\sum 1\left\{X^2 + Y^2 < 1\right\}$. The variance of estimator $\hat{\pi}_2$ is about 2.752 times smaller than that of $\hat{\pi}_1$.

Its variance is given by

$$\operatorname{Var}(\hat{\pi}_1) = 16 \operatorname{Var}\left(n^{-1} \sum_{j=1}^{n} B_j\right) = 16 n^{-1} \frac{\pi}{4} \left(1 - \frac{\pi}{4}\right) \approx 2.697 n^{-1},$$

where B_j are IID Bernoulli trials with success probability $p = \pi/4$. The estimator $\hat{\pi}_2$ in (b) is the proportion of sample points drawn uniformly from the shaded triangle in (b) that fall in the circle. Note that the unshaded triangle in the first quadrant has a known area of $1/2$. We can use this knowledge to avoid unnecessary sampling effort, focusing it instead on the region of interest near the circle's perimeter. The estimator $\hat{\pi}_2$ is likewise given by

$$\hat{\pi}_2 = 4 \left(1 \left\{X'^2 + Y'^2 < 1\right\}\right), \tag{3.4}$$

but with X' and Y' drawn in a dependent manner, for example as the reflection of the independent uniform variables $X, Y \sim \mathrm{U}_{(0,1)}$ in the line $X + Y = 1$. The variance of $\hat{\pi}_2$ is

$$\operatorname{Var}(\hat{\pi}_2) = 4 \operatorname{Var}\left(n^{-1} \sum_{j=1}^{n} B'_j\right) = 4 n^{-1} \frac{\pi - 2}{2} \left(1 - \frac{\pi - 2}{2}\right) \approx 0.980 n^{-1}, \tag{3.5}$$

where B'_j are IID Bernoulli trials with $p = \pi/2 - 1$. The ratio of the variances is $\hat{\pi}_1/\hat{\pi}_2 \approx 2.752$, so the estimator $\hat{\pi}_2$ requires a sample about 2.752 times smaller for the same mean square error.

3.3.1 The feature expectations and normalization term

Importance sampling is a method for determining something about a distribution by drawing samples from a different distribution and weighting the estimates to compensate for the alteration. One of the earliest applications of importance sampling was in nuclear physics in the 1950s [Kahn and Marshall, 1953]. The technique was developed more fully by Hammersley and Hanscomb [1964] and has since found

applications in many fields of science, engineering and economics where Monte Carlo simulation is necessary. As mentioned before, it is particularly useful when simulating rare events. Some example applications are light transport simulation for image rendering [Veach, 1998], insurance and risk management [Boots and Shahabuddin, 2001], reliability analysis in spacecraft design [Avramidis, 2002], and mobile and radar communications [Srinivasen, 2002].

Like the sampling methods described in the previous section, importance sampling requires an auxiliary distribution Q which is simpler to sample from than the model P. The idea of importance sampling is to use knowledge of the integrand to choose Q to focus more attention on parts of the sample space that have the largest impact on the estimates. The mathematical intuition lies in noting that $\mu_i = \mathrm{E}_p f_i(X)$ can be re-expressed (in the continuous case) as

$$\mathrm{E}_p f_i(X) = \int f_i(x)p(x)\,\mathrm{d}x = \int \frac{f_i(x)p(x)}{q(x)}q(x)\,\mathrm{d}x = \mathrm{E}_q \frac{f_i(X)p(X)}{q(X)}$$

for any probability distribution Q with density q whose support includes the support of P. We generate variates $X_1, \ldots, X_n$ from the auxiliary distribution Q and define the 'classical' or *integration* importance sampling estimator of μ_i as

$$\hat{\mu}_{i,\mathrm{int}} = n^{-1} \sum_{j=1}^{n} W_j f_i(X_j), \tag{3.6}$$

where W_j is the j^{th} Monte Carlo observation of

$$W(X) \equiv \frac{p(X)}{q(X)}. \tag{3.7}$$

The statistic W is the likelihood ratio, or Radon–Nikodym derivative of P with respect to Q, representing weights or correction terms due to the change of measure. Like the basic Monte Carlo estimator (3.2), $\hat{\mu}_{i,\mathrm{int}}$ converges almost surely to μ_i as $n \to \infty$ [Hammersley and Hanscomb, 1964]. Its variance is given by

$$\mathrm{Var}\,\hat{\mu}_{i,\mathrm{int}} = \mathrm{Var}\left(n^{-1} \sum_{j=1}^{n} \frac{f_i(X_j)p(X_j)}{q(X_j)} \right) = n^{-1}\mathrm{Var}\,\frac{f_i(X_1)p(X_1)}{q(X_1)}, \tag{3.8}$$

where the second equality holds because we again assume the $\{X_j\}$ are IID. The goal is for this variance to be lower than that in (3.3) for the basic Monte Carlo estimator. Note that a variance reduction is not guaranteed—the variance can increase instead if the auxiliary distribution Q is chosen poorly, as demonstrated by Hopmans and Kleijnen [1979] and Bratley et al. [1983].

Note that the estimator $\hat{\mu}_{i,\text{int}}$ depends on p, which, as we have noted, can be problematic to normalize for large sample spaces $\mathcal{X}$. If we define

$$\dot{W}_j \equiv W_j Z = \dot{p}(X_j)/q(X_j), \tag{3.9}$$

we can re-express (3.6) as

$$\hat{\mu}_{i,\text{int}} = \frac{1}{nZ} \sum_{j=1}^{n} \dot{W}_j f_i(X_j). \tag{3.10}$$

But note that the denominator still contains the Z term, so computing $\hat{\mu}_{\text{int}}$ in this form is likewise problematic.

The normalization term Z also appears in the expression for the entropy dual (2.11). Although it is possible to fit the model without ever computing the entropy dual—by finding the roots of the gradient—greater efficiency is generally possible in optimization by using more information about the function being optimized. Knowing the normalization term is also necessary for evaluating a fitted model on real data or comparing one fitted model to another.

We can derive an estimator for Z as follows, reasoning similarly to Geyer and Thompson [1992] and Rosenfeld et al. [2001]. For any auxiliary density with $q(x) > 0$ for all $x \in \mathcal{X}$, we can express Z as

$$Z(\theta) = \int_{\mathcal{X}} \frac{\exp(f(X) \cdot \theta)\, q(x)}{q(x)} \, \mathrm{d}\nu(x) = \mathrm{E}_q \left[\frac{\exp(f(X) \cdot \theta)}{q(X)} \right]. \tag{3.11}$$

We can therefore estimate Z by

$$\hat{Z}(\theta) = n^{-1} \sum_j \frac{\exp(f(X_j) \cdot \theta)}{q(X_j)} = n^{-1} \sum_j \dot{W}_j \tag{3.12}$$

where $\{X_j\}$ are IID variates with density q. As before, as the sample size n grows, the estimator $\hat{Z}$ tends to Z almost surely and its variance tends to zero.

Returning now to the feature expectations $\{\mu_i\}$, suppose we substitute the estimator $\hat{Z}$ for Z in (3.10). Then we get a different estimator,

$$\hat{\mu}_{i,\text{ratio}} = \frac{1}{n\hat{Z}} \sum_{j=1}^{n} \dot{W}_j f_i(X_j), \tag{3.13}$$

the *ratio* estimator. Hesterberg [1988] and others [e.g. Robert and Casella, 1998] have shown that the ratio estimator is more robust and more effective than the integration estimator in a variety of contexts. In general the ratio estimator is biased, since in general $\mathrm{E}(Y/Z) \neq \mathrm{E}(Y)/\mathrm{E}(Z)$. The bias can most clearly be seen when $n = 1$: then $\hat{\mu}_{i,\text{ratio}} = f_i(X_1)$, whose expectation is $\mathrm{E}_q f(X)$ rather than $\mathrm{E}_p f(X)$. A second-order approximation is possible, based on an Edgeworth expansion[1] for $\mathrm{E}(Y/Z)$ about $\mathrm{E}Y/\mathrm{E}Z$, but for large sample sizes n the bias is likely to be negligible compared to the standard error [Hesterberg, 1988].

3.3.2 Choosing the auxiliary distribution

A natural goal when choosing the auxiliary distribution Q is to minimize the variance of the resulting estimators. In the case when $f_i(x) \geq 0$ for all $x \in \mathcal{X}$, it is straightforward to see that the density

$$q_i^*(x) = \frac{f_i(x)}{\mu} \tag{3.14}$$

yields an estimator $\hat{\mu}_{i,\text{int}}$ whose variance (3.8) is actually zero. The minimum-variance estimator in the more general case of real-valued f_i was given by Rubinstein [1981]

[1] A power series expansion of the characteristic function. See Hall [1992] for an introduction.

as

$$q_i^*(x) \propto p(x)|f_i(x)| \tag{3.15}$$

This choice of density also minimizes the variance of the ratio estimator $\hat{\mu}_{i,\text{ratio}}$ [Hesterberg, 1988], although if f_i takes both positive and negative values the minimum is no longer zero. But in both cases there is a catch—the denominator in (3.14) and the constant of proportionality in (3.15) involve the integrals $\int_{\mathcal{X}} f_i(x)\,\mathrm{d}P$ and $\int_{\mathcal{X}} |f_i(x)|\,\mathrm{d}P$, respectively; the first is equal to the quantity μ_i of interest, and the second is likely just as problematic to compute. But this result is still more than a theoretical curiosity; it can guide our intuition when choosing an auxiliary sampling distribution. Note, in particular, that sampling from the model itself will not minimize variance, and should not be the goal. Note also that the auxiliary distribution should not necessarily be chosen to emphasize regions where the statistic f is extreme, but rather to make fp/q more constant than f. For a more thorough discussion, see Hesterberg [1988].

3.3.3 The derivative of the entropy dual

Substituting the estimator $\hat{Z}$ from (3.12) for its true value Z in (2.11) yields the following estimator for the entropy dual:

$$\hat{L}(\theta) = \log \hat{Z}(\theta) - b \cdot \theta \tag{3.16}$$

This allows one to fit an exponential-family model approximately, using a general-purpose optimization algorithm (see Section 2.5), even if the exact entropy dual cannot be computed.

Now consider trying to reduce the number of iterations by using gradient information in conjunction with this entropy estimator. If we cannot compute the expectations $\mathrm{E}_p f_i(X)$ exactly, but try instead to estimate these too by sampling, the following question arises:

> Should we use the gradient of an estimator of the dual function (2.11),

or an estimator of its gradient (2.12)?

The choice might impact the convergence of an optimization algorithm that uses estimates of both the objective and gradient functions.

The following result shows that, if we estimate Z by $\hat{Z}$ as above and estimate μ by the *ratio* importance sampling estimate $\hat{\mu}_{\text{ratio}}$ with the same auxiliary distribution, these alternatives are in fact the same.

Lemma 3.1. *Suppose $\dot{p}(x)$ is an unnormalized exponential-family density of form $p_0(x)\exp(f(x)\cdot\theta)$ with respect to the measure ν on the sample space $\mathcal{X}$, and suppose q is a normalized density function on the same space, where $q(x) > 0$ whenever $\dot{p}(x) > 0$. Then, if X_j are distributed according to q,*

$$\hat{Z}(\theta) \equiv n^{-1}\sum_{j=1}^{n}\dot{p}(X_j)/q(X_j)$$

is an unbiased estimator for the normalization factor $\int_{\mathcal{X}}\dot{p}(x)\,\mathrm{d}\nu$. Furthermore,

$$\frac{\partial}{\partial\theta_i}\log\hat{Z}(\theta) = \hat{\mu}_{i,\text{ratio}}, \tag{3.17}$$

where $\hat{\mu}_{i,\text{ratio}}$ is the ratio importance sampling estimator of the expectation $\mu_i = \mathrm{E}_{p_\theta} f_i(X)$, defined as in (3.13).

Proof. The first claim follows from re-expressing Z as in (3.11). To see that the second claim holds, apply the chain rule for partial differentiation twice to give

$$\frac{\partial\log\hat{Z}}{\partial\theta_i} = \frac{\partial\hat{Z}}{\partial\theta_i}\cdot\frac{1}{\hat{Z}}$$

and

$$\frac{\partial\hat{Z}}{\partial\theta_i} = n^{-1}\sum_j f_i(X_j)\frac{\dot{p}(X_j)}{q(X_j)} = n^{-1}\sum_j f_i(X_j)\dot{W}_j.$$

□

This property is a natural analogue of the exact case in Equation (2.6). We

have already seen that the ratio estimator can be used to estimate statistics of an unnormalized density, whereas other estimators like the integration estimator cannot. This result allows us to define the *gradient estimator* $\hat{G} \equiv \nabla \hat{L}$ of the entropy dual unambiguously as the vector with components

$$\hat{G}_i \equiv \frac{\partial \hat{L}}{\partial \theta_i} \equiv \hat{\mu}_{i,\text{ratio}} - b_i. \tag{3.18}$$

3.3.4 Reliability

Hesterberg [1988] and Geweke [1989] show that under mild regularity conditions both the integration and ratio estimators (as well as the 'regression estimator', not discussed here) are asymptotically consistent and asymptotically normally distributed, meaning that, as $n \to \infty$, they tend to μ_i almost surely and that $\sqrt{n}\,(\hat{\mu}_i - \mu_i)$ tends in distribution to $\mathrm{N}\,(0, \mathrm{Var}\,(\hat{\mu}_i))$. This allows us to construct approximate $(1-\alpha)$ confidence intervals as before, as

$$\hat{\mu}_i \pm z_{\alpha/2}\hat{\sigma}_i, \tag{3.19}$$

where $z_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$ and $\hat{\sigma}_i$ is the estimated standard error. Define Y_j as the product $f_i(X_j)W(X_j)$ and define $\bar{Y}$ as its sample mean $n^{-1}\sum_1^n Y_j$; then the estimators

$$\hat{\sigma}^2_{i,\text{int}} = \frac{1}{n(n-1)} \sum_{j=1}^{n} (Y_j - \bar{Y})^2 \tag{3.20}$$

$$\hat{\sigma}^2_{i,\text{ratio}} = \frac{1}{n(n-1)} \sum_{j=1}^{n} (Y_j - W_j\hat{\mu}_{i,\text{ratio}})^2 \tag{3.21}$$

are consistent and unbiased estimates of the asymptotic variance of the integration and ratio estimators $\hat{\mu}_{i,\text{int}}$ and $\hat{\mu}_{i,\text{ratio}}$. The corresponding confidence intervals have limiting $(1-\alpha)$ coverage under the same conditions as for asymptotic normality of the $\hat{\mu}_i$ estimators.

Despite this, they are not always accurate enough with practical sample sizes. Hes-

terberg [1988] describes four improvements, two of which I mention here: bootstrap confidence intervals, such as the accelerated BC_a intervals [DiCiccio and Efron, 1996] and bootstrap-t intervals [Choquet et al., 1999]; and intervals of the form

$$\hat{\mu}_i \pm z_{\alpha/2}\hat{\sigma}_i/\bar{W}. \tag{3.22}$$

These latter intervals are chosen to address a common problem with the earlier intervals (3.19) for importance sampling— that they are often optimistic when the average of the weights $\bar{w}$ is small.

3.3.5 Example

Consider the problem posed by Berger et al. [1996] of estimating a simple language model for machine translation of the English preposition 'in' into French. Suppose that in a corpus of parallel texts, such as the proceedings of the European or Canadian parliaments, we observe that the translation is always one of the five French prepositions in the set

$$\mathcal{X} = \{\textit{dans}, \textit{en}, \textit{à}, \textit{au cours de}, \textit{pendant}\},$$

and that the following relations hold:

$$\begin{aligned} \Pr(X \in \{\textit{dans}, \textit{en}\}) &= .3 \\ \Pr(X \in \{\textit{dans}, \textit{à}\}) &= .5. \end{aligned} \tag{3.23}$$

This prior information can be encoded as three constraints $\text{E}f_i(X) = b_i$, for $i = 1, 2, 3$, where f_i are the indicator functions

$$\begin{aligned} f_1(x) &= 1\{x \in \mathcal{X}\} \\ f_2(x) &= 1\{x \in \{\textit{dans}, \textit{en}\}\} \\ f_3(x) &= 1\{x \in \{\textit{dans}, \textit{à}\}\}, \end{aligned} \tag{3.24}$$

and where the target expectations are $b = (1, .3, .5)'$.

The feature statistics corresponding to the five words in the sample space are summarized in Table 3.1. The exact maximum-entropy solution, found using the parameter estimation framework reviewed in Section 2.5, is given in Table 3.2. The software implementation for this and all subsequent parameter estimation tasks is described in Appendix A.

Table 3.1: Feature statistics for the machine translation example of Berger et al. [1996]

x	$f(x)$
dans	$(1, 1, 1)'$
en	$(1, 1, 0)'$
à	$(1, 0, 1)'$
pendant	$(1, 0, 0)'$
au cours de	$(1, 0, 0)'$

Table 3.2: Fitted model for the translation example

x	$P(x)$
dans	.1859
en	.1141
à	.3141
au cours de	.1929
pendant	.1929

Suppose we now attempt to fit the same model by using importance sampling to estimate the expectations of the features (3.24). Suppose we define a uniform auxiliary distribution Q as

x	*dans*	*en*	*à*	*au cours de*	*pendant*
$Q(x)$	.2	.2	.2	.2	.2

and draw random words $x_1, \ldots, x_n$ from Q. Table 3.3 shows the entropy dual function $L(\theta)$ after fitting using the true feature expectations μ and the estimated entropy

Table 3.3: Minimized dual entropy function $L_{\min}$ with feature expectations computed exactly and with importance sampling with different sample sizes n from the auxiliary distribution Q

exact	1.5591
simulated, $n = 10^2$	1.5674 ± 0.0073
simulated, $n = 10^3$	1.5600 ± 0.0009
simulated, $n = 10^4$	1.5592 ± 0.0001

dual $\hat{L}(\theta)$ after sample path optimization (described in Chapter 4) using the ratio estimators $\hat{\mu}_{\text{ratio}}$ given by (3.13). Table 3.4 shows fitted parameters for a range of sample sizes.

The samples used here may appear large for such a small problem, but recall that the variance of Monte Carlo estimates decreases linearly with the sample size (Section 3.2), and that the same asymptotic relation also holds with importance sampling (Section 3.3.1). We have seen that importance sampling does, however, give us the flexibility to choose an auxiliary distribution Q to reduce the estimation error by a constant factor for a given sample size. The next section will show that this model can also be estimated by sampling directly in the space of features.

Table 3.4: Example fitted model parameters for the translation example (each for one random sample)

	θ'
exact	$(0, -0.525, 0.488)$
simulated, $n = 10^2$	$(0, -0.656, 0.585)$
simulated, $n = 10^3$	$(0, -0.504, 0.438)$
simulated, $n = 10^4$	$(0, -0.512, 0.505)$

3.4 Sampling in feature space

The rarest (hardest to simulate) features of a model are the most difficult to estimate accurately with the basic Monte Carlo method. Importance sampling weakens this requirement so that the only features difficult to estimate are those that are difficult to simulate under *any* model on the same space. The basic problem, however, might still remain.

An example is that when modelling sentences, as in Chapter 5, a high-level feature indicating 'grammaticality' or 'meaningfulness', could be problematic to simulate unless the auxiliary distribution used for generating random sentence-like strings were considerably more sophisticated than the best language models in use today. Once a set of linguistic features has been selected, the computational problem is that of simulating sentences that are linguistically balanced with respect to this set.

This section considers whether, for a given set of features f, the procedure outlined so far in this chapter for estimating $\mathrm{E}f(X)$—by generating variates x_j on a probability space $\mathcal{X}$, then computing $f(x_j)$—is necessary, or whether it may be possible instead to sample directly in the space of features $\mathcal{F} = \{f(x) : x \in \mathcal{X}\} \subset \mathbb{R}^m$. The tool we use for this is a generalization of importance sampling and the conditional Monte Carlo method.

Manipulating samples in the feature space directly would be beneficial in several ways. First, it would avoid the possibly expensive computation of the feature statistics of all random variates in the sample. The variates may also be highly redundant (consider images, for example), with the transformation to the feature space yielding significant compression. Second, it would allow more control over the auxiliary distribution used, allowing direct generation of features f that are relevant, rather than trying to guess which regions of the original sample space $\mathcal{X}$ are likely to yield relevant features under f.

This section derives consistent estimators for the moments $\mathrm{E}f(X)$ from a sample drawn directly in feature space. It then presents some examples of how to use these estimators in practice. This extends the scope for estimating exponential-family

models to those with features that are desirable to constrain but problematic to estimate by sampling in the original space.

3.4.1 Conditional Monte Carlo

The *conditional Monte Carlo* method was first presented by Trotter and Tukey [1956] as a method of variance reduction involving observations from the $\mathrm{N}(0,1)$ distribution. Hammersley [1956] described a more general setting for it later that year. It has since been applied for variance reduction and other purposes, including gradient estimation in perturbation analysis [Fu and Hu, 1997], whose applications include financial derivative pricing and inventory control. Lavenberg and Welch [1979] describe applications to queueing networks. It is a remarkably powerful technique, as I hope this section will show.

Suppose X and T are two random variables, not necessarily on the same space, and define $g(T)$ as the conditional expectation

$$g(T) \equiv \mathrm{E}_X \left[f(X) \mid T \right]. \tag{3.25}$$

Then the property

$$\mathrm{E}_T \left[\mathrm{E}_X (X \mid T) \right] = \mathrm{E}X \tag{3.26}$$

of conditional expectations implies that

$$\mu = \mathrm{E}g(T). \tag{3.27}$$

If T is a discrete variable taking values in the finite or countably infinite set $\mathcal{T}$, then another derivation of (3.27) is by the law of total probability, which states

$$\mu = \sum_{t \in \mathcal{T}} g(t) \Pr(T(X) = t) = \mathrm{E}g(T). \tag{3.28}$$

The *conditional Monte Carlo* procedure is to sample $X_1, \ldots, X_n$ from the conditional distribution of X given $T = t$, estimating the conditional expectation $g(t)$ as

$$\hat{g}(t) = n^{-1} \sum_{j=1}^{n} f(X_j). \tag{3.29}$$

A consistent estimator of μ is then

$$\hat{\mu}_{\text{cmc}} = \int_{\mathcal{T}} \hat{g}(t)\, \mathrm{d}T. \tag{3.30}$$

Note that the variates T_j need not be drawn on the same space as X; instead variates T_j can be drawn on a different space, provided the function g can be evaluated for each observation T_j. For more background on the conditional Monte Carlo method, see Hammersley [1956]; Robert and Casella [1998]; Huseby et al. [2004].

Hammersley [1956] gave a more general statement of this, noting the flexibility inherent in this method for choosing the sample spaces, random variable T, and a mapping $x = \xi(t)$ so that the expectation $\mathrm{E}_p f(X)$ is given by

$$\mathrm{E}_p f(X) = \int_{\mathcal{T}} f(\xi(t)) w(t) q(t) dt = \mathrm{E}\left[f\left(\xi(T)\right) w(T)\right]$$

for some manageable weight function w, so it is easy to sample from T, and so the variance of $f\left(\xi(T)\right) w(T)$ is as small as possible.

3.4.2 Sufficiency of the feature statistic

This section describes the property of *sufficiency* of f for θ. The usual definition of sufficiency is as follows:

> The statistic $T(X)$ is *sufficient* for θ if, for each T, the conditional distribution of X given T does not depend on θ.

I give two proofs here that the feature statistic $T = f(X)$ is sufficient for θ when X has the exponential-family form (2.4). The first is a direct one, for the discrete case only, whose notation and ideas we will use later in the section.

Lemma 3.2. *Take $T = f(X)$ and define $\mathcal{X}_t$ as the set $\{x \in \mathcal{X} : f(x) = t\}$. If X is*

a discrete random variable with PMF p_θ *defined as in (2.4), the conditional* PMF *of* X *given* $T = t$ *is given by*

$$p_\theta(x \mid t) = \begin{cases} 1/\left|\mathcal{X}_t\right| & \text{if } x \in \mathcal{X}_t \\ 0 & \text{if } x \notin \mathcal{X}_t \end{cases} \tag{3.31}$$

for all t *such that* $\Pr(X \in \mathcal{X}_t) > 0$*, where* $|\mathcal{X}_t|$ *is the cardinality of* $\mathcal{X}_t$*. Furthermore, since* $p_\theta(x \mid t)$ *is independent of* θ*,* f *is sufficient for* θ*.*

Proof. We have

$$p_\theta(x \mid t) \equiv \frac{\Pr(X = x, f(X) = t)}{\Pr(f(X) = t)} = \frac{\Pr(X = x, f(X) = t)}{\Pr(X \in \mathcal{X}_t)}, \tag{3.32}$$

provided the denominator is not zero. If $\mathcal{X}_t$ is a finite or countably infinite set, the law of total probability implies that the denominator of (3.32) is

$$\Pr(X \in \mathcal{X}_t) = \sum_{x \in \mathcal{X}_t} \frac{1}{Z(\theta)} \exp(t \cdot \theta) = |\mathcal{X}_t| \frac{1}{Z(\theta)} \exp(t \cdot \theta).$$

The numerator of (3.32) is clearly given by

$$\begin{cases} \frac{1}{Z(\theta)} \exp(t \cdot \theta) & \text{if } f(x) = t \\ 0 & \text{if } f(x) \neq t. \end{cases}$$

The (non-zero) expression $\frac{1}{Z(\theta)} \exp(t \cdot \theta)$ cancels from both numerator and denominator, yielding (3.31).

The sufficiency of the f for θ follows from the independence of $p_\theta(x \mid t)$ from θ in (3.32).

□

The second proof follows immediately from the factorization theorem [e.g. Lindsey, 1996, Ch. 3], since the model density p_θ trivially has a factorization of the form

$$p_\theta(x) = a(f(x), \theta)b(x)$$

with $b(x) \equiv 1$ and $a(f(x), \theta) \equiv \frac{1}{Z(\theta)} \exp(f(x) \cdot \theta)$.

An intuitive statement of this is that knowing more about a sample x than that $f(x) = t$ is of no additional help in making any inference about θ; the original sample x can be discarded once the statistics $f(x)$ have been computed. This has a computational benefit in an iterative context, which we harness in Section 4.4.1. This also opens up an opportunity for estimating the feature expectations μ without ever sampling in the original sample space.

3.4.3 Feature-space sampling theorem

For the conditional Monte Carlo estimator $\hat{\mu}_{\text{cmc}}$ to be useful for simulation, we want the conditional distribution $X \mid T$ to have the following properties:

1. Either

   a) the function $g(T) = \mathrm{E}\left[f(X) \mid T\right]$ and the probability density of T should be easily computable, to use (3.27) directly; or

   b) sampling from the conditional distribution $X \mid T$ should be efficient, to use (3.29).

2. The variance $\operatorname{Var} g(T)$ should be substantially smaller than $\operatorname{Var} f(X)$.

   Note that it is guaranteed to be no larger by the following basic property of conditional expectations [Feller, 1968; Bucklew, 2005]:

   $$\operatorname{Var} f(X) = \mathrm{E}_T\left[\operatorname{Var}_X(f(X) \mid T)\right] + \operatorname{Var}_T\left[\mathrm{E}_X(f(X) \mid T)\right],$$

   where the first term on the right side is clearly non-negative and the second

term is $\operatorname{Var} g(T)$. So conditioning in this way will never increase the standard error.

Consider now taking $T(X) = f(X)$. Then the first part of condition (1a) is clearly satisfied, since $g(T) = \mathrm{E}[T \mid T] = T$. The second part is described in the discrete case by the following lemma.

Lemma 3.3. *Suppose X is a discrete random variable on the sample space $\mathcal{X}$ whose probability mass function can be expressed as $p(x) = a(f(x), \theta)$ for some function a. Define $T = f(X)$; then T has probability mass function*

$$\pi(t) = |\mathcal{X}_t|\, a(t, \theta), \tag{3.33}$$

where $|\mathcal{X}_t|$ is the cardinality of the set $\mathcal{X}_t \equiv \{x \in \mathcal{X} : f(x) = t\}$.

Proof. Define $\mathcal{X}_t$ as the set $\{x \in \mathcal{X} : f(x) = t\}$. Supposing X and T are discrete random variables, the PMF $\pi_\theta(t)$ of T is given by

$$\pi_\theta(t) = \Pr(X \in \mathcal{X}_t) = \sum_{x \in \mathcal{X}_t} p_\theta(x) = \sum_{x \in \mathcal{X}_t} a(t, \theta) = |\mathcal{X}_t|\, a(t, \theta). \tag{3.34}$$

□

The machinery of conditional Monte Carlo is not a strict alternative to importance sampling. They can be combined, as noted in the early paper of Hammersley [1956] and more recent works such as Bucklew [2005]. The next theorem shows how they can be combined for discrete PMFs of form $a(f(x), \theta)$ such as exponential-family densities, making it possible to adopt an auxiliary PMF $q(t)$ on the space of features $\mathcal{T} \subset \mathbb{R}^m$ that satisfies condition (2) above.

Theorem 3.1 (Feature-space sampling theorem)**.** *Suppose X is a discrete random variable with PMF $p(x) = a(f(x), \theta)$ for some function a, where f is a vector of m statistics defined on the same sample space $\mathcal{X}$ as X. Let $q(t)$ be an auxiliary PMF*

defined on a sample space $\mathcal{T} \subset \mathbb{R}^m$*, such that* $q(t) > 0$ *whenever* $|\mathcal{X}_t| t \neq 0$*. Then*

$$\mathrm{E}_q \left[\frac{|\mathcal{X}_T|\, a(t,\theta)}{q(T)} T \right] = \mathrm{E}_p f(X). \tag{3.35}$$

Proof. First define $T = f(X)$ and note that T is discrete, since X is discrete. The property (3.26) of conditional expectations implies

$$\mathrm{E} f(X) = \mathrm{E} T = \sum_{t \in \mathcal{T}} t \pi_\theta(t).$$

Since $q(t) > 0$ whenever $t\pi_\theta(t) \neq 0$, we have

$$\mu = \sum_{t \in \mathcal{T}} t \frac{\pi_\theta(t)}{q(t)} q(t) = \mathrm{E}_q \left[T \frac{\pi_\theta(T)}{q(T)} \right],$$

which, by Lemma 3.3, is the statement of (3.35).

□

This theorem describes a combination of a transformation of random variables and importance sampling. The method can be applied more generally in the continuous case, as Hammersley [1956] described—but has been done so surprisingly seldom. The restrictions this theorem imposes on the random variable X being discrete, with a PMF of form $a(f(x), \theta)$, allow a particularly convenient expression of the estimator on the left side of (3.35) in terms of simple 'correction factors' $|\mathcal{X}_T|$. Note that these are constant in the parameters θ, a consequence of f being a sufficient statistic for θ.

An interpretation of this theorem is as follows. Consider defining a random variable T on the space $\mathcal{T} \subset \mathbb{R}^m$ of features, defining an auxiliary PMF on this feature space for importance sampling, and abusing the notation $p_\theta(t) = \exp(t \cdot \theta)/Z(\theta)$. Then draw variates $T_1, \ldots, T_n$ according to q and consider the estimator

$$\hat{\mu}^* \equiv n^{-1} \sum_{j=1}^{n} \frac{p(T_j)}{q(T_j)} T_j \approx \mathrm{E}_q \left[\frac{p(T)}{q(T)} T \right]$$

This looks plausible as an importance sampling estimator for μ, but it is inconsistent whenever f is not a bijection. Unless there is a bijection $t \leftrightarrow x$ the abuse of notation is not justified. Warning!

The theorem shows that the correction factor $|\mathcal{X}_t|$ fills this gap. If f is not injective (if it has points $x_1 \neq x_2$ with $t = f(x_1) = f(x_2)$), then $|\mathcal{X}_t| > 1$ for these t. If f is not surjective (if there are $t \in \mathcal{T}$ but $\notin \mathcal{F}$), then $|\mathcal{X}_t| = 0$ for these t.

The conditions on q are very mild. To apply the theorem to exponential-family models, choose an auxiliary PMF q, draw a sample $\{T_1, \ldots, T_n\}$ of random vectors from q, and use the new ratio estimator

$$\hat{\mu} \equiv \frac{1}{n\hat{Z}} \sum_{j=1}^{n} V_j T_j, \tag{3.36}$$

where

$$V_j \equiv \frac{|\mathcal{X}_{T_j}| \exp(T_j \cdot \theta)}{q(T_j)}, \tag{3.37}$$

along with the corresponding estimator of the dual in (3.16), but with Z defined as

$$\hat{Z} \equiv n^{-1} \sum_{j=1}^{n} V_j \equiv \bar{V}. \tag{3.38}$$

instead of as (3.12). Notice that the unconditional importance sampling estimators presented in Section 3.3.1 are subsumed under this new formulation as a special case, where $|\mathcal{X}_{T_j}| = 1$ and $V_j = \dot{W}_j$ for all j.

The only tricky part is determining the correction factors $|\mathcal{X}_{T_j}|$. The next three examples show this in action.

3.4.4 Examples

Example 1 Consider again the machine translation example from Section 3.3.5. Define two new distributions directly on the space of features $\mathcal{F} = \{f(x) : x \in \mathcal{X}\} \subset \mathcal{T} = \{0, 1\}^3$ with probability mass functions $q_1, q_2 \colon \mathcal{F} \to \mathbb{R}$ given by:

t	$q_1(t)$	$q_2(t)$
$(1,1,1)'$	.25	.2
$(1,1,0)'$	.25	.2
$(1,0,1)'$	.25	.2
$(1,0,0)'$	.25	.4

Consider using importance sampling to estimate $\mu = \mathrm{E}_{p_\theta} f(X)$ with either q_1 and q_2 as an auxiliary PMF by defining the estimator

$$\dot{W}_j^* \equiv \frac{\exp(T_j \cdot \theta)}{q(T_j)} \tag{3.39}$$

analogously to (3.9), and defining the estimators $\hat{Z}$ and $\hat{\mu}$ as in (3.12) and (3.13). This does *not* work: these estimators are inconsistent for both $q = q_1$ and $q = q_2$. Minimizing the estimated entropy dual $\hat{L}(\theta)$ given in (3.16) based on these estimators does not minimize the true entropy dual, and the resulting model does not satisfy the desired constraints. An example run with a sample size $n = 10^4$ is:

	estimated	actual	desired
$p(dans) + p(en)$	0.301	0.224	0.3
$p(dans) + p(\grave{a})$	0.499	0.370	0.5

	estimated	actual	desired
minimized entropy dual	1.302	1.603	1.559

Interestingly, both q_1 and q_2 yield the *same* inconsistent estimators and the same fitted model. Both actually correspond to the same constraints as in (3.23), except on a smaller sample space with only four points, such as $\mathcal{X} = \{$*dans*, *en*, *à*, *au cours de*$\}$ or $\mathcal{X} = \{$*dans*, *en*, *à*, *pendant*$\}$. To fit the desired model we need to account for the presence of two sample points ($x_4 =$ *pendant* and $x_5 =$ *au cours de*) whose feature vectors $f(x_4) = f(x_5) = (1,0,0)'$ are the same.

Example 2 This example shows how to modify the above procedure in the light of Theorem 3.1. To apply the theorem we need:

(a) to define an auxiliary PMF $q(t)$ on some space $\mathcal{T} \subseteq \mathbb{R}^m$, a superset of $\mathcal{F} = \{f(x) : x \in \mathcal{X}\}$, such that $q(t) > 0$ whenever $|\mathcal{X}_t|t \neq 0$. Both the auxiliary distributions from the previous example satisfy this condition.

(b) to determine the sizes of the equivalence classes $\mathcal{X}_t$ for each t in the support of q. For this example we have, for all $t \in \mathcal{F}$,

$$|\mathcal{X}_t| = \begin{cases} 2 & \text{if } t = (1,0,0)' \\ 1 & \text{if } t \neq (1,0,0)' \end{cases}$$

and $|\mathcal{X}_t| = 0$ for all $t \notin \mathcal{F}$.

Now, to apply Theorem 3.1, we draw $T_1, \ldots, T_n$ from q, then estimate μ as in (3.36). Minimizing the estimated entropy dual $\hat{L}(\theta)$ now provides a good fit to the desired model (here again with $n = 10^4$):

	estimated	actual	desired
$p(dans) + p(en)$	0.301	0.301	0.3
$p(dans) + p(\grave{a})$	0.497	0.498	0.5

	estimated	actual	desired
minimized entropy dual	1.560	1.559	1.559

Chapter 4 will present the iterative algorithms used for this.

Example 3 This example shows that any errors in reporting the multiplicity factors $|\mathcal{X}_t|$ for variates generated from q are sufficient to invalidate the estimates, even if such violation occurs with very low probability. Consider the following two auxiliary PMFs:

t	$q_3(t)$	$q_4(t)$
$(0,1,5)'$	.1	.0001
$(1,1,1)'$	.15	.2499
$(1,1,0)'$	.25	.25
$(1,0,1)'$	.25	.25
$(1,0,0)'$	.25	.25

Note that the feature vector $t = (0,1,5)'$ has no corresponding words $x \in \mathcal{X}$—that is, has $t \neq f(x)\ \forall x \in \mathcal{X}$, and so $|\mathcal{X}_t| = 0$. Since both q_3 and q_4 assign a strictly positive probability to this feature ($t \in \sup q_3$ and $t \in \sup q_4$), it is necessary to account for this explicitly in the estimator (3.37). Suppose we were to ignore this, using instead the values:

t	$\lvert\mathcal{X}_t\rvert^*$
$(0,1,5)'$	1
$(1,1,1)'$	1
$(1,1,0)'$	1
$(1,0,1)'$	1
$(1,0,0)'$	2

Then drawing a sample $T_1, \ldots, T_n$ according to either q_3 or q_4 and minimizing the estimated entropy dual (3.16) using the estimators (3.37), (3.12) and (3.13) does not fit the correct model. The following are based on a sample of size $n = 10^5$. With q_3, one example run gives:

	estimated	actual	desired
$p(dans) + p(en)$	0.328	0.330	0.3
$p(dans) + p(\grave{a})$	0.245	0.246	0.5

	estimated	actual	desired
minimized entropy dual	1.724	1.719	1.559

and with q_4:

	estimated	actual	desired
$p(dans) + p(en)$	0.321	0.324	0.3
$p(dans) + p(\grave{a})$	0.228	0.230	0.5

	estimated	actual	desired
minimized entropy dual	1.753	1.742	1.559

If any variates t are generated under an auxiliary model q without correctly accounting for the cardinality terms $|\mathcal{X}_t|$ corresponding to the transformation, the actual probability mass $q(t)$ of these is immaterial. Even if such variates occur with low probability, the estimators will in general be inconsistent unless the correct multiplicities $|\mathcal{X}_t|$ are used in the estimates, as in (3.37).

We can verify that with the correct multiplicity factors

t	$\lvert\mathcal{X}_t\rvert$
$(0,0,1)'$	0
$(1,1,1)'$	1
$(1,1,0)'$	1
$(1,0,1)'$	1
$(1,0,0)'$	2

the fitted models using the estimators (3.37), (3.12) and (3.13) are reasonable. For one example run with a sample of size $n = 10^5$ from q_3, we obtain:

	estimated	actual	desired
$p(dans) + p(en)$	0.300	0.300	0.3
$p(dans) + p(\grave{a})$	0.503	0.501	0.5

	estimated	actual	desired
minimized entropy dual	1.557	1.559	1.559

Using q_4 gives similar results.

3.5 Conclusions

This chapter has discussed how integrals that arise when fitting exponential-family models can be estimated using Monte Carlo methods when evaluating them analytically or numerically is infeasible.

Section 3.2 reviewed two classes of algorithm (the rejection and Metropolis–Hastings independence samplers) for sampling from exponential-family models such as maximum-entropy models; both of these are useful because they can be applied without normalizing the model densities.

Section 3.3 reviewed the alternative of importance sampling, which introduces flexibility for reducing estimation error by defining a new measure on the same probability space that is more efficient or more beneficial to sample from. It constructed 'integration' and 'ratio' importance sampling estimators for the entropy dual and its gradient. Using the ratio estimator for the gradient is beneficial for two reasons: it, too, makes normalization unnecessary; and it equals the derivative of the 'natural' estimator of the dual function.

A natural extension of the idea of importance sampling is to transform the underlying probability space (rather than just the measure). Section 3.4 gave a theory for when and how it is possible to transform the estimators to the space induced by the feature statistics of discrete exponential-family models. The essential requirement is to determine constant factors representing the number of points in the original space that map to each point in a sample in the new space. If this knowledge is available, features that are 'rare' or onerous to simulate in the original space need not pose a problem for estimation, since they can be simulated as often as desired in the new space.

4 Fitting large maximum-entropy models

Chapter 2 described the deterministic problem of estimating exponential-form models on sample spaces over which integration is practical. In general this is analytically intractible and requires an iterative algorithm. This chapter will describe the corresponding stochastic problem that arises when the exact objective and gradient values are uncomputable and must be estimated at each iteration with Monte Carlo methods.

This chapter first reviews the literature in stochastic optimization and relates it to the problem of fitting large exponential-form models. It then derives some algorithms for streamlining the function and gradient evaluations for maximum computational efficiency while avoiding numerical overflow. The resulting implementation is fast, memory-efficient, robust, and general enough to support a variety of applications.

Section 4.2 describes *stochastic approximation*, a class of algorithms for stochastic optimization that steps iteratively towards the optimum in the direction of the negative gradient, as do deterministic gradient-descent algorithms, but formally accommodates simulation error in the evaluations of the objective function and its moments.[1] Such algorithms soften the convergence guarantee to almost sure convergence (with probability 1) as the number of iterations grows.

Section 4.3 describes another, quite different, approach known as *sample path optimization*, which approximates the stochastic optimization problem by a deterministic problem involving a fixed sample. This approach allows the large

[1] Note that this is quite different to accommodating noise or error in empirical data, as in maximum *a posteriori* (MAP) estimation.

and mature body of literature on deterministic optimization to be applied directly, but offers no guarantee of convergence with a single trial.

Section 4.4 presents a computational framework for estimating the objective function and its gradient in terms of matrix–vector operations in log space. This supports efficient implementations that harness the parallel instruction sets of modern superscalar CPUs without difficulties arising from numerical overflow.

Section 4.5 describes the relation between the two approaches to stochastic optimization discussed in Sections 4.2 and 4.3, and compares their applicability for estimating exponential-form models.

4.1 Introduction

Stochastic optimization, also called *simulation optimization* and *stochastic programming*, can be interpreted broadly as the search for a system with optimal expected performance, or for optimal decisions based on uncertain data. It has a correspondingly broad range of applications, including agriculture, financial portfolio management, manufacturing, computer networks, and microprocessor design [see the references in Carson and Maria, 1997; Birge and Louveaux, 1997; Marti, 2005]. The general problem can be framed as

$$\text{Minimize} \quad L(\theta) = \mathrm{E}_p \hat{L}(\theta), \tag{4.1}$$

where the parameters θ can take values in some set Θ, and where $\hat{L}$ is some consistent estimator of the objective function L. The essential feature of stochastic optimization problems is that exact computation of the objective function L and its moments is impossible, impractical, or otherwise undesirable. The estimation problem this thesis examines can be formulated naturally as the stochastic optimization problem

$$\text{Minimize} \quad L(\theta) = \mathrm{E}_p\big[\log \hat{Z}(\theta) - b \cdot \theta\big], \tag{4.2}$$

analogously to the deterministic optimization problem (2.13).

The stochastic problem poses two additional obstacles. One is that the optimization algorithm will be impaired in its efficiency and potentially its convergence properties by a degree of error in the estimates of the objective function and its moments. The other is that these estimates are often costly to compute. When each iteration uses Monte Carlo estimation, one obstacle can be traded for the other with the usual n^2 relation—a factor n decrease in relative error of the estimates requiring a factor n^2 increase in simulation time. In some applications, such as protein structure prediction [Herges et al., 2003], aircraft wing design [Siclari et al., 1996], and oil exploration [Bangerth et al., 2004], each iteration can take several CPU hours on a parallel processor. Stochastic optimization problems can, unsurprisingly, be extremely demanding of computational resources, since they can require many iterations for convergence upon a solution—and sometimes only an approximate solution is feasible.

4.2 Stochastic approximation

Stochastic approximation is motivated by the desire to optimize the performance of some system by adjusting some parameters, where the output of the system is corrupted by 'noise' or uncertainty, and where the nature of the dependence of the system upon the parameters is unknown. Stochastic approximation describes a class of algorithms that step iteratively towards the optimum of a function whose value or gradient is known only with an error term. The algorithms converge on the true solution in a probabilistic sense as the number of iterations increases.

Stochastic approximation algorithms were first put on a solid mathematical footing by Robbins and Monro [1951], then generalized to the multidimensional case by Blum [1954], and later refined by others, including Ruppert [1991]; Polyak and Juditsky [1992]. The goal of Robbins and Monro's initial formulation was to find the root of a noisy function; it was then a natural step to apply this to (local) optimization, in which the root of the gradient is the optimum solution. It has been applied in various fields, such as protein structure prediction [Herges et al., 2003], model predictive

control [Baltcheva et al., 2003], and resource allocation [Hill, 2005], and the theory is simple and tractable for numerical analysis [Chen, 2002]. Its principles have also been used to accelerate some deterministic optimization problems in machine learning, by using individual experimental data points, rather than a large batch, to derive step directions. This method is known as stochastic gradient descent [Spall, 2003].

Algorithm 4 is a multidimensional generalization of the basic Robbins–Monro algorithm. It is based on the presentation of Andradóttir [1998b], but simplified by the assumption that, as with maximum-entropy models, the gradient $G(\theta)$ is defined for all $\theta \in \mathbb{R}^m$.

Algorithm 4: Stochastic approximation

Input: a stochastic function $\hat{G}(\theta)$ with expectation $\mathrm{E}\hat{G}(\theta) = G(\theta)$; an initial solution estimate θ_0; a sequence $\{a_k\}$ of positive real numbers satisfying $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$; a suitable stopping criterion

Output: a solution θ to $G(\theta) = 0$

```
foreach k in ℕ = {0, 1, ...} do
    Compute Ĝ(θ_k)
    if Ĝ(θ_k) satisfies the stopping criterion then
        return θ_k
    θ_{k+1} ← θ_k − a_k Ĝ(θ_k)
end
```

The central result of stochastic approximation is that, with a decreasing step-size a_k with the properties $\sum a_k = \infty$ and $\sum a_k^2 < \infty$, the algorithm will converge with probability 1 to the root of $G(\theta) = 0$ [Wasan, 1969]. An intuitive explanation of this requirement is that the step size must decrease to zero, but slowly enough that a solution can first be reached from anywhere in the feasible region. A simple example of such a sequence is $a_k \propto 1/k$.

Many researchers have considered how to generalize the above algorithm to when the gradient is not available, using finite differences of the function value [Kushner and Yin, 1997, Ch. 11] and other approximations, such as simultaneous perturbation analysis [Spall, 1998, 2004]. When fitting exponential-family models, however, ratio

importance sampling estimates of the gradient are readily available, as Section 3.3.3 showed. Indeed, estimating the gradient of the entropy dual is simpler and likely more stable than estimating the dual directly. The basic method of stochastic approximation is therefore suitable to apply directly to the problem (4.2).

The primary advantages of stochastic approximation methods are their simplicity and robustness. Their simplicity makes them convenient to implement for a variety of applications and tractable for mathematical analysis of their convergence properties. They also have the potential for greater efficiency over methods that require accurate estimates at each iteration, due to the inverse quadratic relationship between sample size and standard error in Monte Carlo estimators.

One disadvantage of stochastic approximation (in the basic form of Algorithm 4) is its instability when the objective function grows faster than linearly in the parameters θ. This is described by Andradóttir [1996], along with a modification for scaling the parameters to ensure convergence even if the function is unbounded. Another variation on the basic stochastic approximation algorithm, due to Ruppert [1982, 1991] and Polyak and Juditsky [1992], involves simply averaging the iterates. Although simple, this has been shown to be effective in improving the stability for functions that increase up to quadratically in the parameters. The Ruppert–Polyak modification has also been shown to accelerate convergence, yielding an asymptotic mean square error identical to what would be obtained using the true Hessian in a stochastic Newton-like algorithm. Numerical studies have verified the benefit of iterate averaging [Kushner and Yin, 1997], although its property of asymptotic optimality does not imply optimality in practical problems with finite sample sizes [Maryak, 1997].

A second disadvantage is that the rate of convergence of stochastic approximation algorithms is highly sensitive to the chosen sequence of step sizes $\{a_k\}$. The fastest possible asymptotic rate of convergence for stochastic approximation algorithms, assuming a constant time per step and an optimal choice of step sizes $\{a_k\}$, would be $k^{-1/2}$, the same rate as any Monte Carlo estimator. Achieving this would be an impressive feat, considering that stochastic approximation algorithms combine estimation with optimization over the parameter space. Simple choices of step sizes

such as $a_k \propto 1/k$ usually yield substantially slower convergence than this. Several researchers have proposed and analyzed variations in which the step size decreases only when there is reason to believe the algorithm has reached a neighborhood of the original solution, such as when the gradient often changes in sign [Delyon and Juditsky, 1991], and this is a field of active research [e.g. Plakhov and Cruz, 2005].

More recent research has attempted to apply successful principles from deterministic optimization to the stochastic setting. Wardi [1990] and Yan and Mukai [1993] discuss choosing step sizes that satisfy the Armijo conditions, which helps to guarantee the convergence of many deterministic optimization algorithms. Shapiro and Wardi [1996a,b] discuss the use of line searches to select step sizes. Other papers draw inspiration from the well-known Newton–Raphson method in deterministic optimization to choose better step directions by approximating the Hessian matrix of the objective function [Ruppert, 1985; Spall, 2000]. Schraudolph and Graepel [2003] discuss generalizations to the stochastic setting of the method of conjugate gradients, which is beneficial for large problems by not requiring storage of an approximate Hessian matrix.

4.3 Sample path optimization

Sample path optimization, also called the *sample average approximation* method, is a different method to solve the stochastic approximation problem (4.2). The idea is to draw a sample $\{x_1, \ldots, x_n\}$ and to estimate the optimum of the original stochastic function L in (4.2) by optimizing the corresponding deterministic function $\hat{L}(\theta)$. This approach is simple and natural, and allows direct use of efficient algorithms from the mature body of research on deterministic optimization.

The sample path optimization problem for exponential-family densities is therefore

$$\text{Minimize } \hat{L}(\theta) = \log \hat{Z}(\theta) - b \cdot \theta, \tag{4.3}$$

where $\hat{L}$ and $\hat{Z}$ denote estimates obtained from a fixed sample $\{x_1, \ldots, x_n\}$ and b

is again the vector of target values for the moment constraints (see Section 2.4). If this sample is IID, and if we denote the optimal solution to the true problem (4.2) as θ^* and the optimal solution to the sample path optimization problem (4.3) as $\hat{\theta}^*$, then the strong law of large numbers implies that $\hat{\theta}^*$ converges with probability 1 to θ^* as the sample size $n \to \infty$. If the function L is smooth near the optimum, the rate of convergence is the same $O(n^{-1/2})$ rate as the underlying Monte Carlo estimator $\hat{L}$. For a more detailed convergence analysis, see Gürkan et al. [1994]; for a more thorough description, see Robinson [1996]; Andradóttir [1998a,b]; Kleywegt and Shapiro [2001].

Sample path optimization has seen particular success with discrete problems. Kleywegt et al. [2001] describe theoretical results that the probability of sample path optimization yielding the true optimum of a discrete stochastic program grows exponentially fast with the sample size n. Shapiro [2001] contrasts this convergence rate with the slower rate in the smooth case. Kleywegt et al. [2001] also apply sample path optimization to a resource allocation problem called the *static stochastic knapsack problem*; Greenwald et al. [2005] describes applications to two larger discrete problems, stochastic bidding and scheduling problems, whose exact solution in both cases is NP-hard. Verweij et al. [2003] describe applications to stochastic routing problems.

Geyer and Thompson [1992] describe the application of sample path optimization to the estimation of exponential-family models. Although couched in the language of maximum likelihood estimation, their paper contains the essence of the method outlined in this section. Rosenfeld et al. [2001] later applied the method to estimating exponential-form language models, perhaps arriving at the idea independently. Section 5.1 will review this paper of Rosenfeld et al. in more detail.

The primary advantage of this method is its efficiency. One reason for this, already mentioned, is the direct applicability of efficient deterministic optimization algorithms. Another is the economy of sampling effort; in its simplest form, the method requires only a single sample drawn up-front for all iterations. Section 4.4 will also show that the dual and its gradient can be estimated highly efficiently from a fixed sample,

with computational time dominated by the cost of two matrix multiplications. We would therefore expect sample path optimization to be efficient overall. Chapter 5 will show that this is indeed true—where feasible.

The reason the method is not always feasible is that the sample average $\hat{L}$ derived from a finite sample is less smooth than the true function L. This is the primary disadvantage of the sample path method for continuous optimization problems. Convergence guarantees for deterministic optimization algorithms rely on certain assumptions about the smoothness of the objective function and its gradient, and approximating these with a finite sample will in general violate these assumptions. This may lead to divergence when the sample size is too small in relation to the variance, or convergence to a local optimum of the sample average approximation that is far from the true optimum. For discrete problems where the probability of finding an optimum is positive, several replications of this procedure behave like multiple Bernoulli trials, and the probability of finding the true optimum approaches 1 as the number of replications increases [Kleywegt et al., 2001]. For continuous problems, such as parameter estimation for exponential-family densities (whether the underlying sample space be continuous or discrete), we have no such guarantee.

Using multiple trials of sample path optimization has, as already mentioned, been successful in discrete stochastic problems. One simple heuristic to apply this variant to continuous problems, suggested by Andradóttir [1998b], is to average the fitted parameters resulting from each trial. Repeating sample path optimization multiple times with independent samples can do more than improve accuracy; it can also help with drawing inferences about the variability of the estimates. Rosenfeld et al. [2001] used this approach to estimate the variance of various estimates when fitting whole-sentence maximum-entropy language models, by generating 10 samples, deriving 10 estimates of the expectations, then testing the variance of the sample means. Intra-sample variability of features can also potentially provide information about the variability of the estimates of the feature expectations μ, but this is somewhat complicated by the estimates not being independent, even if the features are. Note that the bootstrap is an inappropriate tool to apply here; its computational expense

is unnecessary when the simple option of drawing another sample is available, with its added benefit in being usable to improve the estimates.

4.4 Algorithms

So far this thesis has described several tools for estimating large exponential-family models. This section describes the numerical algorithms behind my implementation of this in software, which is now part of SciPy, an Open Source toolkit for scientific and engineering computing (see Appendix A).

Several computational difficulties present themselves when estimating such models. One is that realistic models may have many features, perhaps millions, as with the language models of Rosenfeld [1996]. Since each of these features needs to be evaluated at each observation in a random sample, and since large random samples may be necessary to obtain estimates with a satisfactory standard error, sparse representations of these features are critical, as are efficient means of evaluating the function and gradient at each iteration.

A second computational difficulty is that values of the partition function Z on large sample spaces can easily overflow a standard 64-bit floating-point data type. This introduces difficulties in computing the estimates (3.16) and (3.18) of the entropy dual and its gradient. This section addresses these two issues in turn.

4.4.1 Matrix formulation

We describe first a formulation, originally presented by Malouf [2002], of the objective function and its gradient in terms of matrix–vector operations for the case of a discrete sample space $\mathcal{X} = \{x_1, \ldots, x_k\}$. The primary advantage of such a formulation is that it allows the use of matrix–vector primitives like the BLAS[2] routines used by many scientific software libraries. These routines are often optimized explicitly for the corresponding vector instructions of modern CPUs, either by hardware vendors or

[2]Basic Linear Algebra Subprograms: `http://www.netlib.org/blas`

automatically, by programs such as ATLAS[3] [Whaley et al., 2001; Whaley and Petitet, 2005]. This can provide a performance increase of one or two orders of magnitude over unoptimized compiled code.

This section sets Euclidean vectors and matrices in boldface for clarity

The first insight is that, instead of storing all points $\{x_1, \ldots, x_k\}$ in the sample space $\mathcal{X}$, we need only store an $(m \times k)$ matrix of their features $\boldsymbol{F} = \big(f_i(x_j)\big)$. The following expressions are straightforward consequences of the definitions given in Section 2.4. For a maximum-entropy model parameterized by $\boldsymbol{\theta}$, the unnormalized model probability $\dot{p}_\theta(x_j)$ of the event x_j is given by the j^{th} component of

$$\dot{\boldsymbol{p}} = \exp\left(\boldsymbol{F}'\boldsymbol{\theta}\right), \tag{4.4}$$

where $\boldsymbol{F}'\boldsymbol{\theta}$ is a transposed matrix–vector product and the antilog is element-wise. For the more general case of a model of minimum relative entropy from some non-uniform prior $p_0(x)$, we can express the elementwise logarithm of $\dot{\boldsymbol{p}}_{\boldsymbol{0}}$ neatly as

$$\log \dot{\boldsymbol{p}} = \boldsymbol{F}'\boldsymbol{\theta} + \log \boldsymbol{p}_{\boldsymbol{0}}. \tag{4.5}$$

In this notation, the deterministic optimization problem from Section 2.4 is

$$\text{Minimize} \quad L(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta}) - \boldsymbol{\theta}'\boldsymbol{b}, \tag{4.6}$$

where $\boldsymbol{\theta}'\boldsymbol{b}$ denotes the inner product of the column vectors $\boldsymbol{\theta}$ and $\boldsymbol{b}$; where

$$Z = \mathbf{1}'\dot{\boldsymbol{p}}; \tag{4.7}$$

and where $\mathbf{1}'$ is a row vector of n ones. Recall from Chapter 2 that this is an unconstrained convex minimization problem. The gradient $\boldsymbol{G}(\boldsymbol{\theta})$ of $L(\boldsymbol{\theta})$ is given by

$$\boldsymbol{G} = \boldsymbol{\mu} - \boldsymbol{b}; \tag{4.8}$$

$$\boldsymbol{\mu} = Z^{-1}\boldsymbol{F}\dot{\boldsymbol{p}}. \tag{4.9}$$

[3] Automatically Tuned Linear Algebra Software: `http://math-atlas.sourceforge.net`

Note that the matrix $\boldsymbol{F}$ may be highly sparse, particularly if many of the features f_i are indicator functions. Note also that the columns of $\boldsymbol{F}$ correspond to all events $\{x_1, \ldots, x_k\}$ in the sample space $\mathcal{X}$, not just those in some set $\mathcal{T}$ of training data. Recall from Section 2.6 that this is a design choice, with the advantage that the size of a training data set $\mathcal{T}$ is immaterial once the target moments $\boldsymbol{b}$ have been calculated. This formulation is, however, clearly unsuitable for sample spaces $\mathcal{X}$ that are either continuous or discrete but practically innumerable.

I now describe an adaptation of this formulation to the stochastic optimization problem (4.2) based on the Monte Carlo estimators from Chapter 3. Recall that the sample path optimization problem (4.3) is

$$\text{Minimize} \quad \hat{L}(\boldsymbol{\theta}) = \log \hat{Z}(\boldsymbol{\theta}) - \boldsymbol{\theta}'\boldsymbol{b}, \tag{4.10}$$

where $\hat{L}$ and $\hat{Z}$ are estimates based on a fixed sample $x_1, \ldots, x_n$ from some known distribution Q. Equation (4.10) also describes the stochastic approximation problem, with the difference that $\hat{L}$ and $\hat{Z}$ are instead *estimators* based on random samples that are re-drawn each iteration.

The term $\hat{Z}$ from (3.38) can be expressed in matrix notation as

$$\hat{Z} = n^{-1}\mathbf{1}'\boldsymbol{v} \tag{4.11}$$

in terms of the unnormalized importance sampling weights $\boldsymbol{v} = (v_j)$ given in (3.37). A general expression for $\boldsymbol{v}$ for maximum-entropy models, subsuming both cases of the auxiliary distribution being defined on $\mathcal{X}$ or on $\mathcal{F}$ (explained in Section 3.4.3), is

$$\log \boldsymbol{v} = \boldsymbol{F}'\boldsymbol{\theta} + \log \boldsymbol{c} - \log \boldsymbol{q}, \tag{4.12}$$

where the logarithm is element-wise, $\boldsymbol{q}$ is a column vector of the (normalized) densities of the random sample $\{x_j\}$ under the auxiliary distribution Q, and $\boldsymbol{F} = (f_i(x_j))$ is an $(m \times n)$ matrix of features of the random sample. Note in particular that here $\{x_j\}_1^n$ is a *random* sample, whereas the deterministic case above involved the entire

discrete sample space $\mathcal{X}$. We define $\boldsymbol{c}$ as the vector of multiplicity terms

$$c_j \equiv |\mathcal{X}_{T_j}|,$$

defined as in Lemma 3.2. Recall that this is just the 1-vector when the auxiliary distribution is defined on the same sample space $\mathcal{X}$ as the model, so the $\log \boldsymbol{c}$ term in (4.12) vanishes, as we would expect, giving a neat analogy with (4.5). In the important case that T_j is not reachable from any point $x \in \mathcal{X}$ by the mapping f, the components c_j will be zero, and we define $\log c_j$ and $\log v_j$ to be $-\infty$, and v_j to be zero. Any such variates T_j then make no contribution to the estimators.

When minimizing relative entropy $D(P\|P_0)$ from some non-uniform prior P_0, the expression (4.12) becomes

$$\log \boldsymbol{v} = \boldsymbol{F}'\boldsymbol{\theta} + \log \boldsymbol{p_0} - \log \boldsymbol{q}, \tag{4.13}$$

where $\boldsymbol{p_0}$ is a vector of n terms $(p_0(x_j))$ describing the probability density or mass of the random variates $\{x_j\}$ under the prior.

Finally, note that the ratio estimator of the feature expectations in (3.36) can be expressed as

$$\hat{\boldsymbol{\mu}} = \left(n\hat{Z}\right)^{-1} \boldsymbol{F}\boldsymbol{v}. \tag{4.14}$$

Therefore the gradient $\hat{G}(\boldsymbol{\theta}) = \nabla \hat{L}(\boldsymbol{\theta})$, defined in Section 3.3.3, is given by

$$\hat{\boldsymbol{G}} = \hat{\boldsymbol{\mu}} - \boldsymbol{b}. \tag{4.15}$$

Note the strikingly similarity between the estimators (4.5–4.9) and their exact counterparts (4.10–4.15). These matrix–vector expressions support efficient evaluation of the objective function and its gradient in software. The computation of the objective function estimator $\hat{L}$ is dominated by the cost of the transposed matrix–vector product $\boldsymbol{F}'\boldsymbol{\theta}$ (in either (4.12) or (4.13)). The additional cost for computing the gradient estimator $\hat{\boldsymbol{G}}$ is dominated by another matrix–vector product, $\boldsymbol{F}\boldsymbol{v}$, in (4.14).

Both involve the same matrix $\boldsymbol{F}$ of sufficient statistics derived from the sample.

4.4.2 Numerical overflow

The previous section framed the dual and gradient estimates $\hat{L}$ and $\hat{G}$ in terms of the estimated partition function $\hat{Z}$. For large sample spaces, $Z = \int_{\mathcal{X}} \exp(f(x) \cdot \theta)\, \mathrm{d}\nu(x)$ can be an extraordinarily large value. We may potentially have many features active (non-zero) for any given x. If the sample space $\mathcal{X}$ is discrete, the value of Z will overflow a standard 64-bit floating point type if the inner product $f(x) \cdot \theta$ is greater than approximately 720 for any one of the events $x \in \mathcal{X}$. This therefore demands some thought about how to represent Z or $\hat{Z}$ in a computer.

One idea for a workaround is to attempt to scale the feature statistics f_i down by a large factor (of similar magnitude to $\hat{Z}$). This would not help, since the parameters θ_i would merely be scaled up by the same factor during the fitting process, negating the benefit when computing the inner product terms $f(x) \cdot \theta$. A second idea is to use larger-precision data types, perhaps 128 bits or more. This would work for some problems, but for many problems, including those considered in Chapter 5 the values of Z are truly huge, and this would be insufficient.

This section describes a solution based on the following trick, which is sometimes used in the literature on Turbo Coding [e.g., Tan and Stüber, 2000].

Lemma 4.1. *For all* $a_1, a_2 \in \mathbb{R}$,

$$\log(e^{a_1} + e^{a_2}) = \max\{a_1, a_2\} + \log\left(1 + e^{-|a_1 - a_2|}\right) \tag{4.16}$$

The proof is straightforward, treating the two cases $a_1 \geq a_2$ and $a_1 < a_2$ separately. This generalizes to n terms as follows. Define $a_{\max} \equiv \max_j\{a_j\}$. Then:

Lemma 4.2. *For all* $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$,

$$\operatorname{logsumexp}(\boldsymbol{a}) \equiv \log\left(\sum_{j=1}^{n} e^{a_j}\right) = a_{\max} + \log\left(\sum_{j=1}^{n} e^{a_j - a_{\max}}\right). \tag{4.17}$$

Proof. Since $e^{a_{\max}} > 0$, we have

$$\log\left(\sum_{j=1}^{n} e^{a_j}\right) = \log\left(e^{a_{\max}} \sum_{j=1}^{n} e^{a_j - a_{\max}}\right) = a_{\max} + \log\left(\sum_{j=1}^{n} e^{a_j - a_{\max}}\right).$$

□

Chiang and Boyd [2004] prove that logsumexp is a convex function. This verifies that the maximum-entropy parameter-estimation problem in (4.6) is indeed a convex optimization problem.

This general expression for the logsumexp function is perhaps not as widely known as it should be. It is certainly not widely described in the literature, although it has probably been re-discovered independently several times in different contexts when numerical stability is in question. It has, however, been described in at least three publications: [Schofield, 2004] with exponential-family language models; [Mann, 2006] with hidden Markov models; and [Cohn, 2006] with conditional random fields.

This lemma allows one to compute $\log Z$ and $\log \hat{Z}$, even if Z and $\hat{Z}$ are too large to represent. At each iteration the entropy dual $\hat{L}$ is estimated as in (4.10), but now, instead of first computing $\hat{Z}$, we use (4.11–4.13) to compute the term $\log \hat{Z}(\theta)$ as

$$\log \hat{Z} = \text{logsumexp}(\boldsymbol{F}'\boldsymbol{\theta} + \log \boldsymbol{c} - \log \boldsymbol{q}) - \log n \tag{4.18}$$

when maximizing entropy, or

$$\log \hat{Z} = \text{logsumexp}(\boldsymbol{F}'\boldsymbol{\theta} + \log \boldsymbol{p_0} - \log \boldsymbol{q}) - \log n \tag{4.19}$$

when minimizing relative entropy.

Avoiding numerical overflow when computing the gradient vector $\hat{\boldsymbol{G}} = \boldsymbol{\mu} - \boldsymbol{b}$ requires more thought. The term $\hat{Z}$ in the expression

$$\hat{\boldsymbol{\mu}} = \left(n\hat{Z}\right)^{-1} \boldsymbol{F}\boldsymbol{v}$$

appears by itself, rather than as a logarithm. Re-expressing $\hat{\boldsymbol{\mu}}$ by similarly taking the (element-wise) logarithm of the term $\boldsymbol{F}\boldsymbol{v}$ would lead to trouble with negative and zero features in the sample feature matrix, requiring special handling for these elements. Such handling could involve using complex logarithms or gathering all positive, negative, and zero elements in $\boldsymbol{F}$ separately and invoking an analogue of Lemma 4.1 for the logarithm of a difference of exponentials $\log(e^{a_1} - e^{a_2})$. Both of these solutions would incur more computational overhead.

We can, however, side-step the need to take $\log \boldsymbol{F}$ by shuffling the order of operations. We do this by taking the factor of $1/\hat{Z}$ inside the antilog, which yields

$$\hat{\boldsymbol{\mu}} = n^{-1}\boldsymbol{F}\boldsymbol{u}, \tag{4.20}$$

where $\boldsymbol{u}$ is the vector with elements

$$u_j = \exp(\log v_j - \log \hat{Z}). \tag{4.21}$$

Since $\log \hat{Z}$ is at least as large as any element $\log v_j$, all elements of $\log \boldsymbol{u}$ will be negative, and exponentiating will not result in overflow.

With some architectures and compiler configurations, exponentiating a large negative value may instead raise a numerical underflow exception. The contribution of any such values to the sum in (4.17) is negligibly small, and it is safe to set them explicitly to zero. Most compilers (and all compilers that adhere to the C99 standard) allow a process to specify this behaviour globally.

The software described in the Appendix performs all relevant computations in logarithmic space using these expressions for the estimators (and their exact counterparts). This effectively solves the problem of numerical overflow. The computational overhead it introduces is small, since the time is still dominated by the matrix–vector products $\boldsymbol{F}'\boldsymbol{\theta}$ and $\boldsymbol{F}\boldsymbol{u}$.

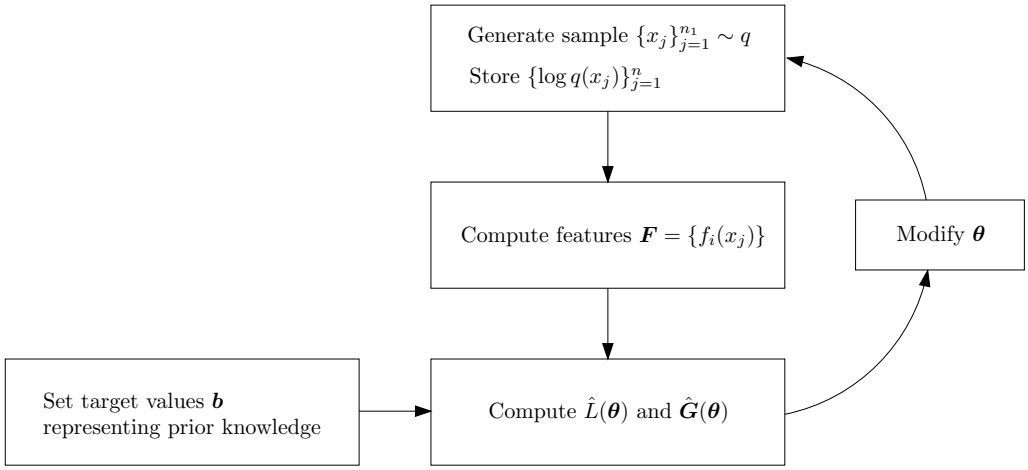
4.5 Discussion

This section discusses the relation between stochastic approximation (SA) and sample path optimization (SPO), and how both methods can use the matrix–vector formulation of the previous section. It also briefly analyzes the computational complexity of obtaining estimates of the entropy dual and its gradient with this formulation.

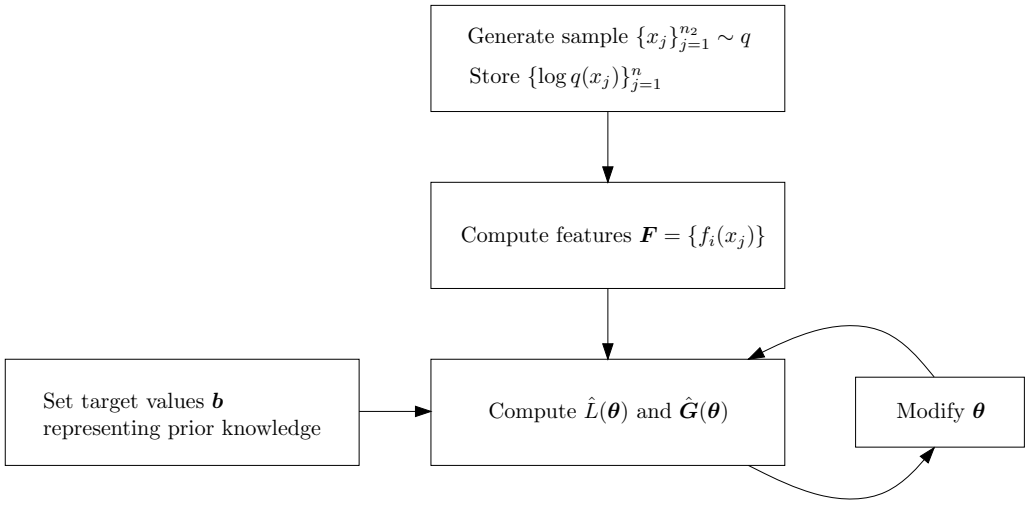
One difference between the two methods is that the original SA formulation of Robbins and Monro (presented as Algorithm 4) uses a single sample point at each iteration, whereas applications of SPO typically use larger samples. But when applying SA one can also clump together more than one sample at each iteration to estimate the gradient more accurately. This makes it possible to use the matrix–vector expressions from the previous section to increase computational efficiency. Recall from Section 3.3.1 that using a sample size larger than 1 is in fact necessary to ensure the consistency of the ratio estimator $\hat{\mu}_{\text{ratio}}$.

A second difference between the two approaches is that of the precise method used to define the step direction and step size at each iteration. This difference is less significant, since the step size and direction are somewhat flexible with both methods.

The most fundamental difference between the two approaches is illustrated in Figure 4.1: whether (a) a new sample is drawn afresh after each one or several iterations, or (b) a single sample is drawn at the outset and used for all iterations. The SPO method, by recycling a single sample over all iterations, is clearly more economical with sampling effort than the SA method. The computational complexity of the SPO method increases linearly with the sample size until memory is exhausted, after which it increases more quickly (although still linearly) until virtual memory (or secondary storage capacity) is exhausted. After this the SPO method can no longer be applied in the same form. This can impose an upper bound on the sample size, after which one can only increase the accuracy of the estimates by reducing the sampling variance with methods such as those presented in Chapter 3. Chapter 5 will give examples of where too little is known about the chosen feature statistics



(a) Stochastic approximation



(b) Sample path optimization

Figure 4.1: High-level overview of (a) stochastic approximation versus (b) sample path optimization

to apply these variance reduction methods effectively, causing the SPO method to diverge.

In the case that the SPO method diverges, the SA method is an alternative. Most SA algorithms guarantee convergence as the number of iterations increases, and drawing a new sample each iteration removes the restriction on sample size imposed by memory size, since estimates can be constructed sequentially.

The computation time of fitting deterministic maximum-entropy models has been noted to be dominated by gradient evaluations [e.g. Ratnaparkhi, 1998, Sec. 2.6.1]. This is likely to be especially true with stochastic approximation, when simulation is necessary. Recall that, when using the framework of the previous section, the computation required at each iteration is dominated by two matrix multiplications. Computing the gradient requires both of these multiplications; computing the dual only requires one of these and the relatively cheap vector dot-product $\boldsymbol{\theta}'\boldsymbol{b}$. This implies that optimization algorithms using this framework would not see a noticeable efficiency benefit from not computing and using the dual value. Malouf [2002] observed this for the small-sample case; we see here that the large-sample case with either importance sampling or conditional Monte Carlo estimators has a similar structure.

4.6 Conclusions

The previous two chapters have presented a suite of tools for estimating large exponential-family models efficiently. Chapter 3 considered two Monte Carlo methods for estimating the entropy dual function and its gradient, whereas Chapter 4 has considered classes of optimization algorithms that can use these Monte Carlo estimates iteratively. Coupling these algorithms with importance sampling yields near-identical expressions with or without a transformation of probability space, and whether maximizing entropy or minimizing relative entropy. Chapter 4 presented these as a unified framework in terms of matrix–vector operations in log space, which allows software implementations to benefit from the parallel instruction sets of modern CPUs.

5 Example: whole-sentence language models

Maximum-entropy methods have had a profound impact on natural language processing (NLP) since the community first adopted them in the early 1990s. They provide an important tool to achieve what NLP researchers have realized to be necessary for at least two decades, but have had difficulty achieving—a fusion between probabilistic models inferred from data and formal representations of linguistic knowledge.

A language model is a crucial component of many NLP systems—for automatic speech recognition, predictive text entry, machine translation, optical character recognition, and grammar checking. An accurate language model should impart a speech recognizer with a strong prior preference for fluent, grammatical utterances; reduce the keystroke count required to use a predictive typing aid; produce more fluent translations and more accurate conversion of scanned documents into text; and give better grammatical advice.

This chapter applies the ideas developed in previous chapters to reduce the computational burden for fitting *whole-sentence models* of exponential form, which were first explored in an innovative paper by Rosenfeld et al. [2001]. This chapter shows how, when the methods from the previous chapters are applied in combination, the computational burden for this task can be reduced by several orders of magnitude versus the methods Rosenfeld et al. [2001] described. Large whole-sentence models can then be fitted accurately within a few minutes on a modern computer.

5.1 Introduction

Speech recognition The need for language models arose in the 1970s in the context of speech recognition. Researchers, beginning with Bahl et al. [1982] at IBM,

posed the problem of speech recognition as follows:

> Given an acoustic signal x, find the word string w for which $p(w \mid x)$ is maximized under an appropriate model p

[See also Jelinek, 1998]. With appropriate definitions for the conditional probability densities, this decomposes by Bayes' Theorem into a constant multiple of the product $p(x \mid w)p(w)$, whose factors are termed the *acoustic model* and *language model* respectively. Language models, in this formulation, represent priors $p(w)$ on the sample space of all word strings. Intuitively, a good model will allocate much of the probability mass to word strings that are meaningful and plausible in the expected context, and will penalize or disqualify word strings that are not.

Machine translation The statistical approach to machine translation is often formulated as the following problem: given a source sentence f, find the translation into a target sentence e that maximizes the posterior probability under an appropriate model [Brown et al., 1990, 1993]. As in speech recognition, this has a decomposition as

$$e^* = \arg\max_e p(e \mid f) = \arg\max_e p(e)p(f \mid e),$$

which Brown et al. [1993] call the 'Fundamental Equation of Machine Translation', although the same principle applies equally well to many other pattern recognition tasks. The term $p(e)$ is a language model—a prior distribution on sentences e in the target language. More recent papers describing the role of language models in machine translation are [Daumé et al., 2002; Charniak et al., 2003].

The maximum-entropy formalism has also been applied to machine translation, starting with the influential paper of Berger et al. [1996], and, more recently, by Foster [2000]; Och and Ney [2002]; Varea et al. [2002]; Ueffing and Ney [2003].

Text prediction and typing aids Predictive typing aids exploit the redundancy of natural languages to increase the efficiency of textual input. Two approaches to doing this are to increase the keystroke rate and to reduce the required number of

keystrokes. Zhai et al. [2002] propose and review various configurations of virtual keyboards usable with a stylus, aiming to minimize the distance between common pairs of consecutive characters. Predictive text-input has been employed for entering text in oriental languages for years; it also has applications for disabled users, such as Dasher [Ward and MacKay, 2002] and FASTY [Trost et al., 2005]; and has found rudimentary but widespread application in mobile phones (T9, LetterWise, Zi Corp) and various PDA operating systems. Typing aids use language models either to predict further input, by suggesting completions of partial words or sentences, or to disambiguate existing input, when too few keys are available to represent the alphabet (as with current phones). Typing aids can also be used for multimodal disambiguation of speech recognition hypotheses, as I demonstrated with Zhiping Zheng in [Schofield and Zheng, 2003].

5.1.1 Language modelling research

There is ample evidence that humans make significant use of prior knowledge of syntax, semantics, and discourse when interpreting speech and text. This is described in the psycholinguistics literature (see, for example, Aitchison [1998]) and accords with common sense. Without this knowledge, as for a foreign language, we cannot segment words from continuous speech; with it, we can often infer the meaning of blurred text or an audio stream garbled by noise.

n-gram models

The original form of language model proposed by Bahl et al. [1983] was as follows. First decompose the probability $\Pr w$ of the word string $w = (w_1, ..., w_m)$ as

$$\Pr(w_1)\Pr(w_2 \mid w_1) \ldots \Pr(w_m \mid w_1, w_2, \ldots, w_{m-1}).$$

Now model the k^{th} term in the above product as

$$p(w_k \mid w_{k-n+1}, w_{k-n+2}, \ldots, w_{k-1})$$

for an appropriate conditional model p. This is equivalent to an $(n-1)^{\text{th}}$ order Markov assumption. Models of this form, known as n-gram models, can, for small enough values of n, be inferred from conditional frequencies in corpora of text. Such models are simple and, for languages such as English without a complex inflectional morphology, quite effective (but significantly worse for Finnish [Siivola et al., 2001]), and have found widespread use despite their obvious shortcomings.

Some of these shortcomings are readily made apparent by inspecting a small sample of text generated according to an n-gram model. Here is a sample generated from a 3-gram model fit by Jurafsky and Martin [2000] to the Wall Street Journal corpus [Paul and Baker, 1992]:

> They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions.

This is an admittedly poor n-gram model, and a longer n-gram model well estimated from more data would likely look different. Even with a better n-gram model, though, the vast majority of sentences randomly generated from the model are likely to be ungrammatical; this is because natural languages do not satisfy the Markov lack-of-memory property.

Syntactic models

Context-free grammars (CFGs) are often used to specify the set of permissible phrases for task-specific speech-recognition systems, such as for call centers. Such grammars are straightforward, if tedious, to specify, and typically yield lower error rates for limited-domain speech-recognition tasks than more flexible language models, such as n-grams. They are less useful for free-form dictation or transcription of conversational speech, since obtaining a fair coverage of a natural language requires, at the least, a large and complex grammar. Indeed, natural languages are widely thought not to be context-free [Shieber, 1985], implying that many linguistic phenomena will

always remain out of the reach of CFGs. This is only partly solved by more flexible grammars, such as head-driven phase-structure grammars [Levine and Meurers, 2006], which are more general than context-free grammars but remain somewhat tractable computationally.

Manning [2003] argues that the boundaries of linguistic phenomena on which theoretical linguistics tends to focus are a complementary source of knowledge to the vast linguistic middle ground accessible to corpus linguistics, and that these could be fused in an appropriate probabilistic model. Maximum entropy is one framework that can serve this purpose.

5.1.2 Model combination and maximum entropy

There has traditionally been a divide between two approaches to language modelling; this mirrors a more general divide within natural language processing (NLP) and several fields requiring pattern recognition. Data-driven approaches, which have been largely pioneered by engineers for specific applications such as speech recognition, feed ravenously on corpora of text as their primary knowledge source. Knowledge-driven approaches to NLP are to encode linguistic phenomena into formal grammars and rule systems. The two perspectives are illustrated well by two quotes:

> But it must be recognized that the notion "probability of a sentence" is an entirely useless one, under any known interpretation of this term.
>
> *Noam Chomsky*, 1969, p. 57

> Any time a linguist leaves the group the recognition rate goes up.
>
> *Fred Jelinek*, IBM speech group, 1988[1]

Chomsky's claim that the 'probability of a sentence' is a meaningless concept may rest on the incorrect assumption that not observing an event forces us to estimate it with probability zero, a point well made by Manning [2003]. Jaynes [1982] described

[1] at the Workshop on the Evaluation of Natural Language Processing Systems, described by [Palmer and Finin, 1990]

such a belief as a 'frequentist preconception', and how maximum-entropy principle, in contrast, does not assign zero probability to any situation unless the prior information and data rule it out. Indeed, the maximum-entropy framework has gained particular momentum in various sub-fields of NLP recently, including language modelling.

The primary reason for the divide appears to have been a lack of understanding about how quite different information sources such as corpus data and theoretical linguistics can be reconciled. Research efforts have been underway to gain such an understanding, particularly since the early 1990s, and to build tools to fuse knowledge from various sources for practical systems.

Word-by-word maximum-entropy models

The maximum-entropy framework was first applied to language modelling by della Pietra et al. [1992], and was then pursued extensively by Rosenfeld [1994, 1996], with considerable success. The approach these authors took, later adopted by [Peters and Klakow, 1999; Khudanpur and Wu, 2000], was to model the occurrence of words w as a stochastic process $p(w \mid h)$ conditional upon the history h of words so far in the sentence or document. Maximum-entropy language models of this conditional form have the structure

$$p(w \mid h) = \frac{1}{Z(h)} \exp(f(w \cdot h)\theta). \tag{5.1}$$

This is slightly more general than the unconditional model form (2.4) considered in this thesis. One noteworthy difference is that the partition function Z in the conditional model form is a function of the conditioning context h.

A drawback [Rosenfeld, 1994] observes with the maximum-entropy formalism for language models is that parameter estimation can be computationally expensive. This can be somewhat mitigated by using more efficient optimization algorithms, as described in [Malouf, 2002] and Section 4.4, and more efficient representations of the underlying data, as described in Section 2.6.

The drawbacks to maximum-entropy language models are, however, less significant than the possibilities they open up. Most significant among these is the flexibility

to fuse different sources of prior information about language—be it a hand-written grammar or corpus-derived frequency information—in a consistent way. Also significant is that the resulting fusion is, in certain senses, an optimal one (see [Shore and Johnson, 1980] and recall Chapter 2). The traditional means for smoothing n-gram models inferred through maximum likelihood estimation—linear interpolation and 'backoff'—are, in fact, a fine illustration of the limitations of maximum likelihood estimation for practical inference problems.

5.1.3 Whole-sentence maximum-entropy models

The word-by-word sequential framework underpinning the n-gram and conditional maximum-entropy language models described above is natural for some applications, such as predictive typing aids. For other applications, it can be convenient for computational reasons, such as Viterbi decoding in speech recognition, but has clear shortcomings for modelling syntax and other whole-sentence phenomena. Modelling language over a sample space of whole sentences brings more flexibility. Rosenfeld et al. [2001] proposed such language models for whole sentences x of the same exponential form

$$p(x) = \frac{1}{Z(\theta; p_0)} p_0(x) \exp(f(x) \cdot \theta) \tag{5.2}$$

as considered in this thesis. Recall from Section 2.4 that models p of this form arise when minimizing the relative entropy $D(p\|p_0)$ from a prior distribution p_0 subject to constraints on the expectations of the features f_i. Examples of features f_i for a model of English sentences are the indicator functions $f_1(x) = 1\{x \text{ contains the } n\text{-gram } w\}$ and $f_2(x) = 1\{x \text{ parses under the grammar } G\}$. Expectation constraints on f_1 and f_2 would then correspond to constraints on the probabilities that the respective predicates are true.

Whole-sentence models also have two strong advantages over word-by-word models. First, they are highly flexible, allowing any predicate or real-valued property of a sentence as a constraint without imposing artificial independence assumptions. Second, the normalization term Z for any fitted model is a true constant (rather

than a function of a conditioning context), allowing very fast evaluation of sentence probabilities in applications, such as re-scoring hypotheses for speech recognition and machine translation. Their primary disadvantage, as noted above, is that they are less well suited to applications that require sequential processing. The informal nature of spontaneous speech, with its interruptions in flow, grammatical sloppiness, and other imperfections, may also be an obstacle to the exploitation of sentence-level syntactic structure. Note, however, that real-world spontaneous speech is nevertheless non-random, with its own structure, and several parsers already exist for it [Bub et al., 1997; Hinrichs et al., 2000; Gavaldà, 2000]).

The essential characteristic of these models that distinguishes them from the word-by-word models described in the previous section is not their being unconditional. Whole-sentence language models can be similarly extended to depend on an arbitrary conditioning context, such as a topic label or information derived from the previous sentences in a document, as Rosenfeld et al. [2001] describe. Rather, it is that the new sample space $\mathcal{X}$ of all possible sentences is countably infinite and no longer feasible to enumerate in practice.

Review of the parameter estimation framework

This section presents a brief summary of the discussion and results of Chapters 3 and 4 to make this chapter more self-contained. It contains no new information, and a reader familiar with Chapters 3 and 4 may wish to skip to the next section.

The problem we consider is of finding the distribution P that minimizes the Kullback–Leibler divergence $D(P\|P_0)$ between P and a prior model P_0, subject to linear constraints

$$\mathrm{E} f_i(X) = b_i \qquad \text{for } i = 1, \ldots, m, \tag{5.3}$$

on the expectations of the feature statistics f_i. Recall from Section 2.4 that this can be done by minimizing the entropy dual

$$L(\theta) = \log Z(\theta) - b \cdot \theta, \tag{5.4}$$

which is a convex function of the parameters θ, and is therefore minimized where its partial derivatives

$$\frac{\partial L}{\partial \theta_i} = \mathrm{E} f_i(X) - b_i \tag{5.5}$$

are zero, where p_θ is given by (5.2). For discrete sample spaces $\mathcal{X} = \{x_1, \ldots, x_n\}$, the partition function Z is given by

$$Z(\theta; p_0) = \sum_{j=1}^{n} p_0(x_j) \exp(f(x_j) \cdot \theta). \tag{5.6}$$

For whole-sentence models, neither $Z(\theta)$ nor $\mathrm{E} f_i(X)$ can be computed explicitly, because summation over the sample space of all possible sentences is infeasible. This thesis describes using Monte Carlo methods to estimate these functions. A basic Monte Carlo estimator for the feature expectation $\mathrm{E} f_i(X)$ is the sample mean

$$\hat{\mu}_i = n^{-1} \sum_{j=1}^{n} f_i(X_j), \tag{5.7}$$

where $X_1, \ldots, X_n$ are random variates (here, sentences) drawn independently from the model P. This estimator is often sub-optimal, in terms of its variance and the computational expense for generating the variates X_j according to the model. The idea of importance sampling, as Chapter 3 described, is to use knowledge of the integrand to focus more attention on those parts of the sample space that have the largest impact on the estimates. Here we generate random sentences $X_1, \ldots, X_n$ from some auxiliary distribution Q, rather than the model P itself, and estimate Z and $\mathrm{E} f(X)$ as

$$\hat{Z} \equiv n^{-1} \sum_{j=1}^{n} \dot{W}_j \tag{5.8}$$

and

$$\hat{\mu}_i \equiv \frac{1}{n\hat{Z}} \sum_{j=1}^{n} \dot{W}_j f_i(X_j), \tag{5.9}$$

where $\dot{W}_j$ is the j^{th} weighting term

$$\dot{W}_j \equiv \dot{p}(X_j)/q(X_j) = p_0(X_j)\exp(f(X_j)\cdot\theta)/q(X_j). \tag{5.10}$$

Here $\hat{\mu}$ is the *ratio estimator* of μ, which is in general biased, but quite efficient. The more common *integration estimator*, and the *regression estimator* [Hesterberg, 1988], are not applicable without knowing the normalization term Z in the denominator, which is no more feasible to compute here than μ. Whereas Z and μ are deterministic functions of the parameters θ, their estimators $\hat{Z}$ and $\hat{\mu}$ are random functions of θ. Modifying the parameters to fit the desired model based on these estimators is therefore a problem in stochastic optimization.

5.2 Experimental setup

This chapter evaluates the methods described in Chapters 2–4 when applied to estimating whole-sentence language models. The focus is on computational efficiency, not on the quality of the resulting models, which depends crucially on the chosen constraints.

Where possible, the test conditions mirror those of Rosenfeld et al. [2001]. The corpus used is Switchboard [Godfrey et al., 1992], with a 3-gram auxiliary model Q fit to 90% of it, about 2.9 million words, using Good–Turing discounting [Good, 1953].

Other particulars of the experimental conditions used in this chapter are as follows. The perplexity values reported in Sections 5.4.2 and 5.4.3 [described in Bahl et al., 1983; Jurafsky and Martin, 2000] are computed per word on a held-out testing set of 10% of the corpus, excluding any end-of-sentence symbol, and assuming a closed vocabulary, in which sentences in the external corpus with out-of-vocabulary words are ignored. The criterion used for convergence was the estimated gradient norm dropping below a threshold of 5×10^{-2}. All timing tests were conducted with a single 2.8 GHz Pentium 4 CPU. Appendix A describes the software used for these

evaluations; Appendix B describes the procedure used for sampling n-gram sentences.

Three sets of features are used in these tests:

1. $\mathcal{S}_1 = \{f_i\}$, where:
   - $i = 0, \ldots, 326561$: $f_i(x) = 1\{x \text{ contains the } n\text{-gram } g_i\}$, for word n-grams in the corpus up to $n = 5$;
   - $i = 326562, \ldots, 521539$: $f_i(x) = 1\{x \text{ contains the trigger } t_i\}$, for word triggers in the corpus (defined in [Rosenfeld, 1996]);
   - $i = 521540, \ldots, 521558$: $f_i(x) = 1\{\text{length}(x) \geq k\}$, for $k = 2, \ldots, 20$.
2. $\mathcal{S}_2 = \{f_i\}$, where:
   - $i = 0, \ldots, 11113$: $f_i(x) = 1\{x \text{ contains the trigger } t_i\}$, for word triggers that occur at least 100 times in the corpus and at least 100 times in the sample;
   - $i = 11114, \ldots, 11132$: $f_i(x) = 1\{\text{length}(x) \geq k\}$, for $k = 2, \ldots, 20$.
3. $\mathcal{S}_3 = \{f_i\}$, where:
   - $i = 0, \ldots, 18$: $f_i(x) = 1\{\text{length}(x) \geq k\}$, for $k = 2, \ldots, 20$.

Sentence-length features

The n-gram and trigger features have been described elsewhere [e.g. Rosenfeld, 1996], but some comments on the sentence-length features are in order. Researchers who have applied n-gram language models have tended to deal with the ends of sentences by adding a special token, such as `</s>`, designating the sentence end to the vocabulary, and treating it for modelling and simulation as another word. I describe briefly now why this biases the marginal distribution of the lengths of sentences under such models.

Consider models $p(x)$ for sentences $x = (w_1, \ldots, w_{k(x)-1}, \texttt{</s>})$. First consider a uniform (0-gram) model, under which the probability of occurrence of any word

(including the token </s>) is equal. The probability of the sentence x under this model is

$$p(x) = (1 - \pi)^{k-1}\pi.$$

where π is the probability that the current word at any given position is the last in the sentence. For a vocabulary of size $|V|$, π is given by $\pi = (|V| + 1)^{-1}$. The distribution of sentence lengths $k(X)$ here is *geometric*, with expectation $1/\pi = |V| + 1$.

Now consider a 1-gram model, under which

$$p(x) = \prod_{w=1}^{k-1} p_w \pi.$$

As before, the sentence length is a random variable equal to the number of Bernoulli trials until the first occurrence of </s>—but under this and higher-order n-gram models, these trials are no longer IID, and the marginal distribution of sentence lengths is no longer precisely geometric. Figure 5.1 shows a histogram of the actual distributions of sentence lengths in the corpus and in the n-gram sample. The third group of features is chosen to constrain the marginal distribution of sentence lengths under the model to equal that of the corpus.

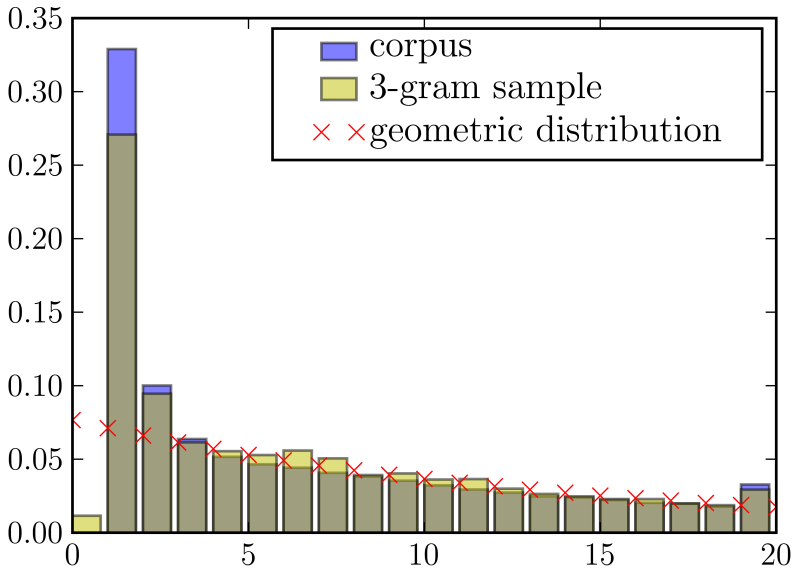
5.3 Tests of sampling methods

This section tests the computational efficiency of the importance sampling estimators from Chapter 3 and their matrix formulations in Section 4.4. The benchmarks for comparison are the sampling methods and estimators tested by Rosenfeld et al. [2001].

The first benchmark is the independent Metropolis–Hastings algorithm. Rosenfeld et al. [2001] described initial tests with this algorithm, concluding that its efficiency was comparable to that of importance sampling.

The second benchmark is the importance sampling setup of Rosenfeld et al. [2001]. Under this setup an MCMC method is used to generate an artificial corpus from an

Figure 5.1: Histogram of lengths of sentences sampled from an n-gram model for $n = 3$



initial model Q such as an n-gram, then importance sampling is used to re-weight the estimators over this artificial corpus at each iteration.

The tests in this section use the same algorithm for parameter estimation as that used for the tests of Amaya and Benedí [2001] and Rosenfeld et al. [2001]; this is the approximation of Rosenfeld et al. [2001] to the *improved iterative scaling* (IIS) algorithm [Della Pietra et al., 1997]. This algorithm is now somewhat obsolete, but this section adopts it to simplify comparisons with previously published results.

Amaya and Benedí [2000] proposed the use of the 'Perfect Sampling' algorithm of Propp and Wilson [1996, 1998] as an alternative to Markov Chain Monte Carlo MCMC algorithms such as the Metropolis–Hastings algorithm tested here. In contrast to MCMC methods, which yield a sample from an approximation that tends to this distribution only in the limit as the number n of samples grows without bound, the 'Perfect Sampling' algorithm yields a sample from the exact equilibrium distribution.

Simple Monte Carlo estimators based on 'Perfect Sampling' are therefore unbiased. Recall that the importance sampling integration estimator (3.6) is also 'perfect' in that it is unbiased, but that this is of little importance in practice, since other estimators (such as the ratio estimator) tend to offer greater stability and lower variance in a variety of contexts [see Hesterberg, 1988]. The Perfect Sampling algorithm also incurs some additional computational complexity over the Metropolis–Hastings algorithm, so this section has taken the latter as a benchmark.

Tables 5.1–5.3 show timing results for the various steps involved in fitting a whole-sentence model to the features f_i in the set $\mathcal{S}_0$ described in Section 5.2. Table 5.1 is for the independent Metropolis–Hastings algorithm; Table 5.2 is for importance sampling as described by Rosenfeld et al. [2001]; Table 5.3 is for importance sampling using the matrix formulation of Section 4.4.1. The timings, although dependent upon the CPU (a 2.8 GHz Pentium 4) and the implementation, indicate the approximate relative importance of each component's contribution to the overall computational requirements. This is more informative than $O(k)$ relations in the number k of iterations, since abstracting away these timings as mere 'constant factors' would obscure the observation that they vary greatly, over several orders of magnitude, a fact that has significant implications for the choice of algorithms to use for parameter estimation.

Rosenfeld et al. [2001]'s account is not explicit about which steps they repeat each iteration and which they pre-compute only once. The authors only acknowledge explicitly that the use of importance sampling allows a single sample to be drawn in step (a). This statement is quite true, but can be strengthened; only one computation of the features f of this sample, in step (d), is necessary, since these are invariant in the parameters θ. The original sample can then be discarded entirely for the purposes of fitting the model, with corresponding memory benefits. The same is, of course, true of the corpus, as Section 2.6 [last para.] described, so steps (b) and (c) need only be performed once.

This is a reasonably large test, with $m = 521,559$ features. Note that the corpus of $n = 217,452$ sentences is relatively small compared to others used in NLP tasks,

Table 5.1: Timings for the steps involved in fitting a whole-sentence model using the independent Metropolis–Hastings (MH) algorithm for sampling. k is the number of iterations of GIS necessary to fit the model; reported times are the best of 3 trials. Trial times in steps (c) and (g) are averages of 20 and 100 repeats, respectively.

	Step	Time per call	# calls
(a1)	Sample initial artificial corpus $\mathcal{S}_0 = x_1, \ldots, x_{10^5}$	1145 s	1
(a2)	Re-sample $\mathcal{S}$ from $\mathcal{S}_0$ using MH		k
(b)	Compute corpus features $f(x_j),\ x_j \in \mathcal{C}$	972 s	k †
(c)	Compute means $\lvert\mathcal{C}\rvert^{-1} \sum f(x_j)$ over $x_j \in \mathcal{C}$	8.46 s	k †
(d)	Compute sample features $f(x_j),\ x_j \in \mathcal{S}$	406 s	k †
(e)	Estimate expectations $\mathrm{E}f(x_j)$ using the sample $\mathcal{S}$	8.71 s	k
(f)	Initialize GIS step-size computation	31.6 s	1
(g)	Compute GIS step sizes	1.65 s	k

Table 5.2: Timings for the steps involved in fitting a whole-sentence model using the importance sampling algorithm for sampling, as tested by Rosenfeld et al. [2001]. k is the number of iterations of GIS necessary to fit the model; reported times are the best of 3 trials. Trial times in steps (c) and (g) are averages of 20 and 100 repeats, respectively.

	Step	Time per call	# calls
(a)	Sample artificial corpus $\mathcal{S} = x_1, \ldots, x_{10^5}$	1145 s	1
(b)	Compute corpus features $f(x_j),\ x_j \in \mathcal{C}$	972 s	k †
(c)	Compute means $\lvert\mathcal{C}\rvert^{-1} \sum f(x_j)$ over $x_j \in \mathcal{C}$	8.46 s	k †
(d)	Compute sample features $f(x_j),\ x_j \in \mathcal{S}$	406 s	k †
(e)	Estimate expectations $\mathrm{E}f(x_j)$ using the sample $\mathcal{S}$	8.71 s	k
(f)	Initialize GIS step-size computation	31.6 s	1
(g)	Compute GIS step sizes	1.65 s	k

† It is not clear from the account of Rosenfeld et al. [2001] whether the authors computed the features of the corpus and the sample (artificial corpus) once or re-computed these at each iteration. Discussed below.

Table 5.3: Timings for the steps involved in fitting a whole-sentence model using the matrix–vector formulation of importance sampling from Chapter 4. k is the number of iterations of GIS necessary to fit the model; reported times are the best of 3 trials. Trial times in steps (c) and (g) are averages of 20 and 100 repeats, respectively.

	Step	Time per call	# calls
(a)	Sample artificial corpus $\mathcal{S} = x_1, \ldots, x_{10^5}$	1145 s	1
(b)	Compute corpus features $f(x_j),\ x_j \in \mathcal{C}$	972 s	1
(c)	Compute means $\lvert\mathcal{C}\rvert^{-1} \sum f(x_j)$ over $x_j \in \mathcal{C}$	8.46 s	1
(d)	Compute sample features $f(x_j),\ x_j \in \mathcal{S}$	406 s	1
(e)	Estimate expectations $\mathrm{E}f(x_j)$ using the sample $\mathcal{S}$	8.71 s	k
(f)	Initialize GIS step-size computation	31.6 s	1
(g)	Compute GIS step sizes	1.65 s	k

but that the corpus size is irrelevant to the computational complexity for all steps except (b) and (c). This point was also made in Section 2.6; only a single pass is necessary over the corpus, so large corpora can be used without affecting the speed of parameter estimation with exponential-form models. Note, in particular, that the sample space is 'large', irrespective of either m or n, in the sense that simulation is necessary for step (e).

An efficient matrix formulation is (as far as I know) only possible with importance sampling or other variance reduction methods (such as control variates), not with a resampling method such as Metropolis–Hastings or Perfect Sampling. The reason this is so effective with importance sampling is that a single sample can be used to estimate the entire entropy dual or likelihood function, a fact also observed by Geyer and Thompson [1992]. Importance sampling is a natural fit for efficient stochastic optimization whenever variate generation and feature computation are relatively expensive, since obtaining estimates at each iteration requires only a re-weighting of a pre-drawn sample. The same is not true for basic Monte Carlo estimates, which are averages over a sample that must be re-drawn each iteration.

5.4 Tests of optimization algorithms

This section describes three tests in the use of the estimation framework of this thesis to fit whole-sentence models. The first test is an attempt to fit a pure maximum-entropy model using the set $\mathcal{S}_1$ of $m = 521,559$ features f_i described in Section 5.2. These features include n-grams, in particular, just as in the study of Rosenfeld et al. [2001]. The idea of the second and third tests, in contrast, is to estimate an n-gram component of a language model by the usual (maximum likelihood) method, and tweak this model to satisfy additional constraints reflecting any other knowledge about the language.

Any model P_0, not just an n-gram, can be used as a baseline in this setup, provided it can be computed up to a (possibly unknown) constant factor. If this baseline is known to have flaws at the whole-sentence level, which can be characterized as a violation of some constraints of the form $Ef_i(X) = b_i$ for any arbitrary functions f_i and real values b_i, then the estimation framework studied in this thesis can be used to find a new model P that satisfies these constraints (5.3) while otherwise remaining as close as possible to the baseline P_0 in the sense of having minimal KL divergence from it. Recall from Section 2.4 that the moment constraints then induce an exponential-form prior that modifies the existing model; the form of the resulting probability mass function is given by (5.2).

This section evaluates the use of sample-path optimization. This section does not explicitly test stochastic approximation (SA) for fitting language models. The reason is that SA, like the Metropolis–Hastings sampler (Table 5.1), requires k iterations of steps (a) and (d) (recall Section 4.5). Since these steps are expensive—among the most expensive overall in the estimation process—this implies that stochastic approximation will be expensive overall. This conclusion is supported by various informal tests I have conducted, but I omit a detailed account of these here, since they impart no new insight.

A primary focus of Chapters 3 and 4 was the derivation of estimators for the entropy dual (2.11) and its gradient (2.12). These chapters showed that there is

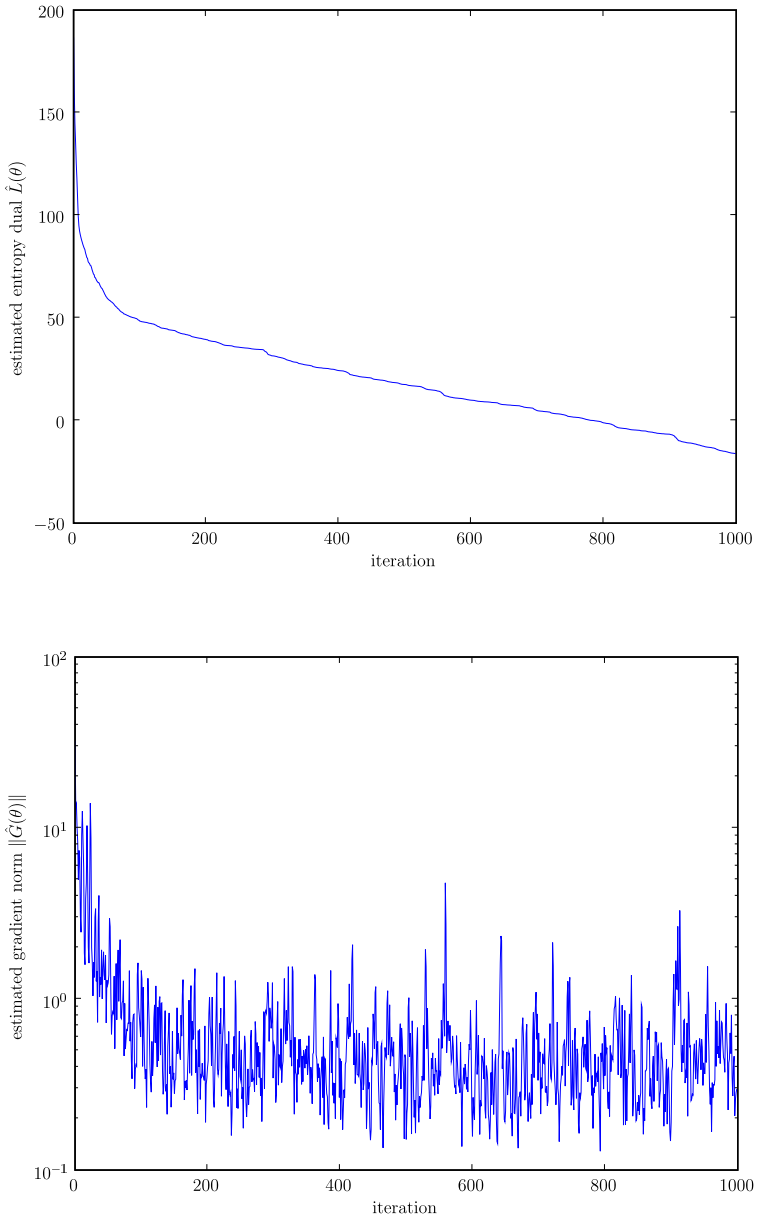
little cost overhead for estimating the entropy dual if its gradient must be estimated anyway; this indicates that optimization algorithms that use both the dual and its gradient, such as quasi-Newton algorithms, are likely to be more efficient overall than algorithms that use only the dual or only the gradient. Note that iterative scaling methods use neither of these, and will not be considered for this section.

Quasi-Newton methods are known to be highly efficient [Boyd and Vandenberghe, 2004], in the sense of converging to an optimum in few iterations, but these do not scale particularly well to optimization problems in a large number m of variables. Some require computation of a Hessian matrix, of size m^2, at each iteration; others save some of this computation by building up an approximation to this over several iterations. But for the set $\mathcal{S}_1$ of $m = 521559$ features, even storing the Hessian is out of the question.

Malouf [2002] evaluated a limited-memory variable-metric (LMVM) algorithm and the method of conjugate gradients (CG) for fitting exponential models for a variety of NLP tasks, concluding that both are relatively efficient (compared to iterative scaling). This section uses the Polak–Ribière conjugate gradient (CG-PR) method [Polak and Ribière, 1969]. One reason is that its space requirements scale very well with the number of variables: linearly in the size m of the parameter vector θ. Another is that implementations are readily available in most scientific software environments.

Note that this choice is an example only; what follows is not an experimental study of the stability or efficiency of different algorithms in sample-path optimization. The precise behaviour of the conjugate-gradients algorithm in this context depends on both the problem and the details of the implementation, including various magic constants chosen from a range of values for which convergence has been proven [Polak and Ribière, 1969; Grippo and Lucidi, 1997], so these results only indicate what is achievable.

Figure 5.2: Estimates of the entropy dual and gradient norm during sample path optimization with PR conjugate gradients, using a sample of size 10^5, with constraints on features $f_i \in \mathcal{S}_1$.



5.4.1 Test 1

The first test uses the set $\mathcal{S}_1$ of $m = 521,559$ features outlined in Section 5.2. Figure 5.2 shows plots of the evolution of the estimated entropy dual $\hat{L}$ and the norm of its gradient over 1000 iterations of conjugate gradients (CG-PR). The plot of the gradient norm shows that the procedure is not converging. Some investigation reveals that some of the parameters θ_i become increasingly large during the procedure, perhaps diverging to $+\infty$, while others are becoming increasingly negative.

The features in $\mathcal{S}_1$ include n-grams and triggers, some of which never occur in the sample of 10^5 sentences from the auxiliary distribution (see Table 5.4). The parameters θ_i corresponding to these features f_i are those growing without bound. This is unsurprising; the expectations $\mathrm{E}f_i(X)$ of any such features using importance sampling are always zero, so the corresponding gradient components $\hat{G}_i = \hat{\mu}_i - b_i$ are (for positive features) always negative.

Table 5.4: The number of features $f_i(X)$ from the set $\mathcal{S}_1$ occurring at different frequencies in the auxiliary sample (of $n = 10^5$ 3-gram sentences).

k	# features that occurred k times
0	69,167
1	55,971
2	41,360
< 10	277,948
< 100	348,956
< 1000	359,294

5.4.2 Test 2

The second test uses the smaller set $\mathcal{S}_2$ of $m = 11,133$ features in a divergence-minimization context, rather than a pure maximum-entropy context. The features in this set do not include n-grams; these are instead represented through a prior P_0 used for divergence minimization, which is an n-gram model estimated by traditional

means (maximum-likelihood with Good–Turing discounting [Good, 1953]). It also includes trigger features limited to those that occurred both 100 times or more in the sample and 100 times or more in the corpus, in order to eliminate features that occur too rarely for reliable estimation.

Figure 5.3 shows plots of the evolution of the estimated entropy dual $\hat{L}$ and its gradient for the features in set $\mathcal{S}_2$. Estimating the parameters took about 30 seconds, under one second per iteration. For comparison, the tests of Rosenfeld et al. [2001, Sec. 4.2], with similar numbers of features, required 'less than three hours on a 200 MHz Pentium Pro computer' for 50 iterations of iterative scaling.

The per-word perplexity of the testing set (10% of the entire corpus) under the fitted model P was 5.6% lower (131.9 versus 138.2) than under the 3-gram prior p_0.

5.4.3 Test 3

This test shows how imposing even a small set of constraints on a baseline model can help to correct its flaws at the sentence level. This test uses the smaller set $\mathcal{S}_3$ of $m = 19$ features representing sentence lengths. Fitting this model required 29 iterations, with 50 function and 50 gradient evaluations, which took about 5 seconds, again using a 2.8 GHz Pentium 4 and the matrix importance-sampling estimators of Chapter 4 with conjugate gradients. Figure 5.4 shows the evolution of the estimated entropy dual and gradient.

The marginal histogram of sentence lengths for a sample from the fitted model is then indistinguishable from that of the corpus. The per-word perplexity of the testing set (10% of the entire corpus) under the fitted model P was 2.0% lower (135.5 versus 138.2) than under the 3-gram prior p_0.

5.5 Discussion

Most language models employed in systems for speech recognition and machine translation are not short of violations of syntax and other blatant flaws. One way to highlight these is to generate random sentence-like strings according to the models,

Figure 5.3: Estimates of the entropy dual and gradient norm during sample path optimization with conjugate gradients, using a sample of size 10^5, with constraints on features $f_i \in \mathcal{S}_2$.

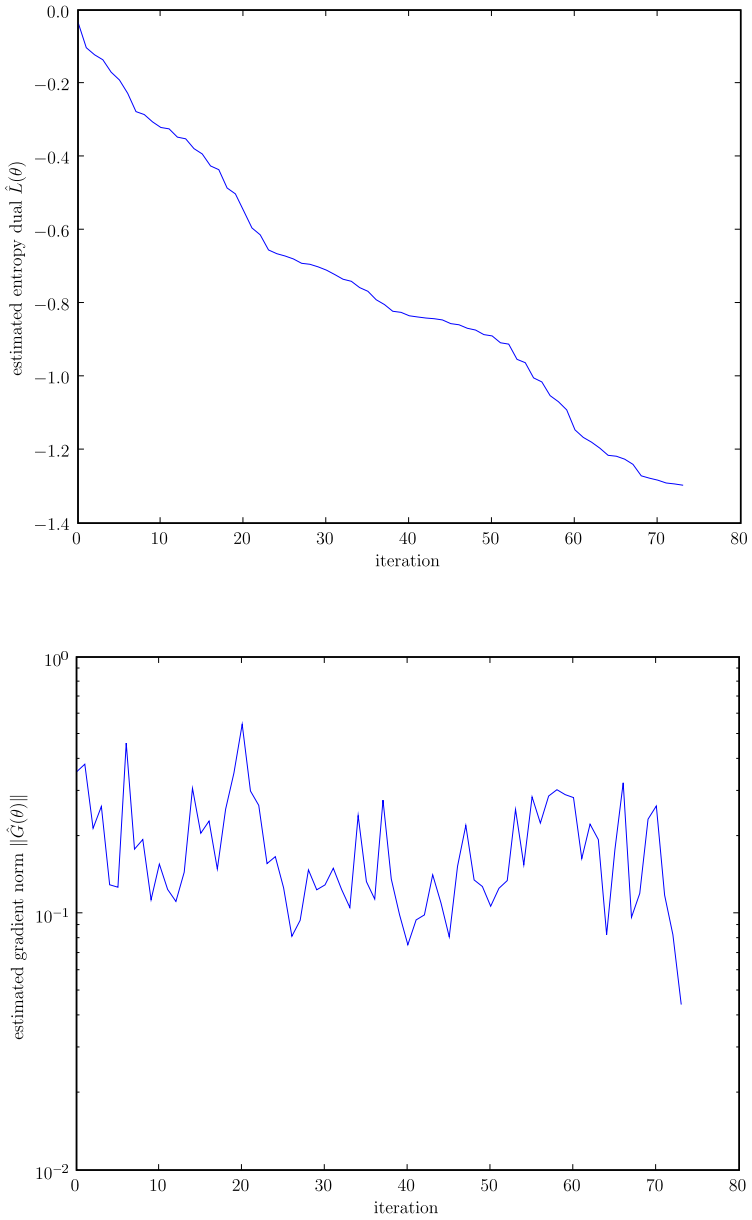
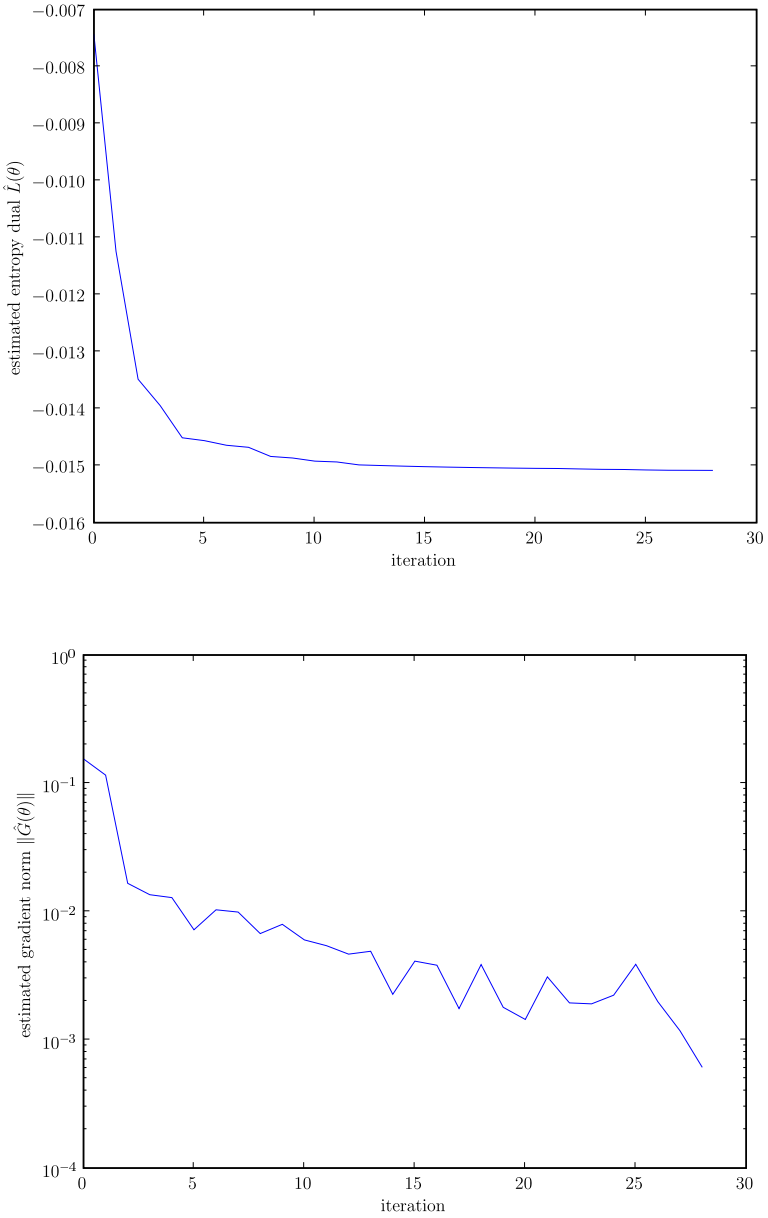


Figure 5.4: Estimates of the entropy dual and gradient norm during sample path optimization with conjugate gradients, using a sample of size 10^5, with constraints on features $f_i \in \mathcal{S}_3$.



using, for example, the Metropolis–Hastings sampler. This presents an opportunity for modelling: generate random sentences from a language model; this will readily reveal flaws upon inspection, which can be encoded into further constraints to improve the model. Rosenfeld et al. [2001] pursued this idea in an automated fashion, selecting sets of lexical and syntactic constraints from a larger pool based on the result of a hypothesis test for whether a random sample generated from the baseline model differed significantly from the corpus. Their results, however, were disappointing—a reduction in per-word perplexity of 1% versus a 3-gram prior, using 6798 constraints.

One hypothesis for why Rosenfeld et al. [2001] observed these results is that their models had, perhaps, not converged satisfactorily, leaving the desired constraints unsatisfied. This seems likely, given that the feature set included both n-gram and trigger features, both of which may easily occur too infrequently in a sample of $n = 10^5$ sentences for reliable estimates (see Table 5.4). In particular, Rosenfeld et al.'s choice to constrain n-gram features explicitly imposes a substantial penalty in computational overhead and imprecision versus the minimization of divergence under different constraints from an n-gram model estimated with traditional maximum-likelihood means. As Chapter 3 showed, the computational requirements of Monte Carlo methods depend largely on how rare (difficult to simulate) its features are. In this case, the variance of the estimators used by Rosenfeld et al. may, like in Section 5.4.1 have been too large for convergence with the given sample size.

Another (related) hypothesis for why Rosenfeld et al. [2001] observed these results is that they used 50 iterations of iterative scaling, rather than a specific convergence criterion such as the gradient norm dropping below a certain threshold, which would indicate satisfaction of the constraints to within a certain tolerance. The tests in Section 5.4 with the method of conjugate gradients required more iterations than this, yet conjugate gradient algorithms generally yield faster convergence than iterative scaling algorithms for fitting exponential-family models [Malouf, 2002; Minka, 2003].

The two successful tests here used divergence minimization, rather than pure entropy maximization, with sets of features that are frequent enough in both the corpus and auxiliary sample to make reliable estimation unproblematic. The perplexity

improvements with this set of features are the best yet reported for whole-sentence language models. These results show that, if the computational aspects are handled correctly, imposing even a small set of sentence-level constraints upon an existing model can be effective in correcting its deficiencies—and that Rosenfeld et al.'s highly flexible approach to language modelling, despite their initial disappointing results with it, merits further attention.

In summary, the ideal application to language modelling of the computational framework presented in this thesis is as follows:

1. Start from an existing model as a baseline, perhaps purely data-driven, like an n-gram model. Smoothing methods for n-gram models are mature, and n-gram frequencies contain a large amount of useful information for a language model.

2. Determine which weaknesses this baseline model has, in terms of whole-sentence properties it satisfies too frequently or too infrequently. Express these properties as indicator functions or other features f_i, and their ideal frequencies as b_i. One simple way to determine a model's weaknesses is by inspecting a random sample of sentences it generates.

3. Use sample-path optimization with importance sampling to find the model of minimum divergence from the baseline that satisfies constraints $\mathrm{E} f_i(X) = b_i$ on these features.

In particular, if n-gram features are incorporated into a whole-sentence language model, they should be included in a prior $p_0(x)$ estimated by traditional maximum-likelihood means and used for divergence minimization, rather than being reconstructed less precisely and more laboriously in exponential form, as Rosenfeld et al. [2001] attempted. Re-weighting a successful existing model simplifies the modelling task, yielding an exponential-form component with vastly fewer features. This has benefits for both efficiency and robustness during estimation.

A note is due on the scalability of this estimation framework to larger problems. Recall (from Section 2.6) that the parameter-estimation task scales trivially to larger

corpora, since the corpus is used only once to derive the vector of constraint targets, and can thereafter be discarded. The computational difficulty lies instead with rare features. Ideally we would have a precise theoretical characterization of the sample size necessary for reliable estimation with a given set of features. It seems unlikely that this is achievable with sample-path optimization, since convergence proofs for most optimization algorithms rely on properties such as the smoothness and differentiability of the objective function which do not hold for estimators derived from a finite sample.

For the practical application of this estimation framework to language modelling, several options exist for overcoming computational difficulties due to rare (difficult to sample) features. The first, of course, is to use coarser features, perhaps by grouping rare features together. Just as grouping words and n-grams into classes [Bahl et al., 1983; Brown et al., 1992] can increase the accuracy of the estimates and reduce the number of parameters of n-gram models, grouping features to impose as constraints would help to avoid sparsity of their occurrence in both a corpus and an auxiliary sample. A second option is to use a better auxiliary distribution, if possible, as described in Chapter 3. A third option is to use a larger sample. If the available computing resources continue to increase rapidly, increasing the sample size will be a relatively cheap and easy option.

5.6 Conclusions

Language models are used as a prior on the probability of sentences and other units of text for fields such as speech recognition and machine translation. Whole-sentence language models offer an appealing flexibility for modelling linguistic phenomena such as syntax, but introduce computational difficulties for fitting them. This chapter has examined the application of stochastic optimization using Monte Carlo estimation to exponential-form models of sentences.

Section 5.3 gave timing data for various component tasks for estimation with whole-sentence maximum-entropy models, and described how a combination of

importance sampling with sample-path optimization allows parameter estimation to be particularly efficient. The key reason is that the computational effort required for re-computing the importance sampling estimates at each iteration (using the matrix–vector formulation from Chapter 4) is negligible versus the effort for generating a random sample of sentences and evaluating its feature statistics.

Section 5.4 described tests fitting language models using importance sampling and sample-path optimization with three different sets of features. The first test included features that were too rare for reliable estimation with the given sample size, and the procedure did not converge. The second test excluded n-gram features and those trigger features that occurred rarely in the sample; the estimators were then accurate enough for sample-path optimization to yield significant improvements over a baseline model in a divergence-minimization setting.The third test, with sentence-length features, showed rapid convergence, demonstrating how minimizing divergence from a baseline model with even a small set of features can correct its sentence-level deficiencies.

Any language model from which one can sample, whatever its quality or complexity, can be tuned by using it as a baseline for divergence minimization. If the model is deficient at a whole-sentence level in some quantifiable way, this can be corrected by imposing additional constraints and fitting a new model that satisfies these constraints while minimizing KL divergence from the original model.

6 Example: truncated Gaussian models

This short chapter demonstrates the usefulness of the theory with a simple continuous modelling problem in n dimensions. Consider, in particular, the problem of finding the entropy-maximizing distribution P over $\mathbb{R}^n$ subject to three constraints

$$\mathrm{E} f_{i,d}(X) = b_{i,d} \quad \text{for } i = 1, 2, 3, \tag{6.1}$$

in each dimension d, where

$$\begin{aligned} f_{1,d}(X) &= X_d \\ f_{2,d}(X) &= {X_d}^2 \\ f_{3,d}(X) &= 1\left\{a_l < X_d < a_u\right\}. \end{aligned}$$

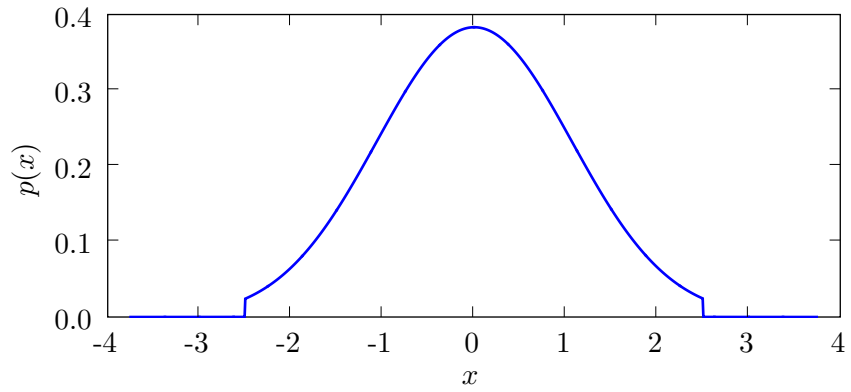
The first two constraints are on the non-central first and second moments. Constraining the marginal variances with $f_{2,d}(X) = (X_d - \mathrm{E}X_d)^2$ would be equivalent to constraining the non-central second moment as here. It is well-known [see Cover and Thomas, 1991] that a normal (Gaussian) distribution is the distribution of maximum entropy subject to the first two of these constraints. The third constraint serves to truncate the distribution in all dimensions d outside the interval (a_l, a_u).

Truncated Gaussian distributions appear in many contexts where some selection mechanism or physical constraint operates, from robotics [Cozman and Krotkov, 1994] to animal husbandry [Wake et al., 1998]. Cozman and Krotkov [1994] prove that truncated Gaussians are distributions of maximal entropy subject to constraints on the expectation and covariance matrix. They also give an expression for the form of the densities and derive algorithms for computing properties such as their

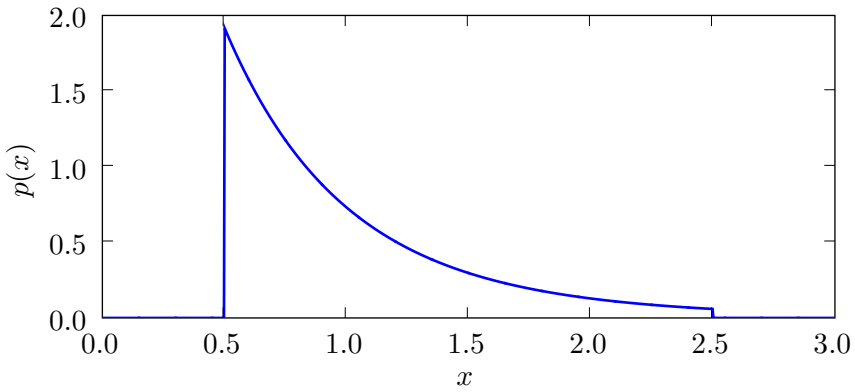
moments.

The theory developed in this thesis has provided a general framework for fitting large exponential-family models, of which the truncated Gaussian (illustrated in Figure 6.1) is merely a special case resulting from the simple choice of constraints above. The generality of the framework renders it unnecessary to derive individual algorithms specifically for estimating models of truncated Gaussians; these can be estimated efficiently using stochastic methods and the power of modern computers.

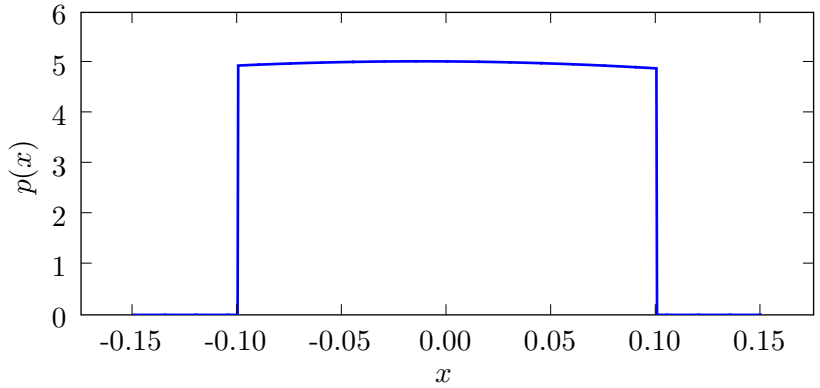
Appendix A quotes source code for estimating truncated Gaussian models effortlessly in hundreds of dimensions.



(a) A truncated Gaussian with $(a_l, a_u) = (-2.5, 2.5)$ and $b = (0, 1, 1)$



(b) A truncated Gaussian with $(a_l, a_u) = (0.5, 2.5)$ and $b = (1.0, 1.2, 1)$



(c) A truncated Gaussian with $(a_l, a_u) = (-0.1, 0.1)$ and $b = (0, 0.0033, 1)$

Figure 6.1: Some truncated Gaussian densities in $\mathbb{R}^1$ computed with the maximum-entropy method. Densities (b) and (c) are approximations to the examples in Cozman and Krotkov [1994, Fig. 1]

7 Conclusions

The thesis has investigated how models of maximum entropy, minimum divergence, and maximum likelihood within the exponential family can be estimated in a stochastic setting. In such a setting, some convenient properties of the equivalent deterministic estimation problem, such as the convexity of the entropy dual, no longer apply. The computational burden of parameter estimation for exponential models—seldom considered light—has the potential to increase dramatically if expensive Monte Carlo simulations are required at each iteration. The main insight contained in this thesis is that Monte Carlo simulations can be cheap in this setting. After drawing a fixed sample once, a matrix of its features can be recycled for all iterations, allowing the entropy dual function and its gradient to be estimated at each iteration with little more than two matrix–vector multiplications. One essential ingredient that makes this possible is that the maximum-entropy and minimum-divergence principles yield exponential-family densities, under which a sample can be expressed as a matrix of sufficient statistics. Another is that importance sampling can be applied highly efficiently in an iterative context to re-weight estimates as the model parameters change.

Maximum-entropy methods do not provide a recipe for automatic learning of relationships or structure in data. They solve only one part of this problem. They specify which model to infer, given prior knowledge, but not how this prior knowledge should be derived. This thesis likewise only concerns itself with the mechanical task of fitting the model that reflects this prior knowledge. Using this computational framework effectively in a given modelling context requires domain-specific knowledge encoded as moment constraints.

Maximum-entropy methods do, however, now have a track record of success in

various modelling domains. It is remarkable that the computational framework described in this thesis and the accompanying software implementation are identical whether the underlying sample space is discrete or continuous, equally applicable to the domains of natural language in Chapter 5 and truncated Gaussians in Chapter 6. With this thesis I hope to remove some barriers to the adoption of maximum-entropy methods for new applications and in new domains.

7.1 Applications and extensions

The main results of this thesis should be directly applicable to many modelling contexts, from wireless channels to financial derivatives. Some of its smaller results and observations could also be applied to other fields:

1. The logsumexp() expression given in Lemma 4.2 for avoiding numerical overflow while computing logs of sums of exponentials is probably applicable outside the context of entropy maximization or divergence minimization—such as for maximum-likelihood estimation with other parametric model forms; expectation maximization with mixture models; logistic regression; and integration arising in Bayesian inference. It may be underutilized in such contexts.

2. Inefficient sampling methods are sometimes used in iterative contexts, such for feature induction on Markov random fields [Della Pietra et al., 1997], where basic Monte Carlo estimates are obtained from an expensive Gibbs sample generated at each iteration. In some of these cases large computational benefits would accrue from applying importance sampling iteratively to re-weight a static sample.

3. Constant-time algorithms such as Marsaglia's (Appendix B) are probably underused in contexts that require discrete variate generation. This also concerns feature induction on Markov random fields, for which sampling is a bottleneck sometimes assumed to require linear time.

I believe the following ideas also merit further investigation:

1. The cardinality terms that appear in the expressions for the feature-space estimators (3.36) can be problematic to compute; this prevented me from applying Theorem 3.1 to estimate the language models in Chapter 5 by sampling in feature space. Perhaps these terms could be approximated or estimated by simulation without losing the benefits of sampling in the new space; this would make the theorem more applicable in practice, for language modelling and other applications. Recent developments in conditioning Monte Carlo estimators on sufficient statistics, such as [Lindqvist and Taraldsen, 2005], could guide such an effort.

2. An analogue of Theorem 3.1 for sampling in the derived space of features could be sought for continuous sample spaces and continuous features. This would involve the Jacobian of the transformation as an analogue of the discrete cardinality terms in Theorem 3.1. This may, like the discrete cardinality terms, be independent of the parameters.

3. Chapter 5 showed that imposing additional whole-sentence constraints upon data-driven language modelsin a minimum-divergence context can help to correct their deficiencies. The field of language modelling should benefit from a systematic study of linguistic features for use in this framework. Such features could highlight either positive features (such as permissible grammars for sentences) or negative features (such as grammars that describe ungrammatical sentences).

A Software implementation

This section describes the software I implemented during the course of this research for fitting exponential models, either with or without simulation. I originally made the software available as the 'FTW Text Modeller' `http://textmodeller.sf.net` (also under an Open Source license). I have now made the code for maximum entropy parameter estimation more generic and contributed it as a component of the larger Open Source project SciPy (see `www.scipy.org`), a set of packages for numerical and scientific computation written in the Python programming language. SciPy requires and builds upon NumPy (`numpy.scipy.org`, see also Oliphant, 2006), a package for efficient high-level manipulation of arrays.

Publishing this implementation as Open Source software should ensure that the experimental results are reproducible and should remove a large barrier to further research into maximum-entropy methods using simulation, and their applications.

My contribution to SciPy has three main components. The first is to the `sparse` package for efficient manipulation of sparse two-dimensional matrices. The second is the `montecarlo` package for efficient generation of discrete variates using Marsaglia's compact lookup-table algorithm (described in Section B.2). The third is the `maxentropy` package, which can use (but does not require) the functionality of the other two packages.

The `maxentropy` package provides several classes that represent exponential-form models, either of maximum entropy (2.4) or of minimum relative entropy (2.9) to a specified prior distribution. These classes are:

`model` for unconditional models on small sample spaces (where simulation is unnecessary);

`conditionalmodel` for conditional models on small sample spaces. Such models are appropriate for discrete classification problems, such as in natural language processing;

`bigmodel` for unconditional models on large sample spaces that require simulation. Such models were the focus of this thesis.

Various other tools now exist for fitting discrete, conditional maximum entropy models, such as the `OpenNLP maxent` package [Baldridge et al., 2005], the Python/C++ package by Zhang Le [2005], and the Toolkit for Advanced Discriminative Modelling (TADM) [Malouf et al., 2006]. The third is more general than the first two, in that it does not require feature values to be non-negative. It is also more efficient, since function and gradient evaluations are performed using matrix operations through CPU-optimized BLAS libraries. Estimation using the SciPy `maxentropy` package should be comparable in speed to using TADM, since both packages express function and gradient evaluations in terms of matrix operations and, in the case of dense matrices, these operations are performed by the same underlying BLAS libraries. The main distinguishing feature of the SciPy `maxentropy` package is its support for stochastic optimization for fitting continuous or large discrete models. I believe this is the only package available for fitting such models.

Documentation on all three packages (`sparse`, `montecarlo`, and `maxentropy`) is available online (at `www.scipy.org`), and as examples and `pydoc` help strings distributed with the source code.

A.1 Code example

This appendix gives a small self-contained code example demonstrating how to use the SciPy `maxentropy` package. The code fits truncated Gaussian models (described in Section 2.8), in an arbitrary number D of dimensions. Recall that the desired

constraints in each dimension d are

$$\mathrm{E} f_{i,d}(X) = b_{i,d} \quad \text{for } i = 1, 2, 3,$$

for the feature statistics

$$f_{1,d}(X) = X_d \tag{A.1}$$
$$f_{2,d}(X) = X_d^2 \tag{A.2}$$
$$f_{3,d}(X) = 1\left\{a_l < X_d < a_u\right\}. \tag{A.3}$$

Python source code to fit models of maximum entropy subject to these constraints is as follows. The dimensionality d, bounds (a_l, a_u), and target moments $\{b_i\}$ are variables; the values given here yield one of the three models plotted in Figure 6.1, as specified by the variable `whichplot`.

```
from scipy import maxentropy, stats
from numpy import *

whichplot = 2           # sub-plot in Figure 6.1 (0, 1, or 2)
d = 1                   # number of dimensions
m = d*3                 # number of features

# Bounds
o = ones(d)
if whichplot == 0:
    lower = o * -2.5;  upper = -lower
elif whichplot == 1:
    lower = o * 0.5;  upper = o * 2.5
elif whichplot == 2:
    lower = o * -0.1;  upper = o * 0.1

def features(xs):
    """ xs should be an (m x n) matrix representing n
```

```
    observations xs[:,j] for j=0,...,n-1. """
    F = empty((m, xs.shape[1]), float)
    for k in xrange(len(xs)):
        F[3*k, :] = xs[k]
        F[3*k+1, :] = xs[k]**2
        F[3*k+2, :] = (lower[k] <= xs[k]) & (xs[k] <= upper[k])
    return F

# Target constraint values
b = empty(m, float)
if whichplot == 0:
    b[0:m:3] = 0        # expectation
    b[1:m:3] = 1        # second moment
    b[2:m:3] = 1        # truncate completely outside bounds
elif whichplot == 1:
    b[0:m:3] = 1.0      # expectation
    b[1:m:3] = 1.2      # second moment
    b[2:m:3] = 1        # truncate completely outside bounds
elif whichplot == 2:
    b[:] = [0., 0.0033, 1]

n = 10**5               # sample size

# Create a generator of features of random points under a
# Gaussian auxiliary dist q with diagonal covariance matrix
def sampleFgen():
    mu = b[0]
    sigma = (b[1] - mu**2)**0.5
    pdf = stats.norm(loc=mu, scale=sigma).pdf
    while True:
        xs = randn(d, n) * sigma + mu
        log_q_xs = log(pdf(xs)).sum(axis=0)
        F = features(xs)  # compute feature matrix from points
```

```
        yield F, log_q_xs, xs

q = sampleFgen()

model = maxentropy.bigmodel()    # create a model
model.setsampleFgen(q)           # use q as the auxiliary
model.fit(b)     # fit under the given constraints using SPO
```

After running this code, the object `model` has a vector of parameters $\theta = (\theta_i)_{i=1}^{3d}$ stored as the array `model.params`. The PDF of the fitted model can then be retrieved with the `pdf` method and plotted using `matplotlib` as follows:

```
# Plot the marginal pdf in dimension 0, letting x_d=0
# for all other dimensions d.
xs = arange(lower[0], upper[0], (upper[0]-lower[0]) / 1000.)
all_xs = zeros((d, len(xs)), float)
all_xs[0, :] = xs
pdf = model.pdf(features(all_xs))
import pylab
pylab.plot(xs, pdf)
pylab.show()
```

The output is shown in Figure 6.1.

B Sampling random sentences

Chapter 3 reviewed three Monte Carlo methods for estimating features over a complex distribution, all three of which require an initial sample drawn from an auxiliary distribution. This section considers how to sample artificial sentences efficiently from an n-gram auxiliary distribution.

An n-gram sampler is not the only possibility for generating random sentences, but it is simple and well-suited to this task. n-gram samplers have advantages as auxiliary distributions for the same reasons that they are the basis of much language-modelling research—they are simple in structure and quite effective. Rosenfeld et al. [2001] used a 3-gram auxiliary distribution for importance sampling to fit models of sentences. Their implementation reportedly required $O(v)$ time for generating each word, where v is the vocabulary size. This section describes two more efficient methods for sampling words from an n-gram model, one in $O(\log v)$ time, the other in constant time. Neither of these methods is new, but both are probably underutilized.

B.1 Naïve method and bisection

Consider sampling words from the following discrete distribution:

word	a	b	c	d
prob	.2245	.1271	.3452	.3032

One method is to assign the words unique indices, construct the cumulative PMF as

x	0	1	2	3
$F(x)$	.2245	.3516	.6968	1.000

and, given any uniform variate $u \in [0, 1)$, find the smallest value x for which $u < F(x)$. This can be done naïvely in linear time or more efficiently with a binary search in logarithmic time. The initial setup of the cumulative PMF requires linear time in the size of the sample space, but need be done only once before generating any number of words. The memory requirement for the table is linear in the vocabulary size.

It is worth reflecting on the severity of this memory requirement. Suppose a vocabulary size v of 10^5 words and that the probability of each word is stored as a 64-bit floating point value. A table of 1-gram (single-word) probabilities would require less than 1 MB of memory, clearly not a burden for modern computers, whereas a full complement of 2-grams (in which every pair of words w_1, w_2 in the vocabulary were represented) would require 80 GB of memory. In practice, though, the number of 2-grams that occur in a natural language corpus will be a small subset of all conceivable word pairs, and is likely to fit easily into memory. Table B.1 gave some figures for the FTW question corpus. For a corpus of this size, it is possible to store complete tables of the cumulative PMFs for 1-gram and 2-gram samplers for this task, and to generate random sentences from these samplers in logarithmic time $O(\log v)$.

Table B.1: The number of unique n-gram tokens in random samples of different sizes from the FTW question corpus, for $n = 1, 2, 3$. This affects the space-complexity of the auxiliary samplers

corpus size (# sentences)	1-grams	2-grams	3-grams
1000	2389	5004	5543
10000	12015	37391	48095
100000	50089	237951	379615

With larger corpora, however, this may not be possible with a 3-gram sampler, since the number of distinct 3-grams in the corpus can be significantly larger. If the cumulative PMF table were generated, instead, on demand, the linear-time setup cost would dominate the logarithmic search cost for the actual sampling. This is likely the source of the linear time cost reported by Rosenfeld et al. [2001] for their 3-gram

sampler. My implementation, which I use for the benchmarks in the next section, stores a cache of the cumulative PMF tables for the most frequent 3-grams to reduce the computational overhead for generating them. This cache is stored as a hash table of tuples of integers (representing the two-word contexts), whose values can be accessed in near-constant time. This makes the overall computational load sub-linear in the vocabulary size v; in the best case, when the corpus is small enough for a cache of all 3-grams, the time to sample each word has the same $O(\log v)$ bound as when sampling in 1-gram and 2-gram contexts.

The 3-gram samplers of this and the next section are smoothed with 2-gram and 1-gram counts under the simple linear interpolation scheme

$$p(w_3 \mid w_1 w_2) = \theta_1 p_{\text{tri}}(w_3 \mid w_1 w_2) + (1-\theta_1)\big[\theta_2 p_{\text{bi}}(w_3 \mid w_2) + (1-\theta_2) p_{\text{uni}}(w_3)\big] \quad \text{(B.1)}$$

where $\theta_1, \theta_2 \in (0, 1)$.

B.2 Compressed table lookup

A logarithmic-time sampler, as described above, is quite efficient: for a vocabulary of 10^5 words, the binary search requires no more than $\log_2 10^5 < 17$ comparisons. Despite this, even more efficient methods exist, with constant-time complexity. Marsaglia et al. [2004] describe three such methods; here we review the first of these, the compact table lookup method.

Paraphrasing Marsaglia et al. [2004], perhaps the fastest way to generate variates from the distribution in the previous section is to construct a lookup table T with 10000 entries consisting of 2245 a's, 1271 b's, 3452 c's and 3032 d's. Then we simulate a random integer i in $[0, \ldots, 9999]$, and choose $x = T[i]$ as an observation from our desired distribution. This is a constant-time operation.

The memory requirements would scale unreasonably with higher precision, but condensed versions of such a lookup table are also feasible. Consider instead expressing the probabilities as rationals with denominator 100; then we have the alternative

generating procedure

```
if(i<9800) return A[i/100];
return B[i-9800];
```

where A and B are the tables

```
A[98] ={22*a,12*b,34*c,30*d}
B[200]={45*a,71*b,52*c,32*d}
```

meaning that A has 98 elements with 22 a's, 12 b's, etc. This generating procedure is equivalent and almost as fast, but the two tables are more compact, using a total of 298 elements, versus 10000 for the original procedure.

Marsaglia et al. [2004] describe an implementation using five tables, rather than the two here, where probabilities are expressed to the nearest rationals with denominator 2^{30}, which is sufficient for events of probability greater than approximately 5×10^{-10}. The implementation also performs the integer division using bit shifts for efficiency. I reimplemented the compact table-lookup sampler (adapting Marsaglia et al.'s implementation) for comparison with the binary search method of the previous section. One difference is that I used the NumPy implementation of the Mersenne Twister [Matsumoto and Nishimura, 1998] instead of Marsaglia's XOR-shift random number generator [2003], since the XOR-shift generator, although faster, is less well tested and has been shown since its publication to have weaknesses [L'Ecuyer and Panneton, 2005]. My implementation, in C with Python bindings, is now available as the `montecarlo` module in SciPy.

The memory requirements and setup costs of this generation method, like those of the previous section, are untroublesome for sampling low-order n-grams, but can be expensive even for $n = 3$ because many such tables are required. My implementation of the 3-gram sampler (`TRI_HASH` in Table B.2) uses a similar cache of cumulative PMF tables to that described earlier for avoiding repeated computation.

Table B.2: Informal timing results for the different auxiliary samplers: number of words generated per second (best of 5 trials). The n-grams were derived from a subset of 10^4 sentences from the FTW question corpus [Schofield, 2003].

UNI_5TBL	BI_5TBL	TRI_5TBL	TRI_HASH
3.8×10^5	1.7×10^4	1.6×10^3	5.0×10^0

Speed comparison

An informal comparison of sampling speed with each of these implementations is given in Table B.2. This indicates that sampling either 1-gram or 2-gram sentences can be orders of magnitude faster than sampling 3-gram sentences. This is especially likely to be true for large vocabulary sizes v, since memory is likely to be sufficient for $O(v)$ sets of lookup tables (one for each word as a conditioning context), but not for the $O(v^2)$ potential 3-gram contexts. The possibility to keep the lookup tables for each context in memory permits 1-gram and 2-gram sampling in constant time, following a single initialization step to construct the tables.

Bibliography

S. P. Abney. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4): 597–618, 1997.

J. Aitchison. *The Articulate Mammal: An Introduction to Psycholinguistics*. Routledge, London, fourth edition, 1998.

Y. Altun and A. Smola. Unifying divergence minimization and statistical inference via convex duality. In *Submitted to the 19th Annual Conference on Learning Theory*, 2006.

F. A. Amaya and J.-M. Benedí. Using perfect sampling in parameter estimation of a whole sentence maximum entropy language model. In C. Cardie, W. Daelemans, C. Nédellec, and E. T. K. Sang, editors, *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000*, pages 79–82, Somerset, New Jersey, 2000. Association for Computational Linguistics. URL `citeseer.ist.psu.edu/article/amaya00using.html`.

F. A. Amaya and J.-M. Benedí. Improvement of a whole sentence maximum entropy language model using grammatical features. In *Meeting of the Association for Computational Linguistics*, pages 10–17, 2001. URL `http://citeseer.nj.nec.com/505752.html`.

S. Andradóttir. A scaled stochastic approximation algorithm. *Management Science*, 42(4):475–498, 1996.

S. Andradóttir. A review of simulation optimization techniques. In *Proceedings of the 1998 Winter Simulation Conference*, 1998a.

S. Andradóttir. Simulation optimization. In J. Banks, editor, *Handbook of Simulation*, pages 307–333. John Wiley & Sons, 1998b.

C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, Feb. 2003. URL `http://citeseer.ist.psu.edu/andrieu03introduction.html`.

A. N. Avramidis. Derivatives and credit risk: importance sampling for multimodal functions and application to pricing exotic options. In *Proceedings of the 34th Winter Simulation Conference*, pages 1493–1501. ACM, 2002. ISBN 0-7803-7615-3.

L. R. Bahl, F. Jelinek, and R.-L. Mercer. A maximum likelihood approach to continuous speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 308–319. Kaufmann, San Mateo, CA, 1982.

L. R. Bahl, F. Jelinek, and R.-L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

J. Baldridge, T. Morton, and G. Bierner. OpenNLP maxent package in Java, 2005. URL `http://maxent.sf.net`.

I. Baltcheva, S. Cristea, F. Vázquez-Abad, and C. De Prada. Simultaneous perturbation stochastic approximation for real-time optimization of model predictive control. In *Proceedings of the First Industrial Simulation Conference (ISC 2003)*, pages 533–537, June 2003.

W. Bangerth, P. Stoffa, M. Sen, H. Klie, and M. F. Wheeler. On optimization algorithms for the reservoir oil well placement problem. *Submitted to Computational Geosciences*, 2004.

A. H. Barnett and D. J. C. MacKay. Bayesian comparison of models for images. In J. Skilling and S. Sibisi, editors, *Maximum Entropy and Bayesian Methods, Cambridge 1994*, pages 239–248, Dordrecht, 1995. Kluwer.

V. E. Benes. *Mathematical Theory of Connecting Networks and Telephone Traffic*. Academic Press, New York, 1965.

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996. URL `http://citeseer.nj.nec.com/berger96maximum.html`.

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.

J. R. Blum. Approximation methods which converge with probability one. *Annals of Mathematical Statistics*, 25:382–386, 1954.

L. Boltzmann. Über einige Fragen der kinetischen Gastheorie. *Wiener Berichte*, 96: 891–918, 1887.

N. K. Boots and P. Shahabuddin. Statistical tools for simulation design and analysis: simulating ruin probabilities in insurance risk processes with subexponential claims. In *Proceedings of the 2001 Winter Simulation Conference*, pages 468–476, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7803-7309-X.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

P. Bratley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation*. Springer-Verlag, New York, 1983.

S. L. Bridle, M. P. Hobson, A. N. Lasenby, and R. Saunders. A maximum-entropy method for reconstructing the projected mass distribution of gravitational lenses. *Monthly Notices of the Royal Astronomical Society*, 299:895, 1998.

P. F. Brown, J. Cocke, S. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. URL `http://citeseer.ist.psu.edu/brown90statistical.html`.

P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992. URL `http://citeseer.nj.nec.com/brown90classbased.html`.

P. F. Brown, S. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 1993. URL `http://citeseer.ist.psu.edu/576330.html`.

T. Bub, W. Wahlster, and A. Waibel. Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 71–74. IEEE, 1997.

J. A. Bucklew. Conditional importance sampling. *IEEE Transactions on Information Theory*, 51(1), Jan. 2005.

N. N. Bugaenko, A. N. Gorban, and M. G. Sadovsky. Maximum entropy method in analysis of genetic text and measurement of its information content. *Open Systems & Information Dynamics*, 5(3):265–278, 1998. ISSN 1230-1612. doi: http://dx.doi.org/10.1023/A:1009637019316.

J. P. Burg. Maximum entropy spectral analysis. In *Proceedings of the 37th meeting of the Society of Exploration Geophysicists*, 1967.

J. P. Burg. *Maximum entropy spectral analysis*. PhD thesis, Stanford University, 1975.

B. S. Caffo, J. Booth, and A. C. Davison. Empirical sup rejection sampling. Technical report, Department of Statistics, University of Florida, Apr. 2001.

Y. Carson and A. Maria. Simulation optimization. In S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, 1997.

M. Carter and B. Van Brunt. *The Lebesgue–Stieltjes integral*. Springer-Verlag, New York, 2000. ISBN 0-387-95012-5.

E. Charniak, K. Knight, and K. Yamada. Syntax-based language models for machine translation. In *Proceedings of MT Summit IX*, New Orleans, 2003.

H. F. Chen. *Stochastic Approximation and its Applications*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

S. F. Chen and R. Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, 1999. URL `http://citeseer.nj.nec.com/chen99gaussian.html`.

M. Chiang and S. Boyd. Geometric programming duals of channel capacity and rate distortion. *IEEE Transactions on Information Theory*, 50(2), Mar. 2004.

N. Chomsky. Quine's empirical assumptions. In D. Davidson and J. Hintikka, editors, *Words and objections. Essays on the work of W. V. Quine*, pages 53–68. D. Reidel, Dordrecht, 1969.

E. K. P. Chong and S. H. Zak. *An Introduction to Optimization*. John Wiley & Sons, New York, NY, second edition, 2001.

D. Choquet, P. L'Ecuyer, and C. Léger. Bootstrap confidence intervals for ratios of expectations. *ACM Transactions on Modeling and Computer Simulation*, 9(4): 326–348, 1999.

T. A. Cohn. *Scaling Conditional Random Fields for Natural Language Processing*. PhD thesis, Faculty of Engineering, University of Melbourne, July 2006.

T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 1991.

F. Cozman and E. Krotkov. Truncated Gaussians as tolerance sets. Technical Report CMU-RI-TR-94-35, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 1994.

I. Csiszár. Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems. *Annals of Statistics*, 19(4):2032–2056, 1991.

I. Csiszár. Maxent, mathematics, and information theory. In K. M. Hanson and R. N. Silver, editors, *Maximum Entropy and Bayesian Methods*, pages 35–50. Kluwer Academic, 1996.

J. Darroch and D. Ratchliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

H. Daumé, K. Knight, I. Langkilde-Geary, D. Marcu, and K. Yamada. The importance of lexicalized syntax models for natural language generation tasks. In *Proceedings of the 2002 International Conference on Natural Language Generation (INLG)*, pages 9–16, Harriman, NY, July 2002.

M. Debbah and R. Müller. MIMO channel modelling and the principle of maximum entropy: An information theoretic point of view, 2003. URL `http://citeseer.ist.psu.edu/article/debbah04mimo.html`.

S. della Pietra, V. della Pietra, R. Mercer, and S. Roukos. Adaptive language modeling using minimum discriminant estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 633–636, Mar. 1992.

S. Della Pietra, V. J. Della Pietra, and J. D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997. URL `http://citeseer.nj.nec.com/dellapietra95inducing.html`.

B. Delyon and A. Juditsky. Accelerated stochastic approximation. Technical Re-

port 601, IRISA, Rennes, France, 1991. Also published in the SIAM Journal of Optimization, 1993.

T. J. DiCiccio and B. Efron. Bootstrap confidence intervals. *Statistical Science*, 3 (11):189–228, Aug. 1996.

W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley & Sons, New York, third edition, 1968.

R. A. Fisher. *Statistical Methods and Scientific Inference*. Oliver and Boyd, Edinburgh, second edition, 1959.

G. S. Fishman and V. G. Kulkarni. Improving Monte Carlo efficiency by increasing variance. *Management Science*, 38(10):1432–1444, 1992. ISSN 0025-1909.

G. Foster. Incorporating position information into a maximum entropy/minimum divergence translation model. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 37–42, Lisbon, Portugal, 2000.

M. Fu and J.-Q. Hu. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Kluwer Academic Publishers, Boston, 1997.

M. Gavaldà. *Growing Semantic Grammars*. PhD thesis, Carnegie Mellon University, 2000. URL `http://citeseer.ist.psu.edu/article/gavalda00growing.html`.

J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1340, 1989. URL `http://ideas.repec.org/a/ecm/emetrp/v57y1989i6p1317-39.html`.

C. J. Geyer and E. A. Thompson. Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society*, 54(3):657–699, 1992.

J. W. Gibbs. *Elementary Principles in Statistical Mechanics: Developed with Especial Reference to the Rational Foundation of Thermodynamics*. Charles Scribner's Sons, New York, 1902.

P. W. Glynn and W. Whitt. The asymptotic efficiency of simulation estimators. *Operations Research*, 40(3):505–520, 1992. ISSN 0030-364X.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 517–520, San Francisco, 1992. IEEE.

A. Golan and H. Gzyl. A generalized maxentropic inversion procedure for noisy data. *Applied Mathematics and Computation*, 127(2–3):249–260, Apr. 2002. ISSN 0096-3003. URL `http://www.elsevier.com/gej-ng/10/9/12/123/27/35/abstract.html`.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

A. Greenwald, B. Guillemette, V. Naroditskiy, and M. Tschantz. Scaling up the sample average approximation method for stochastic optimization with applications to trading agents. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI–05), Workshop on Trading Agent Design and Analysis*, Aug. 2005.

L. Grippo and S. Lucidi. A globally convergent version of the Polak–Ribière conjugate gradient method. *Mathematical Programming*, 78:375–391, 1997. URL `http://citeseer.csail.mit.edu/grippo95globally.html`.

S. F. Gull and G. Daniell. Image reconstruction from incomplete and noisy data. *Nature*, 272:686–690, 1978.

S. F. Gull and J. Skilling. Recent developments at Cambridge. In C. R. Smith and G. J. Erickson, editors, *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, pages 149–160, Dordrecht, 1983. Reidel.

G. Gürkan, A. Y. Özge, and S. M. Robinson. Sample-path optimization in simulation. Working Papers wp94070, International Institute for Applied Systems Analysis, July 1994. available at http://ideas.repec.org/p/wop/iasawp/wp94070.html.

P. Hall. *The Bootstrap and Edgeworth Expansion*. Springer-Verlag, New York, 1992.

J. M. Hammersley. Conditional Monte Carlo. *Journal of the ACM*, 3(2):73–76, Apr. 1956. ISSN 0004-5411.

J. M. Hammersley and D. C. Hanscomb. *Monte Carlo Methods*. Methuen, London, 1964.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modelling and Computer Simulation*, 5(1):43–85, 1995. ISSN 1049-3301.

T. Herges, A. Schug, H. Merlitz, and W. Wenzel. Stochastic optimization methods for structure prediction of biomolecular nanoscale systems. *Nanotechnology*, 14 (11):1161–1167, 2003. URL `http://stacks.iop.org/0957-4484/14/1161`.

G. T. Herman and A. Lent. Iterative reconstruction algorithms. *Computers in Biology and Medicine*, 6:273–294, 1976.

T. Hesterberg. *Advances in Importance Sampling*. PhD thesis, Department of Statistics, Stanford University, 1988.

S. D. Hill. Discrete stochastic approximation with application to resource allocation. *Johns Hopkins APL Technical Digest*, 26(1):15–21, 2005.

E. W. Hinrichs, S. Kübler, V. Kordoni, and F. H. Müller. Robust chunk parsing for spontaneous speech. In W. Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer-Verlag, Berlin, 2000.

A. Hopmans and J. P. C. Kleijnen. Importance sampling in systems simulation: a practical failure? *Mathematics and Computers in Simulation*, 21:209–220, 1979.

A. B. Huseby, M. Naustdal, and I. D. Vårli. System reliability evaluation using conditional Monte Carlo methods. Technical Report 2, Department of Mathematics, University of Oslo, Jan. 2004.

T. Jaakkola. Machine learning seminar notes: maximum entropy estimation. URL `http://people.csail.mit.edu/tommi/papers.html`. Published on the author's website, Mar. 1999.

T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37, Jan. 2000.

E. T. Jaynes. *Probability theory: the logic of science*. Cambridge University Press, 2003.

E. T. Jaynes. Information theory and statistical mechanics I. *Physical Review*, 106: 620–630, 1957.

E. T. Jaynes. Where do we stand on maximum entropy? In R. D. Levine and M. Tribus, editors, *The Maximum Entropy Formalism*, pages 15–118. The MIT Press, 1979.

E. T. Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70:939, 1982.

E. T. Jaynes. Spectrum and Chirp. In C. R. Smith and G. J. Erickson, editors, *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, pages 1–37, Dordrecht, 1983. Reidel.

E. T. Jaynes. Where do we go from here? In C. R. Smith and J. W. T. Grandy, editors, *Maximum-Entropy and Bayesian Methods in Inverse Problems*, pages 21–58. Kluwer Academic Publishers, 1985.

F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts, 1998.

M. Johnson, S. Geman, S. Canon, Z. Chi, and S. Riezler. Estimators for stochastic "unification-based" grammars. In *The Proceedings of the 37th Annual Conference of the Association for Computational Linguistics*, pages 535–541, College Park, Maryland, 1999.

L. K. Jones and C. L. Byrne. General entropy criteria for inverse problems, with applications to data compression, pattern classification, and cluster analysis. *IEEE Transactions on Information Theory*, 36(1):23–30, 1990.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice–Hall, 2000.

H. Kahn and A. W. Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operational Research Society of America*, 1:263–271, 1953.

S. Khudanpur. A method of ME estimation with relaxed constraints. In *John Hopkins University Language Modeling Workshop*, pages 1–17, 1995.

S. Khudanpur and J. Wu. Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. *Computer Speech and Language*, pages 355–372, 2000. URL `http://citeseer.nj.nec.com/308451.html`.

A. J. Kleywegt and A. Shapiro. Stochastic optimization. In G. Salvendy, editor, *Handbook of Industrial Engineering*, pages 2625–2649. John Wiley & Sons, third edition, 2001.

A. J. Kleywegt, A. Shapiro, and T. H. de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12

(2):479–502, Nov./Jan. 2001. ISSN 1052-6234 (print), 1095-7189 (electronic). URL `http://epubs.siam.org/sam-bin/dbq/article/36322`.

D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison–Wesley, Reading, third edition, 1997. ISBN 0–201–89684–2.

S. Kullback. *Information Theory and Statistics*. John Wiley & Sons, 1959.

H. J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.

S. S. Lavenberg and P. D. Welch. Using conditional expectation to reduce variance in discrete event simulation. In *Proceedings of the 1979 Winter Simulation Conference*, pages 291–294. IEEE Press, 1979.

Z. Le. Maximum entropy modeling toolkit for Python and C++, 2005. URL `http://www.nlplab.cn/zhangle/maxent_toolkit.html`.

P. L'Ecuyer. Efficiency improvement and variance reduction. In *Proceedings of the 1994 Winter Simulation Conference*, pages 122–132, San Diego, CA, USA, 1994. Society for Computer Simulation International. ISBN 0-7803-2109-X.

P. L'Ecuyer and F. Panneton. Fast random number generators based on linear recurrences modulo 2: overview and comparison. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, New York, NY, USA, 2005. ACM Press.

R. D. Levine and W. D. Meurers. Head-driven phrase structure grammar: Linguistic approach, formal foundations, and computational realization. In K. Brown, editor, *Encyclopedia of Language and Linguistics*. Elsevier, Oxford, second edition, 2006. ISBN 0-08-044299-4. URL `http://ling.osu.edu/~dm/papers/ell2-hpsg.html`.

B. H. Lindqvist and G. Taraldsen. Monte Carlo conditioning on a sufficient statistic. *Biometrika*, 92:451–464, 2005. URL `http://biomet.oxfordjournals.org/cgi/content/abstract/92/2/451`.

J. K. Lindsey. *Parametric Statistical Inference*. Clarendon Press, Oxford, 1996.

D. J. C. MacKay. Density networks and their application to protein modelling. In J. Skilling and S. Sibisi, editors, *Maximum Entropy and Bayesian Methods, Cambridge 1994*, pages 259–268, Dordrecht, 1996. Kluwer.

R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In D. Roth and A. van den Bosch, editors, *Proceedings of CoNLL-2002*, pages 49–55. Taipei, Taiwan, 2002.

R. Malouf, J. Baldridge, and M. Osborne. The toolkit for advanced discriminative modeling (TADM), June 2006. URL `http://tadm.sourceforge.net/`.

T. P. Mann. Numerically stable hidden markov model implementation. Online tutorial, February 2006. URL `http://www.phrap.org/compbio/mbt599/hmm_scaling_revised.pdf`.

C. Manning. Probabilistic syntax. In R. Bod, J. Hay, and S. Jannedy, editors, *Probabilistic Linguistics*, chapter 8. The MIT Press, Apr. 2003.

G. Marsaglia. Xorshift RNGs. *Journal of Statistical Software*, 8(14), 2003.

G. Marsaglia, W. W. Tsang, and J. Wang. Fast generation of discrete random variables. *Journal of Statistical Software*, 11(3), July 2004.

K. Marti. *Stochastic Optimization Methods*. Springer-Verlag, New York, 2005.

J. Maryak. Some guidelines for using iterate averaging in stochastic approximation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2287–2290, 1997.

M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, Jan. 1998. ISSN 1049-3301.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087–1091, 1953.

T. P. Minka. A comparison of numerical optimizers for logistic regression. URL `http://research.microsoft.com/~minka/papers/logreg`. Published on the author's website, Oct. 2003.

D. Montgomery. Entropies for continua: Fluids and magnetofluids. In J. Skilling and S. Sibisi, editors, *Maximum Entropy and Bayesian Methods*, pages 304–314, Dordrecht, 1996. Kluwer.

J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

W. I. Newman. Extension to the the maximum entropy method. *IEEE Transactions on Image Processing*, 23:89–93, Jan. 1977.

E. Novak and K. Ritter. The curse of dimension and a universal method for numerical integration. In G. Nürnberger, J. W. Schmidt, and G. Walz, editors, *Multivariate Approximation and Splines, ISNM*, pages 177–188. Birkhäuser, Basel, 1997. URL `citeseer.ist.psu.edu/431965.html`.

F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July 2002.

T. E. Oliphant. *A Guide to NumPy*. Trelgol Publishing, 2006. URL `http://www.tramy.us`.

M. Palmer and T. Finin. Workshop on the evaluation of natural language processing systems. *Computational Linguistics*, 16(3):175–181, 1990.

A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, New York, third edition, 1991.

J. B. Paris and A. Vencovská. A note on the inevitability of maximum entropy. *International Journal of Approximate Reasoning*, 4(3):183–223, 1990. ISSN 0888-613X.

D. B. Paul and J. M. Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 357–361. Morgan Kaufmann, 1992.

J. Peters and D. Klakow. Compact maximum entropy language models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 1999. URL `http://citeseer.nj.nec.com/peters99compact.html`.

A. Plakhov and P. Cruz. A stochastic approximation algorithm with multiplicative step size adaptation. *ArXiv Mathematics e-prints*, math/0503434, Mar. 2005. URL `http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:math/0503434`.

E. Polak and G. Ribière. Note sur la convergence de methods de directions conjugées. *Rev. Francaise Informat. Recherche Opérationelle*, 16:35–43, 1969.

W. B. Poland and R. D. Shachter. Mixtures of Gaussians and minimum relative entropy techniques for modeling continuous uncertainties. In *Uncertainty in Artificial Intelligence: Proceedings of the Ninth Conference (UAI-1993)*. Morgan Kaufmann Publishers, 1993.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30:838–855, 1992.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992. ISBN 0521437148.

J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1–2): 223–252, 1996. URL `citeseer.ist.psu.edu/propp96exact.html`.

J. G. Propp and D. B. Wilson. Coupling from the past: a user's guide. In D. Aldous and J. G. Propp, editors, *Microsurveys in Discrete Probability*, volume 41 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 181–192. American Mathematical Society, 1998.

A. Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1998. URL `http://citeseer.ist.psu.edu/ratnaparkhi98maximum.html`.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag, New York, 1998.

S. M. Robinson. Analysis of sample path optimization. *Mathematics of Operations Research*, 21:513–528, 1996.

R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1994. URL `http://citeseer.nj.nec.com/rosenfeld94adaptive.html`.

R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10(3):187–228, 1996. URL `http://citeseer.nj.nec.com/rosenfeld96maximum.html`.

R. Rosenfeld, S. F. Chen, and X. Zhu. Whole-sentence exponential language models: A vehicle for linguistic-statistical integration. *Computer Speech and Language*, 15 (1):55–73, Jan. 2001. URL `http://citeseer.nj.nec.com/448532.html`.

R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 1981. ISBN 0471089176.

R. Ruppert. Almost-sure approximation to the Robbins–Monro and Kiefer–Wolfowitz processes with dependent noise. *Annals of Probability*, 10:178–187, 1982.

R. Ruppert. A Newton–Raphson version of the multivariate Robbins–Monro procedure. *Annals of Statistics*, 13:236–245, 1985.

R. Ruppert. Handbook in sequential analysis. In B. K. Ghosh and P. K. Sen, editors, *Stochastic Approximation*, pages 503–529. Marcel Dekker, New York, 1991.

M. A. Saunders. Interior methods for optimization with application to maximum entropy problems. In *Proceedings of the Sandia CSRI Workshop on Solution Methods for Saddle Point Systems in Computational Mechanics*, Santa Fe, NM, Dec. 2003.

E. J. Schofield. Language models for questions. In *Proceedings of the European Association for Computational Linguistics (EACL)*, Budapest, Hungary, April 2003.

E. J. Schofield. Fast parameter estimation for joint maximum entropy language models. In *Proceedings of Interspeech / ICSLP*, Jeju, Korea, Oct. 2004.

E. J. Schofield and G. Kubin. On interfaces for mobile information retrieval. In F. Paternò, editor, *Proceedings of the 4th Intl. Symposium on Human Computer Interaction with Mobile Devices (MobileHCI)*, number 2411 in Lecture Notes in Computer Science, pages 383–387. Springer-Verlag, Sept. 2002.

E. J. Schofield and Z. Zheng. A speech interface for open-domain question-answering. In *Proceedings of the Association for Computational Linguistics*, Sapporo, Japan, July 2003.

N. N. Schraudolph and T. Graepel. Combining conjugate direction methods with stochastic approximation of gradients. In C. M. Bishop and B. J. Frey, edi-

tors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pages 7–13, Key West, Florida, Jan. 2003. Society for Artificial Intelligence and Statistics. ISBN 0-9727358-0-1.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. Continued in following volume.

C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

A. Shapiro. Monte Carlo simulation approach to stochastic programming. In *Proceedings of the 2001 Winter Simulation Conference*, pages 428–431, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7803-7309-X.

A. Shapiro and Y. Wardi. Convergence analysis of stochastic algorithms. *Mathematics of Operations Research*, 21:615–628, 1996a.

A. Shapiro and Y. Wardi. Convergence analysis of gradient descent stochastic algorithms. *Journal of Optimization Theory and Applications*, 91:439–454, 1996b.

S. M. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.

A. Shimony. The status of the principle of maximum entropy. *Synthese*, 63:35–53, 1985.

J. Shore and R. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, IT-26:26–37, 1980.

M. J. Siclari, W. Van Nostrand, and F. Austin. The design of transonic airfoil sections for an adaptive wing concept using a stochastic optimization method. In *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA)*, number 96-0329, Reno, NV, Jan. 1996.

V. Siivola, M. Kurimo, and K. Lagus. Large vocabulary statistical language modeling for continuous speech recognition in Finnish. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, Aalborg, Denmark, 2001.

B. Skyrms. Maximum entropy inference as a special case of conditionalization. *Synthese*, 63:55–74, 1985.

J. C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45(10):1839–1853, 2000.

J. C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, April 2003. ISBN 0471330523. URL `http://www.amazon.co.uk/exec/obidos/ASIN/0471330523/citeulike-21`.

J. C. Spall. Stochastic optimization. In J. E. Gentle, W. Haerdle, and Y. Mori, editors, *Handbook of Computational Statistics*, pages 169–198. Springer-Verlag, Berlin, 2004. Chapter II.6.

J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4), 1998.

R. Srinivasen. *Importance of Sampling: Applications in Communications and Detection by Srinivasen*. Springer-Verlag, July 2002. ISBN 3540435208.

J. Tan and G. L. Stüber. A MAP equivalent SOVA for non-binary Turbo Codes. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 2, pages 602–606, New Orleans, LA, USA, June 2000. ISBN 0-7803-6283-7. URL `http://users.ece.gatech.edu/~stuber/nsf2/pubs/c2.ps`.

L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22:1701–1728, 1994.

H. Trost, J. Matiasek, and M. Baroni. The language component of the FASTY text prediction system. *Applied Artificial Intelligence*, 19(8):743–781, 2005.

H. F. Trotter and J. W. Tukey. Conditional Monte Carlo for normal samples. In H. A. Meyer, editor, *Symposium on Monte Carlo Methods*, pages 64–79, New York, 1956. John Wiley & Sons.

N. Ueffing and H. Ney. Using pos information for statistical machine translation into morphologically rich languages. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 347–354, Morristown, NJ, USA, 2003. Association for Computational Linguistics. ISBN 1-333-56789-0.

J. Uffink. The constraint rule of the maximum entropy principle. *Studies in History and Philosophy of Modern Physics*, 27(1):47–79, 1996. URL `http://citeseer.ist.psu.edu/uffink96constraint.html`.

I. G. Varea, F. J. Och, H. Ney, and F. Casacuberta. Improving alignment quality in statistical machine translation using context-dependent maximum entropy models. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1051–1054, Taipei, Taiwan, 2002.

E. Veach. *Robust Monte Carlo methods for light transport simulation.* PhD thesis, Stanford Computer Graphics Laboratory, 1998.

B. Verweij, S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Comput. Optim. Appl.*, 24(2-3):289–333, 2003. ISSN 0926-6003.

G. C. Wake, T. K. Soboleva, and A. B. Pleasants. Evolution of a truncated Gaussian density. In *Proceedings of the 1998 New Zealand Mathematics Colloquium*, July 1998. URL `http://atlas-conferences.com/c/a/b/d/08.htm`.

H. Wallach. Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002. URL `citeseer.ist.psu.edu/wallach02efficient.html`.

D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. *Nature*, 418:838, 2002. URL `http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cs/0204030`.

Y. Wardi. Stochastic algorithms with Armijo stepsizes for minimization of functions. *Journal of Optimization Theory and Applications*, 64:399–417, 1990.

M. T. Wasan. *Stochastic Approximation*. Cambridge Press, 1969.

R. C. Whaley and A. Petitet. Minimizing development and maintenance costs in supporting persistently optimized BLAS. *Software: Practice and Experience*, 35(2):101–121, February 2005. `http://www.cs.utsa.edu/~whaley/papers/spercw04.ps`.

R. C. Whaley, A. Petitet, and J. J. Dongarra. Automated empirical optimization of software and the ATLAS project. *Parallel Computing*, 27(1–2):3–35, 2001. Also available as University of Tennessee LAPACK Working Note #147, UT-CS-00-448, 2000 (`www.netlib.org/lapack/lawns/lawn147.ps`).

P. M. Williams. Bayesian conditionalisation and the principle of maximum entropy. *British Journal for the Philosophy of Science*, 31:131–144, 1980.

D. Yan and H. Mukai. Optimization algorithm with probabilistic estimation. *Journal of Optimization Theory and Applications*, 79:345–371, 1993.

S. Zhai, M. Hunter, and B. A. Smith. Performance optimization of virtual keyboards. *Human-Computer Interaction*, 17, 2002.