Mean Field Variational Inference for Dirichlet Process Mixtures of Generalized Linear Models and Applications in Approximate Q-learning

Lawrence W. Manning

Advisor: Professor Warren B. Powell

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Operations Research and Financial Engineering

Princeton University

11 April 2011

I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

_____

Lawrence W. Manning

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Lawrence W. Manning

Abstract

This thesis extends the theory of Dirichlet process mixtures of generalized linear models (DP-GLMs) with the derivation of a mean field variational inference algorithm for approximate posterior inference, which provides a fast, deterministic, and parallelizable alternative to approaches based on Markov Chain Monte Carlo (MCMC). It then applies DP-GLMs to develop a computationally stable approximation to the Q-learning algorithm for moderate- to high-dimensional, continuous state-action spaces and explores several applications in energy storage.

Acknowledgements

This thesis, like its author, is the work of many hands. I would like to express tremendous gratitude to Professor Warren Powell for his consistent patience and guidance. He has consistently challenged and encouraged me to expand my intellectual horizons and has in many ways defined my Princeton education.

Likewise, I want to thank Lauren Hannah for her patience and many contributions to my understanding of Dirichlet processes; Ilya Ryzhov for his unique blend of humor and insights; and Professor David Blei for his role in deepening and formalizing my education in Bayesian statistics.

My fondest affection goes to my many dear friends who have made my Princeton experience so memorable, and with whom I look forward to many long years of friendship: Amy Osterman, Ben Oseroff, Cassie Taylor, Gerald van den Berg, Farah Naim, Julie Kheyfets, Kunal Nayyar, Manav Lalwani, and Rohit Gawande, to name a few.

I am particularly indebted to Cassie Taylor, for her continual love, support, and indulgence; to Julie Kheyfets, for her consistent friendship and continuous supply of caffeine and sarcasm; to Manav Lalwani, for his inspiring patience and constant supply of Indian food; to Rohit Gawande, for his immeasurable coolness; and to Gerald van den Berg, for his inescapable companionship through this whole Princeton adventure.

Finally, I want to thank my family: my Mom, Carol, for her unwavering love and encouragement; my Dad, Tony, for his thoughtful perspective on all aspects of life; and my brother, Sam, for his boundless enthusiasm. Without them, this thesis would have never existed.

In memory of Margarita Misako Whalen,

whose courage, perseverance, and love are my constant inspiration.

Contents

List of Figures

Chapter 1

Introduction

Over the last half century, the study of statistics, as well as the rest of mathematics, has undergone a paradigm shift. Whereas the work of preceding millennia focused on the discovery of analytical formulae, the advent of massive computational resources has drawn the focus instead to the study of algorithms. Mathematics can now explore areas of research and applications requiring computations for which no closed form exists, problems that would require lifetimes of calculation by hand to approximate.

This thesis, broadly speaking, addresses such previously-intractable algorithms for the optimal treatment of *information*, with particular emphasis on its assimilation and its subsequent integration into decision making under uncertainty. This twofold purpose corresponds to a naturally dichotomized structure. The first two chapters operate within the framework of theoretical Bayesian statistics and machine learning, developing and extending the theory of Dirichlet process mixtures of generalized linear models. DP-GLMs, as they are often abbreviated, provide an extremely flexible algorithmic approach to posterior inference and regression, and we investigate two modes of approximate inference for such models.

We then progress to a discussion of Q-learning, a seminal algorithm in the theory of reinforcement learning, and propose an algorithmic extension of Q-learning to

continuous state and action spaces based upon DP-GLM regression. We spend the latter half of the thesis developing the requisite theory and performing empirical experimentation with this composite algorithm, which we refer to as approximate Q-learning with DP-GLMs.

Chapter 2

Dirichlet Process mixtures of Generalized Linear Models

This chapter develops the theory of Dirichlet process mixtures of generalized linear models (DP-GLMs), a Bayesian nonparametric solution to the general regression problem:

$$Y \mid X \sim f(m(X)), \tag{2.1}$$

where the distribution of Y, the *response* variable, is dependent upon the p-dimension set of *covariates* X through a deterministic mean function $m(\cdot)$.

We briefly discuss the theory of *generalized linear models* (GLMs) — a broad and flexible class of parametric regression models — before moving on to a discussion of Dirichlet processes (DPs) and subsequently introducing the composite DP-GLM model. After presenting the underlying theory, we apply DP-GLM regression to several empirical examples in an attempt to demonstrate the intuition of the model.

2.1 Generalized Linear Models

We start by considering ordinary linear regression, the most commonly used form of regression in both Bayesian and classical statistics. In ordinary linear regression, the observation errors are assumed to be independent and identically distributed normal random variables with equal variance σ^2. In vector notation:

$$y \mid \beta, \sigma^2, X \sim N(X\beta, \sigma^2 I), \tag{2.2}$$

where I is the $n \times n$ identity matrix, X is the $n \times p$ design matrix of covariate observations, and y is the $n \times 1$ vector of response observations.

Assuming a noninformative prior distribution $p(\beta, \sigma^2) \propto \sigma^{-2}$, we get the following conditional posterior for β:

$$\beta \mid \sigma^2, X, y \sim N(\hat{\beta}, V_\beta \sigma^2), \tag{2.3}$$

where

$$\hat{\beta} = (X^\top X)^{-1} X^\top y, \tag{2.4}$$

$$V_\beta = (X^\top X)^{-1}. \tag{2.5}$$

The marginal posterior of σ^2 has a scaled inverse χ^2 distribution:

$$\sigma^2 \mid X, y \sim \text{Inv-}\chi^2(n-p, s^2), \tag{2.6}$$

where

$$s^2 = \frac{1}{n-k}(y - X\hat{\beta})^\top (y - X\hat{\beta}). \tag{2.7}$$

For the purposes of computing the expectation of y_{n+1}, given the value of x_{n+1}, the

tower principle of conditional expectations implies that we may use $\hat{\beta}$ in place of β in Equation (2.2), yielding the namesake linear relationship between X and the expectation $\mathbb{E}(y \mid X)$.

Assuming the noninformative prior, Bayesian linear regression is analogous to classical least squares regression, which estimates β and σ^2 with $\hat{\beta}$ and s^2, respectively. For alternative variants of Bayesian linear regression — including parametric models for unequal variances and estimation with conjugate priors — see Gelman et al. (2003).

Generalized linear models extend classical linear regression to a wider variety of response types and distributions. In particular, GLMs relax the assumptions of a linear relationship between X and $\mathbb{E}(y \mid X)$ and of normally distributed errors. GLMs are characterized by three components:

1. The *linear predictor*, $\eta = X\beta$;

2. The *link function* $g(\cdot)$, which is a one-to-one invertible function that maps the linear predictor to the mean of the response: $\mu = g^{-1}(\eta) = g^{-1}(X\beta)$;

3. An exponential family distribution specifying the random component of the response variable with mean $\mathbb{E}(y \mid X) = \mu$.

Distributions in the exponential family can be expressed as a function of their (possibly vector- or matrix-valued) *natural parameter* θ, sufficient statistic $t(x)$, and underlying measure $h(x)$:

$$p(x \mid \theta) = h(x) \exp\{\theta^\top t(x) - a(\theta)\}, \tag{2.8}$$

where

$$a(\theta) = \log \int h(x) \exp\{\theta^\top t(x)\} dx \tag{2.9}$$

is the cumulant (or log-partition) function. Two examples are presented below.

Example 2.1 (Univariate Normal Distribution)**.** We may rearrange the standard form of the Gaussian distribution as follows:

$$\begin{aligned} p(x \mid \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\} \\ &= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\log\sigma^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}x^2 - \frac{1}{2\sigma^2}\mu^2\right\}. \end{aligned} \tag{2.10}$$

Equation (2.10) implies the natural parameterization:

$$\begin{aligned} \theta &= \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right)^\top \\ t(x) &= (x, x^2)^\top \\ h(x) &= \frac{1}{\sqrt{2\pi}} \\ a(\theta) &= \frac{1}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\mu^2 = -\frac{1}{2}\log(-2\theta_2) + \frac{\theta_1^2}{4\theta_2}. \end{aligned}$$

Example 2.2 (Poisson Distribution)**.** The Poisson distribution with rate λ — the de facto standard distribution for modeling count data — is also a member of the exponential family with parameter $\theta = \log\lambda$, sufficient statistic $t(x) = x$, underlying measure $h(x) = \frac{1}{x!}$, and cumulant function $a(\theta) = \exp(\theta) = \lambda$:

$$\begin{aligned} p(x \mid \lambda) &= \frac{1}{x!}\lambda^x e^{-\lambda} \\ &= \frac{1}{x!}\exp\{x\log\lambda - \lambda\}. \end{aligned}$$

The flexibility and algebraic elegance of the exponential family in modeling various forms of response and covariate data (continuous, count, categorical, circular, etc.) make GLMs a diverse, versatile, and computationally tractable family of *parametric* regression models. However, GLMs are still restricted by their dependence upon their parameters — the linear coefficients, the dispersion parameters, etc. — which are situated in finite-dimensional space and therefore have a finite capacity for adaptation

to the data. A fitted GLM is assumed to apply uniformly across all possible values of the covariates, but real-world data often lacks such uniformity. Dirichlet process mixtures of generalized linear models, which are introduced in the following sections, fall into the broad class of *nonparametric* regression algorithms, which by design attempt to remedy this limitation.

2.2 Dirichlet Processes

Dirichlet process mixture models (DPMMs) are a class of Bayesian hierarchical models that represent the true density of a random variable as a mixture of parametric densities conditioned on the latent parameter θ, which is imbued with an uncertain prior distribution $\mathbb{G}$. The marginal probability of an observation x_i is given by the continuous mixture:

$$f(x_i) = \int_{\mathcal{T}} f(x_i \mid \theta)\mathbb{G}(d\theta),$$

where $\mathcal{T}$ is the set of all possible parameters. Uncertainty about the prior density $\mathbb{G}$ can be modeled with a *Dirichlet process*, which places a measure on probability measures (Ferguson, 1973). If $\mathbb{G}$ is drawn from a Dirichlet process with base measure $\mathbb{G}_0$ and positive scaling parameter α, then for any finite partition $A_1, \ldots, A_k$ of $\mathcal{T}$,

$$(\mathbb{G}(A_1), \ldots, \mathbb{G}(A_k)) \sim Dir(\alpha\mathbb{G}_0(A_1), \ldots, \alpha\mathbb{G}_0(A_k)),$$

where $Dir(a_1, \ldots, a_k)$ represents the Dirichlet distribution with strictly positive parameters $(a_1, \ldots, a_k)$, and $\mathbb{G}_0$ is a measure on $\mathcal{T}$.

A more intuitively appealing definition of the Dirichlet process is provided in the *stick-breaking construction* of Sethuraman (1994). Consider two infinite collections of random variables, $V_i \sim \text{Beta}(1, \alpha)$ and $\theta_i \sim \mathbb{G}_0$ for $i = \{1, 2, \ldots\}$. The stick-breaking

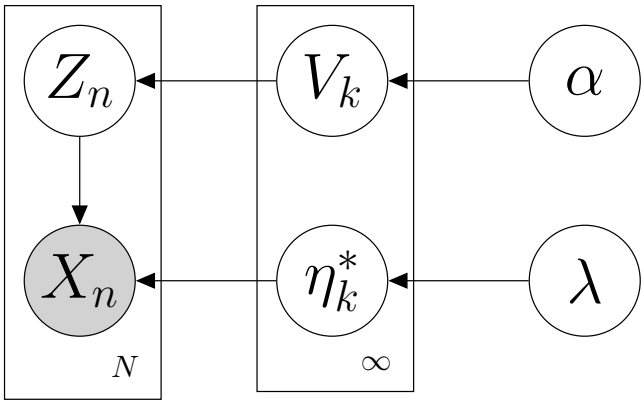

Figure 2.1: Graphical model representation of a Dirichlet process mixture under the stick-breaking construction. Nodes denote random variables, edges denote potential dependence, and plates denote replication.

process is represented by:

$$\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1} 1 - v_j, \tag{2.11}$$

$$\mathbb{G} = \sum_{i=1}^{\infty} \pi_i(\mathbf{v}) \delta_{\theta_i}. \tag{2.12}$$

We can think of the countably infinite vector of mixing proportions $\pi(\mathbf{v})$ as having arisen by repeatedly breaking a "stick" of length one according to the values of v_i, which will always be on the open interval $(0, 1)$. The size of each successive "piece" is determined by a random draw from a Beta$(1, \alpha)$ distribution, proportional to the length of the remainder of the "stick." Thus, the first piece is of length v_1, the second is of length $v_2 \times (1 - v_1)$, and so on. As described by Blei and Jordan (2006), we may think of the data in a Dirichlet process mixture model as having been generated by the following process:

1. Draw $V_i \mid \alpha \sim \text{Beta}(1, \alpha), \quad i = \{1, 2, \dots\}$.

2. Draw $\Theta_i \mid \mathbb{G}_0 \sim \mathbb{G}_0, \quad i = \{1, 2, \dots\}$.

3. For the nth data point:

   (a) Draw $Z_n \mid \{v_1, v_2, \dots\} \sim \text{Mult}(\pi(\mathbf{v}))$.

   (b) Draw $X_n \mid z_n, \{\theta_1, \theta_2, \dots\} \sim p(x_n \mid \theta_{z_n})$.

This stick-breaking construction is very important in the construction of a mean field variational inference algorithm for approximating DP-GLMs in Section 3.2.

An analytically convenient property of measures drawn from Dirichlet processes is that they may be integrated out of the conditional distribution of Θ_n given $\theta_{1:(n-1)}$. The random variable Θ_n has a Pólya's Urn distribution (Blackwell and MacQueen, 1973):

$$p(\Theta_n \mid \theta_{1:(n-1)}) \propto \alpha \mathbb{G}_0(\Theta_n) + \sum_{i=1}^{n-1} \delta_{\theta_i}. \tag{2.13}$$

The clustering effect of the Dirichlet process can be inferred from the conditional distribution of Θ_n, as there is a strictly positive probability that Θ_n will take on the value of a previous θ_i for $i = 1, \dots, n-1$. Otherwise, Θ_n will be drawn independently of $\theta_{1:(n-1)}$ from the base measure $\mathbb{G}_0$. The parameter α determines how likely Θ_n is to be newly drawn instead of assuming one of the previously observed values. Additionally, as n approaches infinity, the conditional distribution of Θ_n converges almost surely and in mean to a multinomial distribution over a vector with a finite number of elements. In turn, θ is a latent parameter to an observed datum x, such that

$$\begin{aligned} \mathbb{G} &\sim DP(\alpha, \mathbb{G}_0), \\ \theta_i &\sim \mathbb{G}, \\ x_i \mid \theta_i &\sim f(\cdot \mid \theta_i). \end{aligned} \tag{2.14}$$

Using Bayes' Theorem, we can derive the posterior distribution of $\theta_{1:n}$ given $x_{1:n}$, which renders apparent the interpretation as an infinite clustering model. Observations

are grouped by their common parameters, but the number of unique parameters is random and unknown, and newly examined data points can constitute observations of clusters that have not yet been observed(Antoniak, 1974; Hannah, 2010; Manning, 2010).

2.3 The DP-GLM Regression Model

A Dirichlet process mixture of generalized linear models, as its name suggests, utilizes a Dirichlet process to cluster data, fits a GLM within each cluster, and makes predictions for a datum x based on a probability-weighted sum of predictions from each of the observed clusters. More formally, DP-GLMs assume that: we may model the covariates X with some mixture of exponential family distributions; that we may model the response Y as a mixture of GLMs conditioned on the covariates X; and that with each component of the mixture we may associate a distinct set of GLM coefficients. For notational clarity, we partition $\theta = (\theta_x, \theta_y)$ and let $\mathbb{G}_0$ denote a base measure on the space of both X and Y; thus the full model is:

$$\begin{aligned} \mathbb{G} &\sim DP(\alpha \mathbb{G}_0), \\ (\theta_{x,i}, \theta_{y,i}) &\sim \mathbb{G}, \\ X_i \mid \theta_{x,i} &\sim f_x(\cdot \mid \theta_{x,i}), \qquad (2.15) \\ Y_i \mid x_i, \theta_{y,i} &\sim GLM(\cdot \mid x_i, \theta_{y,i}). \qquad (2.16) \end{aligned}$$

Note the similarities to Equation (2.14). The random measure $\mathbb{G}$ is generated from a Dirichlet process, and in turn θ is distributed according to $\mathbb{G}$. f_x is parameterized by θ_x and specifies the distribution of the covariates. The GLM for Y depends on the parameter θ_y and further varies by the form of the response (continuous, categorical, count, circular, etc.) and by the link function. Equations (2.15) and (2.16) are taken

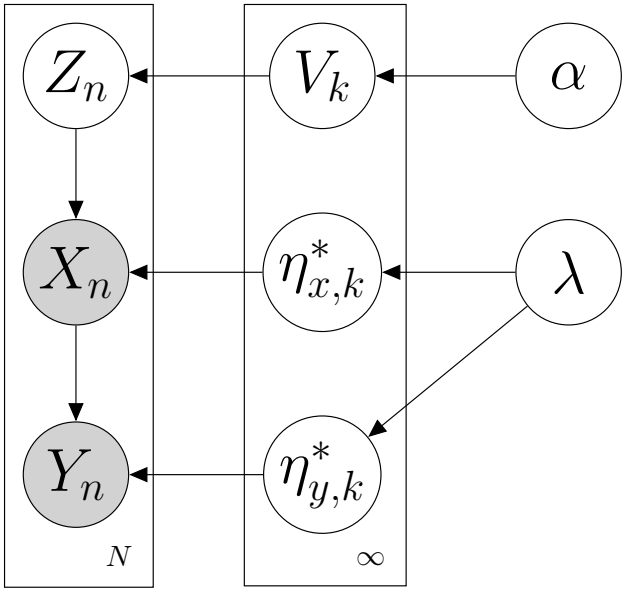

Figure 2.2: Graphical model representation of a Dirichlet process mixture of generalized linear models.

over $i = 1, \ldots, n$, though there generally will not be n distinct submodels due to the clustering effect of the Dirichlet process.

Given some set of covariate-response pairs $\{(x_i, y_i)\}_{i=1}^n$, one can use a DP-GLM to predict a response for a new datum x:

$$\mathbb{E}[Y \mid x, (X_i, Y_i)_{1:n}] = \mathbb{E}[Y \mid x, \theta_{1:n}]. \tag{2.17}$$

Hannah (2010) suggests approximating this generally intractable expression with:

$$\mathbb{E}[Y \mid x, \theta_{1:n}] \approx \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}[Y \mid x, \theta_{1:n}^{(m)}], \tag{2.18}$$

where $\theta_{1:n}^{(m)}$ represent a single sample realization of $\Theta_{1:n}$ and M is a sufficiently large integer. If the base measure $\mathbb{G}_0$ is conjugate, then the conditional posterior distribution of the parameters $\theta_{1:n}$ has a closed form, so the samples required by Equation (2.18) may be obtained by Gibbs sampling. If the base measure is non-conjugate, then we must

apply an appropriate variant of Markov Chain Monte Carlo (MCMC) that incorporates numerical integration in lieu of exact updates, such as Neal (2000)[Algorithm 8]. We survey the empirical application of such methods to DP-GLM regression in the following section.

2.4 Empirical Studies of DP-GLM with MCMC

In this section, we examine two data sets that each possess only a single, scalar covariate and a scalar response in order to facilitate easy visualization. In both cases, we take the covariate and the response to be continuous.

In the first data set, our response is a zero-noise quadratic function of the covariate. We use a normal-inverse-gamma prior for the parameters of the scalar covariate likelihood function. The normal-inverse-gamma distribution is a convenient choice, as it is conjugate to a normal likelihood with unknown mean and variance:

$$\mu_i \mid \sigma_i^2 \sim N(\mu_0, \sigma_i^2/\kappa_0).$$
$$\frac{1}{\sigma_i^2} \sim Gamma(\alpha, \beta).$$

For the response, we choose a conjugate linear base measure (i.e., linear regression). Though the absence of noise may make the data set seem trivial, such a data set allows us to better analyze the effects of the discrepancy between a linear base measure and a quadratic generative model. Figure 2.3 below shows the results.

The colored data points represent the training data; points with the same color are considered to be within the same "cluster" and are fit locally using a linear model. The solid black line represents the predictions for the hold-out data, which were generated in the same fashion as the training data and are represented by small black points. Though the fit appears to be continuous, it is not as smooth as the true response or as standard parametric models. This is an expected consequence of the nonparametric

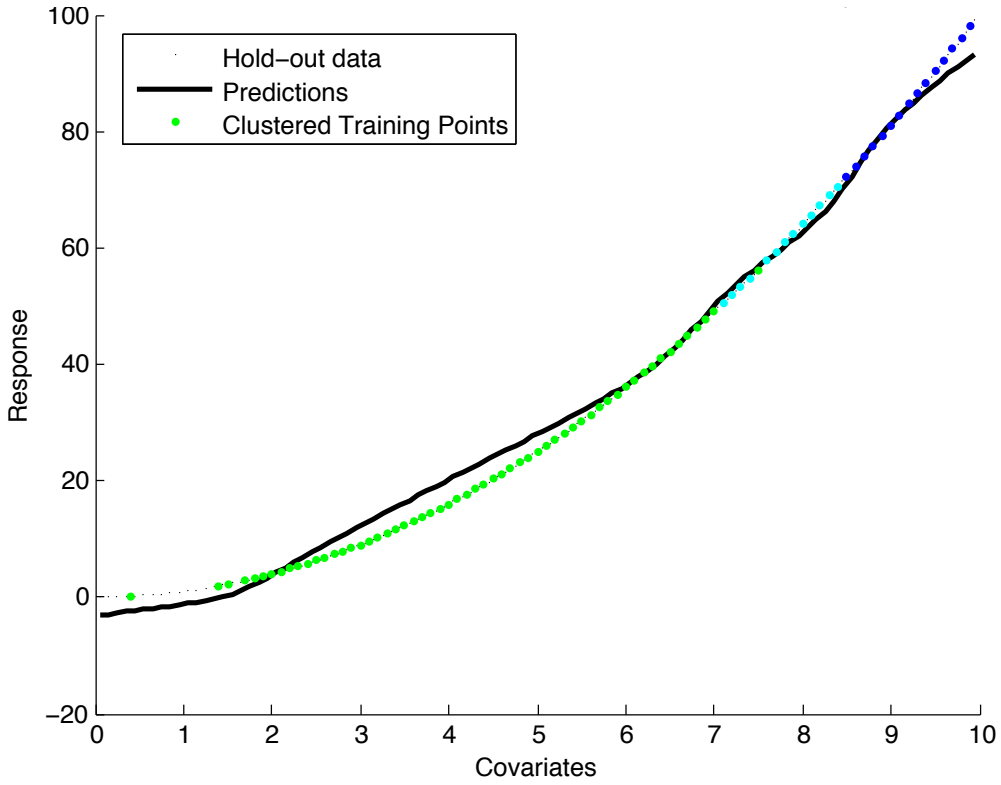

Figure 2.3: DP-GLM regression on a zero-noise quadratic function

nature of the DP-GLM model; rather than adhering to a previously defined structure, it conforms as closely as it can to the training data within the limitations of its base measure. In this example, the linear response constrains the ability of the DP-GLM to fit to a quadratic function, but the Dirichlet process simply generates more clusters in search of a better approximation. The final result is a reasonable if imperfect function approximation.

The second data set that we analyze is the cosmic microwave background (CMB) radiation data set from Hannah (2010), which exhibits marked heteroscedasticity (i.e., unequal variance) and is not well modeled by the conjugate prior. Instead, we utilize the non-conjugate prior suggested by Shahbaba and Neal (2009), which places a normal prior on the mean parameter and a lognormal prior on the variance parameter

of a normal likelihood:

$$\mu_i \sim N(\mu_0, \sigma_0^2)$$
$$\log \sigma_i^2 \sim N(M_\sigma, V_\sigma),$$

where μ_0, σ_0^2, M_σ, and V_σ are fixed hyperparameters. The loss of conjugacy requires an additional numerical integration within each sequential update of the Markov Chain. We follow the procedure of Neal (2000, Algorithm 8) mentioned in the preceding section in order to approximate this integral. For the response, a unique linear solution no longer exists, so we sample an approximate solution using the Hamiltonian Monte Carlo methods presented by Duane (1987) and Neal (2010).

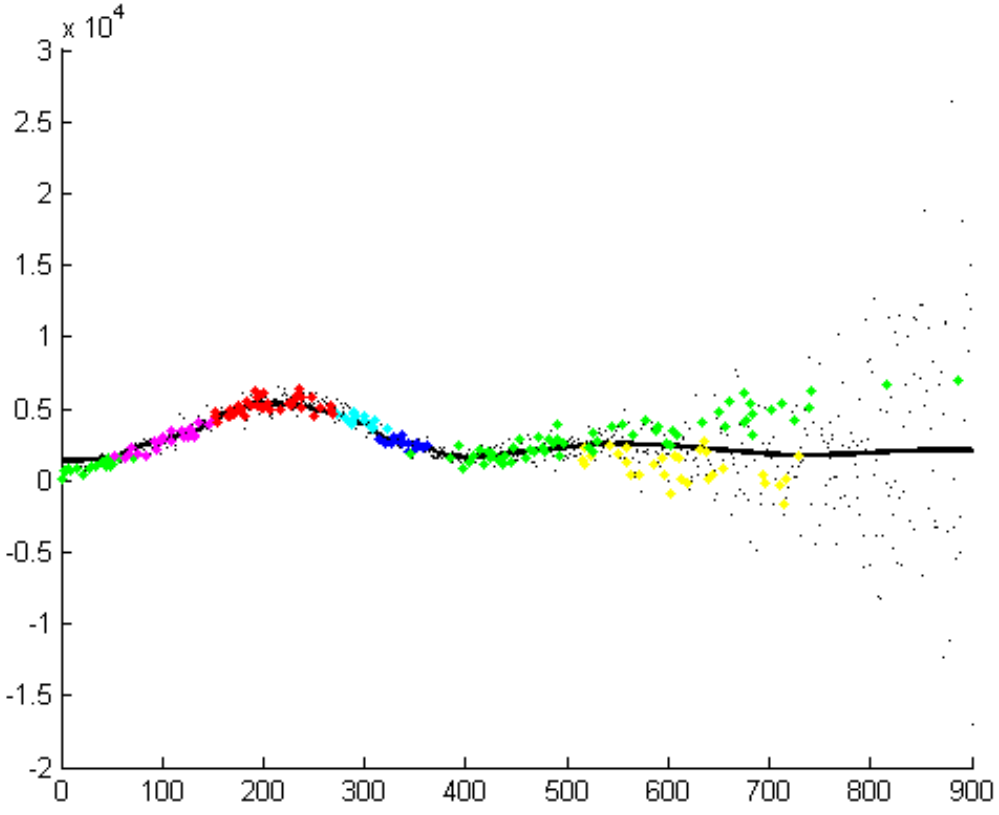

Figure 2.4: DP-GLM regression on heteroscedastic CMB data

As we can see from Figure 2.4, the DP-GLM algorithm can adjust quite well to

heteroscedastic data. It smoothly fits the low-variance segment of the data (roughly 0 to 400) with a sequence of localized linear models (again denoted by the multicolored training points) and then reverts to a smoothed moving average as the variance increases. The discriminatory power of the data for clustering also decreases as the variance increases, as evidenced by the decrease in the number of clusters from five in the low variance section to two in the high variance one.

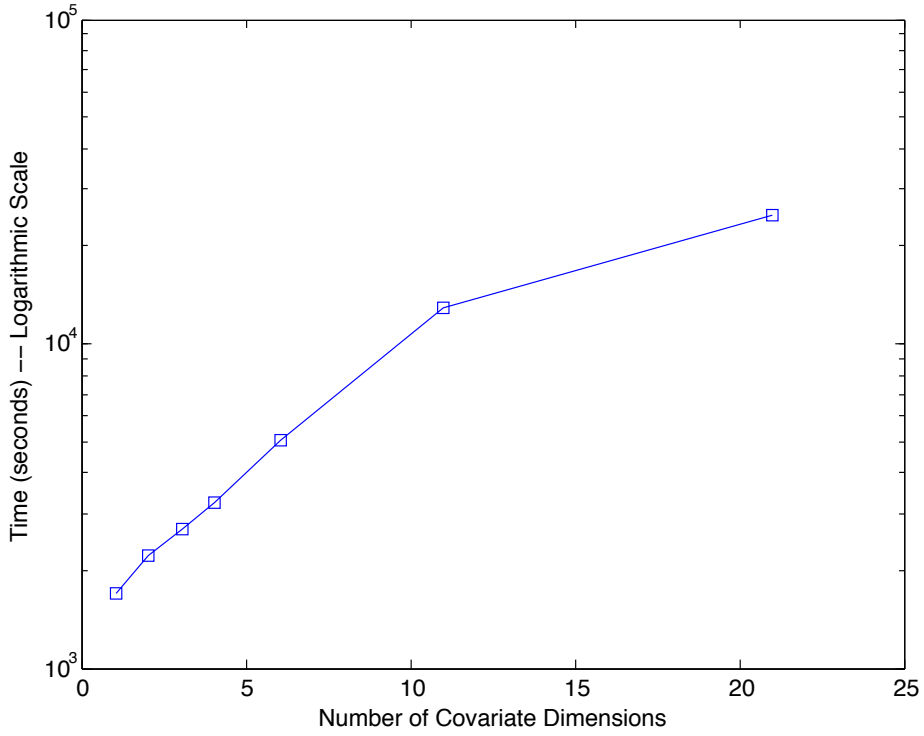

Figure 2.5: DP-GLM run times (logarithmic scale) plotted against the number of covariate dimensions on data sets of otherwise equal sizes.

Hannah (2010), Hannah et al. (2009), and Manning (2011) discuss DP-GLM regression based upon MCMC methods in greater depth. Though MCMC methods converge in the limit and can produce very good approximations in finite time, they often lack satisfactory convergence criteria, and their run times can be several orders of magnitude longer than those of parametric alternatives. The run times also increase dramatically as a function of the number of covariate dimensions, as

shown in Figure 2.5. While computational efficiency is not always of utmost priority, iterative applications such as those in Chapter 5 require repeated calculations for which MCMC methods are not well suited. The need for a "burn-in" period in order to eliminate bias from the initialization state precludes recursive calculation, and the non-deterministic behavior of the sampling procedure can introduce unwanted noise in finite time applications. The various shortfalls of MCMC methods provide the impetus for the next chapter, which seeks to develop a fast, deterministic alternative inference method for DP-GLMs.

Chapter 3

Mean Field Variational Inference for DP-GLMs

This chapter develops a variational algorithm for approximate posterior inference with Dirichlet process mixture of generalized linear models (DP-GLMs). Variational inference methods provide an approach to approximate posterior inference based on nonlinear *optimization* rather than *sampling* (Wainwright and Jordan, 2008). We begin with an introduction to variational inference and then extend the framework developed by Blei and Jordan (2006) for Dirichlet process mixture models of *conjugate-exponential* family distributions to encompass conjugate DP-GLMs. We then derive the necessary formulae for applying the algorithm to multivariate, continuous covariates and continuous responses.

Notably, we limit our discussion to conjugate-exponential models, in which the complete data likelihood function is a member of the exponential family and the prior distribution on the parameters is conjugate to the specified likelihood. As we see in Section 3.2, conjugacy allows us to obtain a closed-form mean field algorithm (Ghahramani and Beal, 2001). This is a consequence of the fact that, by definition, the posterior distribution has a closed functional form identical to that of the conjugate

prior. The only requirement for posterior inference is the calculation of the posterior parameters, which can be done with analytical — rather than numerical — methods. Interested readers may refer to Gelman et al. (2003) for a thorough, general treatment of conjugate Bayesian analysis.

Let us consider a model with a vector of hyperparameters θ, latent variables $\mathbf{W} = \{W_1, \ldots, W_M\}$, and observations $\mathbf{x} = \{x_1, \ldots, x_n\}$. The posterior distribution of the latent variables can be formulated as:

$$p(\mathbf{w} \mid \mathbf{x}, \theta) = \exp\{\log p(\mathbf{x}, \mathbf{w} \mid \theta) - \log p(\mathbf{x} \mid \theta)\}. \tag{3.1}$$

Exact inference is often infeasible due to the need to compute the marginal log-probability of the observations:

$$\log p(\mathbf{x} \mid \theta) = \log \int p(\mathbf{x}, \mathbf{w} \mid \theta) d\mathbf{w}. \tag{3.2}$$

Because the latent variables become interdependent when conditioned on the data, an analytic solution is usually not available, and, in general, naïve approximations have exponential computational complexity.

MCMC methods take advantage of the fact that Equation (3.2) is little more than a normalizing constant by using a ratio of joint probabilities to construct a Markov Chain whose ergodic distribution is the target posterior (Neal, 1993; Ross, 2006). The Gibbs sampler, a special case of the Metropolis-Hastings family of MCMC algorithms, simply samples each latent variable conditioned on the previously sampled values of the other latent variables:

$$p(w_i \mid \mathbf{w}_{-i}, \mathbf{x}, \theta) = \exp\{\log p(\mathbf{w}, \mathbf{x} \mid \theta) - \log p(\mathbf{w}_{-i}, \mathbf{x} \mid \theta)\}. \tag{3.3}$$

The Gibbs sampler assumes that the conditional distributions of the latent variables

have closed, analytic forms, which is a much less stringent requirement than finding a closed form for the marginal log-likelihood of the observations. It then uses these conditional distributions to perform a random walk through the posterior probability space.

The primary advantage of MCMC methods is that the simulated Markov chain converges to the desired posterior in the limit, which allows us to draw a sample that is independent of our starting state. Thus, we may pick any starting state that we desire, let it run forever, draw a sample, and repeat an infinite number of times in order to obtain the exact posterior distribution. In practice, of course, infinite run times are usually not acceptable, so we approximate the posterior with a finite number of samples that are drawn at sufficiently large intervals both from the beginning of the chain and from one another. However, evaluation of what constitutes "sufficiently large" is quite subjective. Theoretical bounds are often excessively conservative, forcing practitioners to assess the trade-off between faster run times and potential instability of predictions. Even worse, the number of iterations necessary for convergence in conjunction with the fact that Markov chains must inherently be run in series can make certain applications utterly infeasible.

Variational inference works by reformulating the computation of the posterior distribution as an optimization problem, perturbing that problem (usually by either adding or removing constraints), and finding solutions to the perturbed problem (Wainwright and Jordan, 2008). We deal specifically with *mean field* variational methods, which minimize the Kullback-Leibler (KL) divergence between the *variational distribution* and the true posterior distribution $p(\mathbf{w} \mid \mathbf{x}, \theta)$. Mean field variational methods provide a fast, deterministic, and stable alternative to MCMC-based

techniques. The KL divergence is given by:

$$\begin{aligned} KL(q_\nu(\mathbf{w}) \parallel p(\mathbf{w} \mid \mathbf{x}, \theta)) &= \int_{\mathbf{w}} q_\nu(\mathbf{w}) \log \frac{q_\nu(\mathbf{w})}{p(\mathbf{w} \mid \mathbf{x})} d\mathbf{w} \\ &= \mathbb{E}_q \left[\log \frac{q_\nu(\mathbf{w})}{p(\mathbf{w} \mid \mathbf{x})} \right] \\ &= \mathbb{E}_q[\log q_\nu(\mathbf{w})] - \mathbb{E}_q[\log p(\mathbf{w} \mid \mathbf{x})] \\ &= \mathbb{E}_q[\log q_\nu(\mathbf{w})] - \mathbb{E}_q[\log p(\mathbf{w}, \mathbf{x} \mid \theta)] + \log p(\mathbf{x} \mid \theta), \end{aligned} \tag{3.4}$$

where $\mathbb{E}_q$ denotes the expectation taken with respect to $q_\nu(\mathbf{w})$. We may think of the KL divergence as a distance metric for probability measures, though it is not a proper metric because $KL(q \parallel p)$ is not necessarily equal to $KL(p \parallel q)$ (Rudin, 1976). Figure 3.1 illustrates the notion of variational inference as a problem of finding the "closest" member of the variational family to the target posterior and of then substituting the computed variational distribution for posterior inference.

Note that the last term of Equation (3.4) is the precise marginal probability from Equation (3.2) that we are trying to circumvent. However, since it does not depend on the collection of variational parameters ν, it is simply a constant term and may thus be ignored for the purposes of optimization. With that in mind, we recast the minimization of Equation (3.4) as the maximization of the lower bound on the marginal log-probability:

$$\begin{aligned} \log p(\mathbf{x}) &= \log \int_{\mathbf{w}} p(\mathbf{x}, \mathbf{w}) d\mathbf{w} \\ &= \log \int_{\mathbf{w}} q_\nu(\mathbf{w}) \frac{p(\mathbf{x}, \mathbf{w})}{q_\nu(\mathbf{w})} d\mathbf{w} \\ &\geq \int_{\mathbf{w}} q_\nu(\mathbf{w}) \log \frac{p(\mathbf{x}, \mathbf{w})}{q_\nu(\mathbf{w})} d\mathbf{w} \qquad (3.5) \\ &= \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{w})] - \mathbb{E}_q[\log q_\nu(\mathbf{w})] \triangleq \mathcal{L}(\nu) \qquad (3.6) \end{aligned}$$

We obtain Equation (3.5) from the preceding line by applying Jensen's inequality,

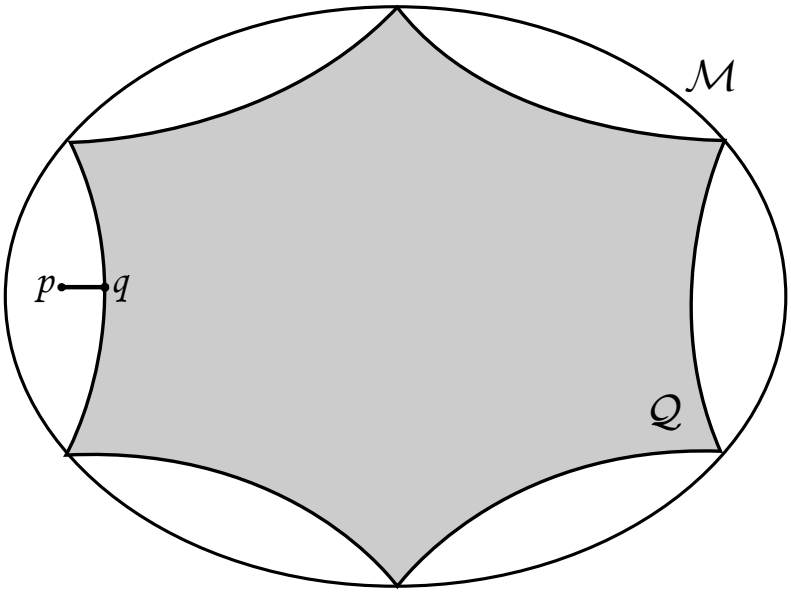

Figure 3.1: A graphical representation of variational inference. The outermost elliptical region represents $\mathcal{M}$, the set of all distributions over the latent variables, while the shaded internal region represents the subset $\mathcal{Q}$ of such distributions within the chosen variational family. Our goal is to find the variational distribution q closest to the target distribution p.

which in probabilistic form states that for any convex function $\varphi(x)$ and any random variable X, $\varphi(\mathbb{E}[X]) \geq \mathbb{E}[\varphi(X)]$ (Jensen, 1906; Rudin, 1986). The function $\mathcal{L}$ is often called the *Evidence Lower Bound*, or "ELBO" for short.

The difference between the true marginal log-probability and the ELBO is precisely the KL divergence between $q_\nu(\mathbf{w})$ and the true posterior. In order to achieve a computationally tractable approximation, we must choose a family of distributions $q_\nu(\mathbf{w})$ over which we can reasonably optimize Equation (3.6), which we do in the following section.

3.1 Mean field variational inference for exponential families

We now present an extended version of the derivation of Blei and Jordan (2006) for an iterative optimization method for variational inference in exponential families, which forms the basis of our subsequent derivation of a variational inference algorithm for DP-GLMs.

Let us assume that the conditional distribution of each latent variable is a member of the exponential family:

$$p(w_i \mid \mathbf{w}_{-i}, \mathbf{x}, \theta) = h(w_i) \exp\left\{g_i(\mathbf{w}_{-i}, \mathbf{x}, \theta)^\top w_i - a(g_i(\mathbf{w}_{-i}, \mathbf{x}, \theta))\right\}, \tag{3.7}$$

where $g_i(\mathbf{w}_{-i}, \mathbf{x}, \theta)$ is the natural parameter (analogous to θ in Equation (2.8)) for w_i conditioned on the remaining latent variables and the observations. Under the relatively flexible assumption given above, it is natural to utilize a fully factorized family as mean field variational approximations (Ghahramani and Beal, 2001):

$$q_\nu(\mathbf{w}) = \prod_{i=1}^{M} h(w_i) \exp\left\{\nu_i^\top w_i - a(\eta_i)\right\}, \tag{3.8}$$

where $\nu = (\nu_1, \ldots, \nu_M)$. With this parameterization, we may rewrite the bound in Equation (3.6) using the chain rule as:

$$\log p(\mathbf{x} \mid \theta) \geq \log p(\mathbf{x} \mid \theta) + \sum_{i=1}^{M} \left\{\mathbb{E}_q[\log p(W_i \mid \mathbf{x}, W_1, \ldots, W_{i-1}, \theta)] - \mathbb{E}_q[\log q_{\nu_i}(W_i)]\right\}. \tag{3.9}$$

If we wish to optimize with respect to ν_i, we may reorder $\mathbf{w}$ such that w_i is at the end. Doing so implies that the component of Equation (3.9) that depends on ν_i is given by:

$$\ell_i = \mathbb{E}_q[\log p(W_i \mid \mathbf{W}_i, \mathbf{x}, \theta)] - \mathbb{E}_q[\log q_{\nu_i}(W_i)]. \tag{3.10}$$

The variational distribution $q_{\nu_i}(W_i)$ is by definition a member of the exponential family, so this expression simplifies to:

$$\begin{aligned}\ell_i &= \mathbb{E}_q[\log p(W_i \mid \mathbf{W}_{-i}, \mathbf{x}, \theta)] - \mathbb{E}_q[\log h(W_i) + \nu_i^\top W_i - a(\nu_i)] \\ &= \mathbb{E}_q[\log p(W_i \mid \mathbf{W}_{-i}, \mathbf{x}, \theta)] - \mathbb{E}_q[\log h(W_i)] - \nu_i^\top a'(\nu_i) + a(\nu_i),\end{aligned}$$

using the fact that, for exponential family distributions, $E_q[W_i] = a'(\nu_i)$.
The derivative with respect to ν_i is:

$$\frac{\partial}{\partial \nu_i}\ell_i = \frac{\partial}{\partial \nu_i}(\mathbb{E}_q[\log p(W_i \mid \mathbf{W}_{-i}, \mathbf{x}, \theta)] - \mathbb{E}_q[\log h(W_i)]) - \nu_i^\top a''(\nu_i),$$

which indicates that the optimal ν_i satisfies:

$$\nu_i^* = [a''(\nu_i^*)]^{-1}\left(\frac{\partial}{\partial \nu_i}\mathbb{E}_q[\log p(W_i \mid \mathbf{W}_{-i}, \mathbf{x}, \theta)] - \frac{\partial}{\partial \nu_i}\mathbb{E}_q[\log h(W_i)]\right). \tag{3.11}$$

This result does not rely on the membership of $p(W_i \mid \mathbf{W}_i, \mathbf{x}, \theta)$ in the exponential family and is thus applicable to any well-specified distribution $p(\cdot)$. However, the fact that $p(\cdot)$ is a member of the exponential family allows us to obtain convenient, analytic expressions for the expected conditional log-probability and its derivative:

$$\begin{aligned}\mathbb{E}_q[\log p(W_i \mid \mathbf{W}_{-i}, \mathbf{x}, \theta)] &= \mathbb{E}_q[h(W_i)] + \mathbb{E}_q[g_i(\mathbf{W}_{-i}, \mathbf{x}, \theta)]^\top a'(\nu_i) - \mathbb{E}_q[a(g_i(\mathbf{W}_{-i}, \mathbf{x}, \theta))] \\ \frac{\partial}{\partial \nu_i}\mathbb{E}_q[\log p(W_i \mid \mathbf{W}_{-i}, \mathbf{x}, \theta)] &= \frac{\partial}{\partial \nu_i}\mathbb{E}_q[\log h(W_i)] + \mathbb{E}_q[g_i(\mathbf{W}_{-i}, \mathbf{x}, \theta)]^\top a''(\nu_i).\end{aligned}$$

Substituting those expressions into Equation (3.11) yields:

$$\nu_i = \mathbb{E}_q[g_i(\mathbf{W}_{-i}, \mathbf{x}, \theta)]. \tag{3.12}$$

Furthermore, given that our variational distribution is conjugate-exponential, we can

further reduce Equation (3.12) by considering the form of the general conjugate prior:

$$p(\theta \mid \lambda) = h(\theta) \exp\{\lambda_1^\top \theta + \lambda_2(-a(\theta)) - a_c(\lambda)\}, \tag{3.13}$$

where θ is the natural parameter of the exponential family likelihood in Equation (2.8) and $(\theta, -a(\theta))$ are its sufficient statistics. The posterior distribution, given such a prior and a set of observations $\{x_i\}_{i=1}^n$, can be calculated as:

$$\begin{aligned} p(\theta \mid \lambda, \{x_i\}_{i=1}^n) &\propto p(\theta \mid \lambda) \prod_{i=1}^n p(x_i \mid \theta) \\ &= h(\theta) h(x)^n \exp\left\{ \lambda_1^\top \theta + \lambda_2(-a(\theta)) - a_c(\lambda) + \theta^\top \sum_{i=1}^n t(x_i) - N a(\theta) \right\} \\ &\propto h(\theta) \exp\left\{ (\lambda_1 + \sum_{i=1}^n t(x_i))^\top \theta + (\lambda_2 + n)(-a(\eta)) \right\} \end{aligned} \tag{3.14}$$

Equation (3.14) demonstrates the extremely powerful analytic properties of the conjugate prior. Namely, for any conjugate prior with natural parameter λ, the posterior is of the same functional form as the prior with natural parameter $\hat{\lambda}$:

$$\begin{aligned} \hat{\lambda}_1 &= \lambda_1 + \sum_{i=1}^n t(x_i) \\ \hat{\lambda}_2 &= \lambda_2 + n. \end{aligned}$$

Thus, Equation (3.12) reduces to:

$$\nu_{i,1} = \mathbb{E}_q \left[\hat{\lambda}_1 \right] = \mathbb{E}_q[\lambda_1] + \sum_{i=1}^n \mathbb{E}_q[t(x_i)], \tag{3.15}$$

$$\nu_{i,2} = \mathbb{E}_q \left[\hat{\lambda}_2 \right] = \mathbb{E}_q \left[\lambda_2 \right] + \sum_{i=1}^n \mathbb{E}_q[\mathbf{1}\left[i \right]], \tag{3.16}$$

where $\mathbf{1}\left[i \right]$ is an indicator variable representing whether the datum x_i is a draw from the component of the generative process associated with the given prior distribution.

In the case of non-mixture models, this indicator is always equal to unity.

This formulation intuitively suggests an iterative optimization algorithm. In fact, repeatedly updating each parameter in sequence is equivalent to stepwise coordinate ascent in the KL divergence. Interestingly, this approach closely resembles the Gibbs sampler insofar as one iteratively computes a value for each parameter conditioned on the current values of the other parameters. In Gibbs sampling, the computed value is a random sample; in mean field variational inference, it is an expectation.

An even more striking analogy can be drawn between variational inference and the Expectation-Maximization (EM) algorithm of Dempster et al. (1977) and Wu (1983), a longtime staple of both maximum *a posteriori* (MAP) and maximum likelihood (MLE) estimation. We observe that the ELBO presented in Equation (3.6) is defined in precisely the same manner as the expected complete log-likelihood that is used as the objective function in EM. In fact, mean field variational inference may be viewed as a special, fully Bayesian application of EM in which the so-called "M-step" collapses into the similarly-named "E-step." Interested readers should refer to Gelman et al. (2003) for a summary of the EM algorithm and its applications in Bayesian inference.

3.2 Mean field variational inference for DP-GLMs

In this section, we extend the mean field variational algorithm developed in Blei and Jordan (2006) for Dirichlet process mixture models to DP-GLMs with conjugate base measures. We base the approximation upon a truncated version of the stick-breaking characterization of the DP given in Section 2.2. The latent variables under consideration are the scaling parameter of the Dirichlet process (on which we place a gamma prior), the "stick" sizes, the atomic distributions for the parameters of both the covariance and response likelihood functions (which are independent when conditioned on the data), and the cluster assignments, which we denote collectively

as $\mathbf{W} = (\alpha, \boldsymbol{V}, \boldsymbol{\eta}_{\boldsymbol{X}}^*, \boldsymbol{\eta}_{\boldsymbol{Y}}^*, \boldsymbol{Z})$. We assume knowledge of the necessary hyperparameters, which are merely the parameters of the conjugate prior for the scaling parameter, covariates, and response. We denote these hyperparameters by ω, λ_X, and λ_Y, respectively. We may write the ELBO defined in the previous section as:

$$\begin{aligned}\log p(\mathbf{x} \mid \lambda) \geq\ & \mathbb{E}_q[\log p(\alpha \mid \omega)] + \mathbb{E}_q[\log p(\boldsymbol{V} \mid \alpha)] \\ & + \mathbb{E}_q[\log p(\boldsymbol{\eta}_{\boldsymbol{X}}^* \mid \lambda_X)] + \mathbb{E}_q[\log p(\boldsymbol{\eta}_{\boldsymbol{Y}}^* \mid \lambda_Y)] \\ & + \sum_{n=1}^{N} \left(\mathbb{E}_q[\log p(Z_n \mid \boldsymbol{V})] + \mathbb{E}_q[\log p(x_n \mid Z_n)] + \mathbb{E}_q[\log p(y_n \mid Z_n]\right) \\ & - \mathbb{E}_q[\log q(\alpha, \boldsymbol{V}, \boldsymbol{\eta}_{\boldsymbol{X}}^*, \boldsymbol{\eta}_{\boldsymbol{Y}}^*, \boldsymbol{Z})]. \end{aligned} \tag{3.17}$$

We must define a variational distribution q that can accurately approximate the infinite dimensional random measure $\mathbb{G}$. In addition to assuming conjugacy of the atomic distributions, we approximate the DP itself by using a truncated version of our stick-breaking process. Specifically, we set $v_T = 1$ for some truncation level T so that the mixing proportions $\pi(t)$ are equal to zero for all $t > T$. The truncation level itself is a variational parameter that can be set arbitrarily, as it is not a part of the prior probability specification. Hence, we may run the algorithm to approximate convergence for a given T and can then increase or decrease T and compare the KL divergences.

For the purpose of computational tractability, we choose a fully factorized family of variational distributions:

$$q(\alpha, \boldsymbol{v}, \boldsymbol{\eta}_{\boldsymbol{X}}^*, \boldsymbol{\eta}_{\boldsymbol{Y}}^*, \boldsymbol{z}) = q_\psi(\alpha) \prod_{t=1}^{T-1} q_{\gamma_t}(v_t) \prod_{t=1}^{T} q_{\tau_t}(\eta_{t,X}^*) q_{\theta_t}(\eta_{t,Y}^*) \prod_{n=1}^{N} q_{\phi_n}(z_n), \tag{3.18}$$

where $q_\psi(\alpha)$ is a gamma distribution parameterized by (ψ_1, ψ_2); $q_{\gamma_t}(v_t)$ is a beta distribution parameterized by $(\gamma_{t,1}, \gamma_{t,2})$ for each $t = 1, \ldots, T-1$; $q_{\tau_t}(\eta_{t,X}^*)$ and $q_{\theta_t}(\eta_{t,X}^*)$ are conjugate-exponential family distributions for the covariates and the

response with natural parameters τ_t and θ_t, respectively; and q_{ϕ_n} is a multinomial distribution parameterized by $(\phi_{n,1}, \ldots, \phi_{n,T})$ for each $n = 1, \ldots, N$. Reverting to the notation of the preceding section, the full collection of variational parameters is:

$$\boldsymbol{\nu} = \{\alpha, \gamma_1, \ldots, \gamma_{T-1}, \tau_1, \ldots, \tau_T, \theta_1, \ldots, \theta_T, \phi_1, \ldots, \phi_N\}.$$

With this choice of q, the ELBO of Equation (3.17) can be explicitly computed using fairly standard exponential family computations, with the exception of the term $\mathbb{E}_q[\log p(Z_n \mid \boldsymbol{V})]$. We can rewrite this term using indicator variables:

$$\begin{aligned}\mathbb{E}_q[\log p(Z_n \mid \boldsymbol{V})] &= \mathbb{E}_q\left[\log\left(\prod_{i=1}^{\infty}(1-V_i)^{\mathbf{1}[Z_n>i]}V_i^{\mathbf{1}[Z_n=i]}\right)\right] \\ &= \sum_{i=1}^{\infty} q(z_n > i)\mathbb{E}_q[\log(1-V_i)] + q(z_n = i)\mathbb{E}_q[\log V_i].\end{aligned}$$

Under our truncated stick-breaking process, $\mathbb{E}_q[\log(1-V_T)] \triangleq 0$ and $q(z_n > T) \triangleq 0$. Thus,

$$\mathbb{E}_q[\log p(Z_n \mid \boldsymbol{V})] = \sum_{i=1}^{T} q(z_n > i)\mathbb{E}_q[\log(1-V_i)] + q(z_n = i)\mathbb{E}_q[\log V_i],$$

where

$$\begin{aligned}q(z_n = i) &= \phi_{n,i} \\ q(z_n > i) &= \sum_{j=i+1}^{T}\phi_{n,j} \\ \mathbb{E}_q[\log V_i] &= \Psi(\gamma_{i,1}) - \Psi(\gamma_{i,1}+\gamma_{i,2}) \\ \mathbb{E}_q[\log(1-V_i)] &= \Psi(\gamma_{i,2}) - \Psi(\gamma_{i,1}+\gamma_{i,2}),\end{aligned}$$

and where $\Psi(\cdot)$ denotes the digamma function. The digamma function is the first derivative of the natural logarithm of the gamma function and occurs here as a

consequence of taking the derivative of the log-normalizer of the appropriate beta distribution.

With this in mind, the results given in Equations (3.12), (3.15), and (3.16) can then be applied to yield an iterative mean field algorithm using the following sequential update equations:

$$\psi_1 = \omega_1 + T - 1, \tag{3.19}$$

$$\psi_2 = \omega_2 - \sum_{i=1}^{T-1} \mathbb{E}_q[\log(1 - V_i)], \tag{3.20}$$

$$\gamma_{t,1} = 1 + \sum_n \phi_{n,t}, \tag{3.21}$$

$$\gamma_{t,2} = \mathbb{E}_q[\alpha] + \sum_n \sum_{j=t+1}^{T} \phi_{n,j}, \tag{3.22}$$

$$\tau_{t,1} = \lambda_{X,1} + \sum_n \phi_{n,t} t(x_n), \tag{3.23}$$

$$\tau_{t,2} = \lambda_{X,2} + \sum_n \phi_{n,t}, \tag{3.24}$$

$$\theta_{t,1} = \lambda_{Y,1} + \sum_n \phi_{n,t} t(x_n, y_n), \tag{3.25}$$

$$\theta_{t,2} = \lambda_{Y,2} + \sum_n \phi_{n,t}, \tag{3.26}$$

$$\phi_{n,t} \propto \exp(S_{n,t}), \tag{3.27}$$

for $t \in \{1, \ldots, T\}$ and $n \in \{1, \ldots, N\}$, where:

$$\begin{aligned} S_{n,t} = \mathbb{E}_q[\log V_t] + \sum_{i=1}^{t-1} \mathbb{E}_q[\log(1 - V_i)] + \mathbb{E}_q[\eta^*_{X,t}] t(x_n) - E_q[a(\eta^*_{X,t})] \\ + \mathbb{E}_q[\eta^*_{Y,t}] t(x_n, y_n) - E_q[a(\eta^*_{Y,t})]. \end{aligned} \tag{3.28}$$

Let us recall that $t(\cdot)$ is a vector of sufficient statistics and that $a(\cdot)$ is a log-normalizing function specific to each member of the exponential family. In cases where there is no

possibility of ambiguity, we favor argument-based indexing for such functions in lieu of extensive subscripts both here and elsewhere in the text.

Repeatedly updating the variational parameters according to Equations (3.19) through (3.27) yields a local maximum of the ELBO by Proposition 2.7.1 of Bertsekas (1999) because it is strictly convex (Blei and Jordan, 2006; Wainwright and Jordan, 2008; Xing et al., 2003).

The majority of the requisite calculations are standard for the exponential family and are trivial to implement. The overwhelming advantage of conjugacy in the base measure is visible in the closed form of the updating equations for the parameters of the atomic distributions, τ_i and θ_i. In particular, updates for the covariate parameters τ_i are easily derived from the natural exponential form of the prior distribution, which is thoroughly discussed in Bayesian statistical literature. Updates of the response parameters θ_i are more complex and are the topic of the following section.

3.3 Conjugate-Exponential Linear Regression

As we have emphasized multiple times throughout this chapter, conjugacy is a necessary condition for finding analytic versions of Equations (3.19) through (3.27). Conjugate priors have been widely studied for most standard exponential family distributions, and Bayesian linear regression itself can be formulated with a conjugate prior. This section derives that conjugate prior in natural exponential form in order to obtain the explicit updating equations for θ for linear conditional distributions for the response.

In modeling the response, we assume the homoscedastic, multivariate normal likelihood of Equation (2.2):

$$y \mid \beta, \sigma^2, X \sim N(X\beta, \sigma^2 I).$$

In contrast to Equation (2.3), we assume an informative prior on β and σ^2 for

generality:

$$\beta \sim N(\hat{\beta}_0, \Sigma_\beta)$$
$$\sigma^2 \sim \text{Inv-}\chi^2(n_0, \sigma_0^2).$$

Under this prior, given the $n \times p$ matrix X of covariate observations and the $n \times 1$ vector y of their associated responses, the conditional posterior distribution is:

$$\beta \mid \sigma^2, X, y \sim N(\hat{\beta}, V_\beta \sigma^2), \tag{3.29}$$

where:

$$\hat{\beta} = (\Sigma_\beta^{-1} + X^\top X)^{-1}(\Sigma_\beta^{-1}\hat{\beta}_0 + X^\top y), \tag{3.30}$$
$$V_\beta = (\Sigma_\beta^{-1} + X^\top X)^{-1}. \tag{3.31}$$

This conditional posterior distribution can be rearranged as:

$$\begin{aligned} f(\beta \mid \hat{\beta}, V_\beta) &= (2\pi)^{-p/2}|V_\beta|^{-1/2} \exp\left\{-\frac{1}{2}(\beta - \hat{\beta})^\top V_\beta^{-1}(\beta - \hat{\beta})\right\} \\ &= (2\pi)^{-p/2} \exp\left\{-\frac{1}{2}\log|V_\beta| - \frac{1}{2}\hat{\beta}^\top V_\beta^{-1}\hat{\beta} + \hat{\beta}^\top V_\beta^{-1}\beta - \frac{1}{2}\beta^\top V_\beta^{-1}\beta\right\}, \end{aligned}$$

which is clearly expressible as a member of the multivariate natural exponential family:

$$f(\beta \mid \hat{\lambda}_1, \hat{\lambda}_2) = h(\beta) \exp\left\{\hat{\lambda}_1^\top \beta + \text{tr}(\hat{\lambda}_2 \beta\beta^\top) - a(\hat{\lambda})\right\}, \tag{3.32}$$

where:

$$\begin{aligned} h(\beta) &= (2\pi)^{-p/2} \\ \hat{\lambda}_1 &= \Sigma_\beta^{-1}\hat{\beta}_0 + X^\top y \\ \hat{\lambda}_2 &= -\frac{1}{2}V_\beta^{-1} \\ &= -\frac{1}{2}\left(\Sigma_\beta^{-1} + X^\top X\right) \\ a(\hat{\lambda}) &= \frac{1}{2}\log|-2\hat{\lambda}^2| + \frac{1}{4}\operatorname{tr}(\lambda_2^{-1}\lambda_1\lambda_1^\top). \end{aligned}$$

By cross-referencing this posterior with the standard prior parameterization, we can work backwards to determine the natural exponential parameters of the prior and the sufficient statistics:

$$\begin{aligned} \lambda_1 &= \Sigma_\beta^{-1}\hat{\beta}_0 \\ \lambda_2 &= -\frac{1}{2}\Sigma_\beta^{-1} \\ t_{\beta,1}(X,y) &= X^\top y \\ t_{\beta,2}(X,y) &= -\frac{1}{2}X^\top X. \end{aligned}$$

For practical regression purposes, $\hat{\beta}$, the mean vector of regression parameters, is the most important component. We can write $\hat{\beta}$ in terms of the natural parameters:

$$\hat{\beta} = \hat{\lambda}_2^{-1}\hat{\lambda}_1.$$

The posterior distribution of the dispersion parameter σ^2 is easily obtained through conjugacy:

$$\sigma^2 \mid X, y \sim \text{Inv-}\chi^2\left(n_0 + n, \frac{n_0\sigma_0^2 + ns^2}{n_0 + n}\right), \tag{3.33}$$

where $ns^2 = (y - X\hat{\beta})^\top(y - X\hat{\beta})$. We state without proof that the prior natural parameter of σ^2 is $\eta = (n_0\sigma_0^2, n_0)^\top$ and that the sufficient statistic is $t_{\sigma^2}(X, y) =$

$(ns^2, n)^\top$.

For the purposes of mean field variational inference, we now only need the expectations of the sufficient statistics $t_1(\cdot,\cdot)$ and $t_2(\cdot,\cdot)$. Noting that the conditional sufficient statistics are, in fact, independent of σ^2 and applying the tower rule of conditional expectations, we can conclude that the expectations of the conditional statistics are equivalent to those of their unconditional counterparts. Since σ^2 is independent of β by definition (with dependency only on $\hat{\beta}$, X, and y), we can effectively treat β and σ^2 as two separate variational parameters.

Thus, for any $t \in \{1, \ldots, T\}$, we let Q_t be an $n \times n$ diagonal matrix with $Q_{ii}^{-1} = \phi_{i,t}$. By recognizing that the expectation of a matrix is equivalent to the matrix of the expectations of its elements and rewriting the matrix products $X^\top X$ and $X^\top y$ as two collections of sums, we see that:

$$\mathbb{E}_q[t_{\beta,1}(X,y)] = X^\top Q^{-1} y, \tag{3.34}$$

$$\mathbb{E}_q[t_{\beta,2}(X,y)] = X^\top Q^{-1} X, \tag{3.35}$$

$$\mathbb{E}_q[t_{\sigma^2,1}(X,y)] = \mathbb{E}_q[(y - X\hat{\beta})^\top (y - X\hat{\beta})] \tag{3.36}$$

$$= y' * Q^{-1} * y + \mathbb{E}_q[\hat{\lambda}_1]^\top \mathbb{E}_q[\hat{\lambda}_2]^{-1} \mathbb{E}_q[\hat{\lambda}_2], \tag{3.37}$$

where:

$$\mathbb{E}_q[\hat{\lambda}_1] = \lambda_1 + X^\top Q^{-1} y, \tag{3.38}$$

$$\mathbb{E}_q[\hat{\lambda}_2] = \lambda_2 - \frac{1}{2} X^\top Q^{-1} X. \tag{3.39}$$

The expectations given in Equations (3.37) through (3.39) are precisely the values of $\theta_{t,1}$ needed to complete our variational inference algorithm.

It is interesting to note that the variational update for responses modeled with Bayesian linear regression reduces to weighted linear regression with Q^{-1} as the

diagonal matrix of weights. Thus, in the linear case, we can reinterpret our regression model from Equation (2.2) as:

$$y \mid \beta, \sigma^2, X \sim N(X\beta, Q\sigma^2).$$

An analogous closed-form solution is postulated to exist for Poisson regression, though most other common forms of regression are unlikely to be expressible in conjugate-exponential form. The Delta method of Braun and McAuliffe (2010) presents a promising approach for extending variational inference to the full class of GLM responses and constitutes an exciting area for future research.

Chapter 4

Q-learning

The goal of this chapter is to provide the theoretical and algorithmic framework within which the remainder of this thesis will operate. We begin with an overview of the dynamic programming paradigm, then develope and discuss the Q-learning algorithm of Watkins and Dayan (1992). The next chapter then applies the results of Chapter 3 to extend the fundamentals presented here.

Broadly speaking, we are interested in optimal decision-making within stochastic systems. The application domain is immaterial; problems with decision-making under uncertainty may be found in virtually every imaginable discipline.

We conceptualize a system at time t in a given *state* in which we must take an *action*, which will influence the state at time $t + 1$. We assume that the states have some sort of natural ordering that may be interpreted as values or *contributions*; that is, some states are more desirable to visit than others. Our goal is to find a *policy* that can be used for making *decisions* (i.e., choosing actions) that maximize the discounted contributions that we accrue. Intuitively, a good policy will enable us to visit more desirable states with greater frequency and less desirable states accordingly less often. Powell (2011, Chapters 5 & 6) gives an elegant and comprehensive framework for modeling such problems, which we employ throughout the remainder of the text.

4.1 Dynamic Programming & Bellman's Equation

We begin by considering the very broad class of problems that may be formulated as *Markov decision process* (MDPs). In classical Markov decision processes, the states and actions are discrete, and the sets of all possible states and actions–$\mathcal{S}$ and $\mathcal{A}$, respectively–have finite cardinalities. Thus every possible state-action pair can be enumerated. We denote the state at time t as $S_t \in \mathcal{S}$ and the associated action as $a_t \in \mathcal{A}$. The transitions from state i to state j are presumed to be governed by a time-invariant transition probability matrix P of size $|\mathcal{S}| \times |\mathcal{S}| \times |\mathcal{A}|$, where:

$$P(S_{t+1} = j \mid S_t = i, a_t = k) = P_{ij}(k),$$

where we use the functional representation $P_{ij}(k)$ in place of P_{ijk} for notational clarity.

The most direct way of optimizing such an MDP is a technique referred to as *backward dynamic programming*. Backward dynamic programming calculates the value of being in state $s \in \mathcal{S}$ under policy π with the Bellman equation:

$$V^{\pi}(s) = \mathbb{E}[c(s, \pi(s))] + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}(\pi(s)) V^{\pi}(s'), \tag{4.1}$$

$$= \mathbb{E}[c(s, \pi(s))] + \gamma \mathbb{E}[V^{\pi}(s') \mid s, \pi(s)], \tag{4.2}$$

where c denotes the random reward received for being in s and taking action $\pi(s)$, s' denotes the subsequent state, $\gamma \in [0, 1)$ is the discount factor, and P is the transition probability matrix. If our MDP has a finite time horizon, which we term T, then we simply set $V^{\pi}(S_{T+1}) = 0$ for all possible values of S_{T+1} and all policies π. We can then repeatedly apply Bellman's equation to calculate the exact values of all possible states at times T, $T-1$, $T-2$, $\ldots$, 0. If our MDP has an infinite time horizon ($T = \infty$), then we may approximate this solution to arbitrary precision by solving the equivalent finite-time MDP with a sufficiently large horizon.

Assuming, among other things, that our MDP is stationary, then there is a stationary optimal policy denoted by π^* that satisfies:

$$V^{\pi^*}(s) = V^*(x) = \max_{a \in \mathcal{A}} \left(\mathbb{E}[c(s,a)] + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}(a) V^*(s') \right). \tag{4.3}$$

However, obtaining an exact solution with backward dynamic programming requires sufficient computational power to perform a calculation of $\mathcal{O}(|\mathcal{S}||\mathcal{A}|)$ complexity to valuate a single state. Doing so for every state is prohibitively expensive for all but the simplest problems. This is particularly true of vector-valued state spaces, in which the number of states grows exponentially with the number of dimensions. Consider, for example, a game of chess, in which the state is given by the position of each of the 32 pieces on the 64-square board. Though the initial state is fixed, the long-term number of potential states is $\frac{64!}{32!} \approx 10^{53}$. The number of possible actions is comparably enormous. This dilemma is generally known as the *curse of dimensionality* and has given rise to the fields of approximate dynamic programming (ADP) and reinforcement learning (RL), which attempt to circumvent such intractable calculations with a wide variety of techniques. Readers are again referred to Powell (2011), as well as to Bertsekas and Tsitsiklis (1996) and Sutton and Barto (1998) for thorough treatments of the subject.

In addition to issues of computational complexity, the transition matrix is often unknown or impossible to determine in real-world problems. We begin by addressing this latter problem with the Q-learning algorithm of Watkins and Dayan (1992), which is introduced in the following section. We return to a discussion of the curse of dimensionality in Section 4.4, which motivates the results of Chapter 5.

4.2 Q-learning: Model-Free Valuation of State-Action Pairs

Q-learning is one of the oldest algorithms in the RL literature and is so named for its use of the variable $Q(s, a)$ to denote the value of being in state s and taking action a. Specifically, we consider $\bar{Q}^n(s, a)$, which is our statistical estimate of the so-called "Q-factor" after n iterations. Further, if we assume that we receive some (potentially random) contribution $\hat{C}(S^n, a^n)$ from taking action a^n while in state s^n, then we can consider the following as an "observation" of $Q(s, a)$:

$$\hat{q}^n = \hat{C}(S^n, a^n) + \gamma \max_{a' \in \mathcal{A}} \bar{Q}(S^{n+1}, a'). \tag{4.4}$$

We then use this $\hat{q}$ to update our statistical estimate of the appropriate Q-factor:

$$\bar{Q}^n(S^n, a^n) = \alpha_{n-1}\hat{q}^n + (1 - \alpha_{n-1})\bar{Q}^{n-1}(S^n, a^n), \tag{4.5}$$

where α_{n-1} is the learning rate, or *stepsize*. The properties of the sequence $\{\alpha_n\}_{n=0}^{\infty}$ are discussed briefly in Section 4.3 and again in Section 5.1.1.

Intuitively, $\hat{q}$ is a linear combination of the short-term reward and the perceived value of the following state that resembles Equation (4.1), but it does not require knowledge of the transition probability matrix. For this reason, Q-learning is generally referred to as an example of *model-free* reinforcement learning; it is indifferent to the dynamics of the underlying process.

Our goal is to find the set of Q-factors $\{Q^*(s, a)\}$ that denote the value of being in state s and choosing option a, given that we follow the optimal policy thereafter. Thus, if we are in state S^n and know $Q^*(S^n, a)$ for all $a \in \mathcal{A}$, the optimal action is merely:

$$a^* = \arg\max_{a \in \mathcal{A}} \bar{Q}^{n-1}(S^n, a). \tag{4.6}$$

Equivalently, we can rewrite Bellman's Equation as:

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a) \quad \forall\, s \in \mathcal{S}. \tag{4.7}$$

It turns out that–under the appropriate conditions–$Q^n(s,a) \to Q^*(s,a)$ as $n \to \infty$. We present a proof of this fact in the following section.

4.3 Stochastic Approximation Methods and the Convergence of Q-learning

In this section, we present a version of the formal derivation of the Q-learning algorithm presented in Tsitsiklis (1994). This proof hinges on the realization that Q-learning is structurally similar to stochastic approximation algorithms in that it can be decomposed into a sequence of random correction terms that are used to recursively update the existing estimates. For the first part of this section, we prove the convergence of the more general stochastic approximation methods and use the notation of that problem class accordingly. In the latter part of the section, we formally construct Q-learning as a case of stochastic approximation in order to complete the proof.

4.3.1 Convergence of Stochastic Approximations

Consider the vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, the sequence of mappings $F_1, \ldots, F_n$ from $\mathbb{R}^n$ to $\mathbb{R}$, and the countably infinite set of nonnegative "time" indices T^i at which we perform an update of x_i. Then, for every nonnegative integer t:

$$x_i(t+1) = \begin{cases} x_i(t) + \alpha_i(t)[F_i(x(t)) - x_i(t) + w_i(t)] & t \in T^i, \\ x_i(t) & t \notin T^i, \end{cases} \tag{4.8}$$

where w_i is a random, zero-mean noise term, and α is a small and usually decreasing stepsize.[1]

We begin by presenting the necessary assumptions. We address only the special case in which the discount factor γ is on the half-open interval $[0, 1)$ and in which the updates are synchronous. More precisely, we do not address the special cases of undiscounted problems ($\gamma = 1$) or of scenarios in which we perform updates using outdated information.

Assumption 4.1. *The random variables x, w, and α defined above obey:*

1. *$x(0)$ is $\mathcal{F}(0)$-measurable.*
2. *For every i and t, $w_i(t)$ is $\mathcal{F}(t+1)$-measurable.*
3. *For every i and t, $\alpha_i(t)$ is $\mathcal{F}(t)$-measurable.*
4. *For every i and t, we have $\mathbb{E}[w_i(t) \mid \mathcal{F}(t)] = 0$.*
5. *There exist deterministic constants A and B such that*

$$\mathbb{E}[w_i^2(t) \mid \mathcal{F}(t)] \leq A + B \max_j \max_t |x_j(t)|^2.$$

In essence, parts 1-3 of Assumption 4.1 imply that the values of $x(t)$ and $\alpha(t)$ are considered "observed" given the available information at time t, while the noise term is not known until time $t+1$. In other words, $w(t)$ is a random variable up through time t and a fixed constant starting at time $t+1$. Part 4 implies that $x(t+1)$ is a martingale with respect to $\mathcal{F}(t)$, and Part 5 implies that the variance of the noise term is finite. Generally speaking, these assumptions are all satisfied easily. The next

[1] Under this recursion, our estimate at time t is essentially a weighted sum of the "observations" $F_i(x(\tau)) + w_i(\tau)$ for $\tau \in \{0, \ldots, t-1\}$. If we let $\alpha_i(\tau) = 1/\tau$, then $x_i(t+1)$ reduces to $\frac{1}{t}\sum_{\tau=0}^{t}(F_i(x_i(\tau)) + w_i(\tau))$, which is just the sample mean. In general, such a stepsize choice is unwise, as we shall see in Section 5.1.1.

assumption is a classical constraint on the stepsize α for stochastic approximation algorithms:

Assumption 4.2. *For every i:*

$$\sum_{t=0}^{\infty} \alpha_i(t) = \infty, \quad \textit{almost surely,}$$

and there exists some deterministic constant C such that:

$$\sum_{t=0}^{\infty} \alpha_i^2(t) \leq C, \quad \textit{almost surely.}$$

Assumption 4.3. *There exists a vector $x^* \in \mathbb{R}^n$, a positive vector v, and a scalar $\gamma \in [0, 1)$, such that:*

$$\|F(x) - x^*\|_v \ \leq \ \gamma \|x - x^*\|_v \qquad \forall x \in \mathbb{R}^n.$$

In Assumption 4.3 and in the remainder of this section, we let $x \leq y$ be interpreted as $x_i \leq y_i$ for all i and for all $x, y \in \mathbb{R}^n$. We also define the norm $\|\cdot\|_v$ on $\mathbb{R}^n$:

$$\|x\|_v = \max_i \frac{|x|}{v_i}, \qquad x \in \mathbb{R}^n,$$

for any positive vector $v = (v_1, \ldots, v_n)$. We note that if $v_i = 1$ for all i, then $\|\cdot\|_v$ is equivalent to the maximum norm $\|\cdot\|_\infty$.

Theorem 4.1. *Let Assumptions 4.1-4.3 hold. Then the sequence $\{x(t)\}_{t=0}^{\infty}$ converges almost surely to x^*.*

In order to prove this statement, we use a lemma and theorem from Tsitsiklis (1994), which we present here without proof as Lemma 4.1 and Theorem 4.2.

Lemma 4.1. *Let $\{\mathcal{F}(t)\}$ be an increasing sequence of σ-fields. For each t, let $\alpha(t)$,*

$w(t-1)$, and $B(t)$ be $\mathcal{F}(t)$-measurable scalar random variables. Let C be a deterministic constant. Suppose that the following hold with probability 1:

1. $\mathbb{E}[w(t) \mid \mathcal{F}(t)] = 0$.
2. $\mathbb{E}[w^2(t) \mid \mathcal{F}(t)] \leq B(t)$.
3. $\alpha(t) \in [0, 1]$.
4. $\sum_{t=0}^{\infty} \alpha(t) = \infty$.
5. $\sum_{t=0}^{\infty} \alpha^2(t) \leq C$.

Suppose further that the sequence $\{B(t)\}$ is bounded with probability 1, and let $W(t)$ satisfy the recursion:

$$W(t+1) = (1 - \alpha(t))W(t) + \alpha(t)w_i.$$

Then with probability 1, $\lim_{t\to\infty} W(t) = 0$.

This lemma proves the intuitive result that the α-weighted sum of the sequence of zero-mean random noise $w_i(t)$ converges to zero under Assumptions 4.1-4.3. The logical consequence is that the random noise terms do not bias our limiting estimators of Q^* — a clearly desirable property and one of which we take advantage in the upcoming proof. Additionally, we can apply this lemma to obtain the following theorem:

Theorem 4.2. *Let Assumptions 4.1, 4.2, and 4.3 hold. Then the sequence $\{x(t)\}$ is bounded with probability 1.*

With these two results in our arsenal, we are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Without loss of generality, we assume that $x^* = 0$ and that the elements of the vector v from Assumption 4.3 are uniformly equal to one. Such conditions can always be met with simple linear transformations of the coordinate

system. In conjunction with our previous assumptions, Theorem 4.2 establishes that the sequence $\{x(t)\}$ is bounded.

More specifically, Theorem 4.2 implies that there exists some potentially random D_0 such that $\|x_t\|_\infty \leq D_0$ for all t. We fix some $\epsilon > 0$ such that $\gamma(1+2\epsilon) < 1$ and define the recursive sequence:

$$D_{k+1} = \gamma(1+2\epsilon)D_k, \quad k \geq 0,$$

which clearly converges to zero as k tends toward infinity. Now suppose that there exists some time t_k such that $\|x(t)\|_\infty \leq D_k$ for all $t \geq t_k$. This is true by definition for D_0, and we will show that the existence of such a t_k implies the existence of some t_{k+1} such that $\|x(t)\|_\infty \leq D_{k+1}$ for all $t \geq t_{k+1}$. Doing so will complete the proof by induction of the convergence of $x(t)$ to zero. We define the sequence:

$$W_i(t+1) = (1-\alpha_i(t))W_i(t) + \alpha_i(t)w_i(t),$$

where $W_i(0) \triangleq 0$. We then have $\lim_{t\to\infty} W_i(t) = 0$ by Lemma 4.1. We also define $W_i(t_0;t_0) = 0$ for any t_0 and:

$$W_i(t+1;t_0) = (1-\alpha_i(t))W_i(t;t_0) + \alpha_i(t)w_i(t), \quad t \geq t_0.$$

We note that the collection of sequences specified by $W_i(t;t_0)$ are generalizations of the sequence $W_i(t)$, where $W_i(t) = W_i(t;0)$. Furthermore, we can factorize $W_i(t;0)$ as:

$$W_i(t;0) = \left[\prod_{\tau=t_0}^{t-1}(1-\alpha_i(\tau))\right] W_i(t_0;0) + W_i(t;t_0), \tag{4.9}$$

for any $t_0 \leq t$. This expression implies that $|W_i(t;t_0)| \leq |W_i(t;0)| + |W_i(t_0;0)|$.

Additionally, for arbitrary $\delta > 0$, there exists a nonnegative integer T such that

$|W_i(t;0)| \leq \delta/2$ for all $t \geq T$ (cf. Lemma 4.1). Combining this insight with the previous conclusion, we see that $|W_t(t;t_0)| \leq \delta$ for all $T \leq t_0 \leq t$.

Applying both this fact and the supposition at the beginning of the induction, we now define $\tau_k \geq t_k$ such that $|W_i(t;\tau_k)| \leq \gamma\epsilon D_k$ and such that $\|x(t)\|_\infty \leq D_k$ for all $t \geq \tau_k$. We let $Y_i(\tau_k) = D_k$ and:

$$Y_i(t+1) = (1-\alpha_i(t))Y_i(t) + \alpha_i(t)\gamma D_k, \quad t \geq \tau_k, \tag{4.10}$$

which clearly converges to γD_k as t tends to infinity for any fixed k.

We now digress into a second, nested induction proof on t to prove that:

$$-Y_i(t) + W_i(t;\tau_k) \leq x_i(t) \leq Y_i(t) + W_i(t;\tau_k), \quad \forall t \geq \tau_k. \tag{4.11}$$

Since $Y_i(\tau_k) = D_k$, $\|x(\tau_k)\|_\infty \leq D_k$, and $W_i(\tau_k;\tau_k) = 0$ by definition, Equation (4.11) clearly holds for $t = \tau_k$. Now suppose that it holds for some $t > \tau_k$. We then have that:

$$\begin{aligned} x_i(t+1) &\leq (1-\alpha_i(t))(Y_i(t) + W_i(t;\tau_k) + \alpha_i(t)(\gamma D_k + w_i(t)) \\ &= Y_i(t+1) + W_i(t+1;\tau_k), \end{aligned}$$

by substituting the right half of Equation (4.11) into the recursive definition for $x_i(t+1)$ and applying the bounds defined in the preceding paragraph. A symmetrical argument can be made for $-Y_i(t+1) + W_i(t+1;\tau_k) \leq x_i(t+1)$, completing the inner induction.

Combining Equations (4.10) and (4.11) yields:

$$\limsup_{t\to\infty} |x_i(t)| \leq \gamma(1+\epsilon)D_k,$$

which is in turn less than D_{k+1}. We recall that $D_{k+1} = \beta(1+2\epsilon)D_k$, which completes our proof by induction. Letting k tend to infinity in the above expression clearly gives our desired result. □

4.3.2 Q-learning as a stochastic approximation method

We consider states $s \in \mathcal{S}$, actions $a \in \mathcal{A}(s)$ (the set of feasible actions in state s), contributions $c(s,a)$, and discount factor γ exactly as in the beginning of the chapter, as well as a nonnegative stepsize $\alpha_t(s,a)$ satisfying Assumption 4.2. We now rewrite the Q-learning algorithm of Section 4.2 in the form of Equation (4.8):

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)\left[c(s,a) + \gamma \max_{a' \in A(s'(s,a))} Q_t(s'(s,a),a') - Q_t(s,a)\right], \tag{4.12}$$

where $s'(s,a)$ denotes the newly observed state resulting from being in state s and taking action a. Doing so gives rise to the following theorem.

Theorem 4.3. *Consider the Q-learning algorithm, and let* $Q^*(s,a) = \mathbb{E}[c(s,a)] + \gamma \sum_{s'} P_{ss'}(a)V^*(s')$. *Then, assuming that the variance of* $c(s,a)$ *is finite for all* $s \in \mathcal{S}$ *and all* $a \in \mathcal{A}(s)$, $\{Q_t(s,a)\}_{t=0}^{\infty}$ *converges to* $Q^*(s,a)$ *with probability 1, for every* s *and* a *and for any* $\gamma \in [0,1)$.

Proof. The proof of Theorem 4.3 follows largely from Equation (4.12). We define:

$$F_{sa}(Q) = \mathbb{E}[c(s,a)] + \gamma \mathbb{E}\left[\max_{a' \in A(s'(s,a))} Q_t(s'(s,a),a')\right], \tag{4.13}$$

and note that:

$$\mathbb{E}\left[\max_{a' \in A(s'(s,a))} Q_t(s'(s,a),a')\right] = \sum_{s' \in \mathcal{S}} p_{ss'}(u) \max_{a' \in A(s'(s,a))} Q_t(s'(s,a),a').$$

From this expression it is clear that any fixed point Q^* of F satisfies Equation

(4.7). Parts 1-4 of Assumption 4.1 are naturally satisfied in all MDPs, and part 5 is satisfied by the assumption that the variance of $c(s, a)$ is finite. We may easily satisfy Assumption 4.2 by choosing appropriate stepsizes, of which the literature has many examples. Finally, Assumption 4.3 is satisfied for any $\gamma \in [0, 1)$ because Equation (4.13) can be rearranged to yield:

$$|F_{sa}(Q) - F_{sa}(Q')| \leq \gamma \max_{s' \in \mathcal{S}, a' \in \mathcal{A}(s')} |Q_{s'a'} - Q'_{s'a'}|,$$

which is a contraction mapping with respect to the maximum norm. □

4.4 Revisiting the Curse of Dimensionality

The proof presented in Section 4.3 ensures that Q-learning converges to the optimal values $Q^*(s, a)$ in any well-specified system with a finite number of states and actions, assuming that we have a sampling policy that will observe every state-action pair infinitely often. Additionally, the model-free nature of Q-learning solves the one of the two problems mentioned at the end of Section 4.1 — namely, the dependence of exact dynamic programming techniques on knowledge of the transition matrix.

However, Q-learning is nearly as susceptible to the curse of dimensionality as are its model-dependent counterparts. In practice, Q-learning is impractical even for problems with moderately sized state and action spaces. The following chapter introduces an approximation to the Q-learning algorithm that utilizes the theory of DP-GLMs from the preceding chapters and is naturally extensible to very large or even continuous state and action spaces.

Chapter 5

Approximate Q-learning with DP-GLMs

This chapter addresses the difficulties in scaling Q-learning to large or continuous state-action spaces, which we raised in Section 4.4. Since each state-action pair is typically assessed independently, a single iteration of the original algorithm is $\mathcal{O}(|\mathcal{S}| * |\mathcal{A}|)$, where $\mathcal{S}$ and $\mathcal{A}$ are defined as in the preceding chapter. However, if we suppose that there is some similarity between nearby state-action pairs — a reasonable assumption in most circumstances — then we propose that finding a Q-factor surface on the state-action space may be reduced to a simple regression problem.

The immediate issue with such an approach, however, is the availability of a regression model that will adapt appropriately to the data in the presumably unknown state-action-Q space. In this chapter, we develop a variant of Q-learning that uses DP-GLMs to approximate the Q-factors, which we generally refer to as approximate Q-learning with DP-GLMs, or just approximate Q-learning. We conclude by drawing parallels with the kernel-based reinforcement learning of Ormoneit and Sen (2002) and discussing potential theoretical properties.

5.1 Approximate Q-Learning with DP-GLMs

Recall the iterative definition of Q-learning for state s, action a, and $n \geq 1$ from Chapter 4:

$$\bar{Q}^n(s,a) = (1-\alpha_{n-1})\bar{Q}^{n-1}(s,a) + \alpha_{n-1}(c(s,a) + \max_a \bar{Q}^{n-1}(s',a)), \tag{5.1}$$

where $\bar{Q}^0(s,a)$ is defined arbitrarily.[1]

In approximate Q-learning, we let $\bar{Q}^n(s,a)$ denote the predicted value from a DP-GLM regression of $Q(s,a)$ at iteration n. The DP-GLM model takes the joint state-action space as its covariate domain and the space of Q-factors as its response domain. We define $\hat{q}^{n-1}(s,a)$ to once again be the noisy "observations" of the Q-factors:

$$\hat{q}^n(S_t,a_t) = (1-\alpha_{n-1})\bar{Q}^{n-1}(S_t,a_t) + \alpha_{n-1}\left(c(S_t,a_t) + \max_{a'} \bar{Q}^{n-1}(S_{t+1},a')\right), \tag{5.2}$$

where t indexes the elements of the sample path upon which the nth regression model will be trained. We define the DP-GLM operator Λ, such that:

$$\bar{Q}^{n+1}(s,a) = (\Lambda\hat{q}^n)(s,a) \quad \forall\, s,a, \tag{5.3}$$

generates the $(n+1)$st set of estimates of the Q-factors by using the set of $\{\hat{q}^n(s,a)\}$ as training data for a DP-GLM regression. We observe the similarities between these equations and their analogs in the preceding chapter.

This adaptation of traditional Q-learning allows Q-factors to be approximated for unobserved state-action pairs, which can dramatically reduce the minimum number of samples required in order to find a good policy. Particularly in large or continuous state spaces and in large action spaces with a finite number of observations — such

[1] Convention suggests initializing to zero for all s and a, but $c(s,a)$ can also serve as an intuitively reasonable starting point.

Algorithm 5.1 Q-Learning with DP-GLM

Require: Sampling policy π, stepsize sequence $\{\alpha\}$
  Initialize $\bar{q}^0(s,a)$, $\hat{q}^0(s,a) \leftarrow c(s,a)$, $n \leftarrow 0$
  loop
    Generate a set of starting states $\{s_{0,k}\}_{k=1:K}$
    for $k = 1 : K$ **do**
      for $t = 1 : T$ **do**
        Choose an action $a^{\pi}_{t-1,k}$ by policy π
        Generate state $s_{t,k}$ by taking action $a^{\pi}_{t-1,k}$ in state $s_{t-1,k}$
        $\hat{q}^n(s_{t-1,k}, a_{t-1,k}) \leftarrow (1 - \alpha_{n-1})\bar{q}^{n-1}(s_{t-1,k}, a_{t-1,k}) + \alpha_{n-1}(c(s,a) + \max_a \bar{q}^{n-1}(s_{t,k}, a))$
      end for
    end for
    $\bar{q}^n(s,a) \leftarrow \Lambda\{\hat{q}^n(s_{t,k}, a_{t,k})\}$
    $n \leftarrow n + 1$
  end loop

as those found in the real world — this property is quite advantageous. The full algorithm for approximate Q-learning with DP-GLMs is presented in Algorithm 5.1.

Note the reference to a sampling policy — denoted by π — in Algorithm 5.1. The sampling policy is a necessary practical construct for determining which action to sample given the current state. Under the assumptions of Section 4.3.1, the only constraint is that, in the limit, it must sample every possible action in every state an infinite number of times.[2]

That said, our goal is to quickly and accurately discover the optimal set of actions. Intuition suggests that we should therefore investigate actions that we believe to be more promising with correspondingly higher frequency. However, if we only sample the possibilities that we believe to be good, then we run the risk of inappropriately discarding good, or even optimal, options because we (wrongly) believe that they are less desirable. This tradeoff between choosing what we believe to be the best action and choosing an action that we consider less appealing but about which we may be wrong is often referred to as the dilemma of *exploitation* versus *exploration*. The terms are so named because our choice can be framed as a decision between exploiting our

[2]This is a consequence of our assumption that the set of update times T^i is countably infinite.

current knowledge and exploring in hopes of gaining new, valuable information. The study of sampling policies that address this problem is an entire area of active research called *optimal learning*. Interested readers are referred to Frazier et al. (2009) and Powell and Frazier (2008) for an introduction to the field.

For the practical applications of the upcoming section, we use an ϵ-greedy sampling policy, which for some $\epsilon \in (0, 1)$ chooses a uniformly random choice (explores) with probability ϵ and selects what we believe to be the best action (exploits) with probability $1 - \epsilon$.

5.1.1 Stepsize Selection and the Rate of Convergence

We digress briefly for a discussion of the stepsize α. Although Q-learning is guaranteed to converge in the limit, the robustness of its convergence in finite time is an obvious issue of practical concern. Empirical results indicate that the rate of convergence for stochastic approximation methods, such as Q-learning, depends strongly on the stepsize α. In particular, Ryzhov et al. (2009) show that for the common stepsize choice of $1/n$, convergence can take on the order of 10^{20} iterations for a trivial one-state, one-action problem. We thus recommend that practitioners take special care in choosing a stepsize rule. The so-called optimal stepsize of Ryzhov et al. is a good choice but can be difficult to implement. As a heuristic alternative, we suggest a variant on the standard harmonic rule that is tied to the discount factor γ:

$$\alpha_n = \frac{a_0}{a_0 + n}, \quad \text{where} \quad a_0 \triangleq \frac{1}{1 - \gamma} + b$$

for some constant $b \in \mathbb{R}^+$. Harmonic rules have the advantage of generally fast, reliable empirical convergence, and allowing the tunable parameter a_0 to increase with the discount factor γ has the intuitive appeal of letting the stepsizes stay large while the estimates "work up" to the neighborhood of the final value. Conversely, the

approximation stabilizes as the stepsize decreases, because small stepsizes mitigate the effects of random noise. The undesirable behavior of a $1/n$ stepsize rule occurs because the stepsize declines too quickly at first, thereby causing the estimator to be unduly biased by the first several terms. Stepsize selection reduces to an assessment of the tradeoffs between smoothness and efficiency; good stepsize rules retain a high degree of both.

5.2 Applications of Approximate Q-Learning

In this section, we apply Algorithm 5.1 to two related problems in energy storage. Specifically, we consider the situation of an idealized wind farm with one wind turbine and one battery storage device into which it can siphon excess electricity. The available energy supply is modeled as a stochastic process, since wind speed is neither constant nor known in advance. Specifically, we assume that the wind speed is determined by an order-1 autoregressive process on the square root of the wind speed, as suggested by Brown et al. (1984). We assume that our turbine is maximally efficient and can thus convert wind of velocity v_t (m/s) to power in accordance with the limit provided by Betz' law:

$$P_t = \frac{1}{2} * B * \omega * S * v_t^3, \tag{5.4}$$

where $B \approx 0.593$ is the Betz limit and $\omega = 1.225 kg/m^3$ is the density of air. S is the area swept by the rotor blades of the turbine, which we choose to be $S = \pi 70^2 m^2$. The quantity of energy generated in an interval $(t, t + \Delta t)$ with constant wind speed v_t is then equal to $E = P * \Delta t$.

We let $R_t \geq 0$ represent the level of charge in the battery at time t and assume that the battery has a known maximum capacity R_{max}, beyond which it cannot store any more energy. We also assume that with the battery are associated constants $\rho_{in}, \rho_{out} \in [0, 1]$ that represent the efficiency with which it may be charged and

discharged. That is, for every unit of energy that we put in the battery (assuming that it is not full to capacity), we may extract $\rho_{in}\rho_{out} \leq 1$ units at some arbitrary point in the future.

Our wind farm has an obligation to supply a minimum quantity D_{min} of energy per unit time to an external entity ("the grid"), but may sell more than this minimum up to a constant demand constraint D. Failure to meet the minimum obligation carries a penalty, which we denote as c_u per unit of energy.

Finally, choosing to sell $x_{t,b} + x_{t,w}$ units of energy to the grid at time t accrues a contribution of $p_t * (x_{t,b} + x_{t,w})$, where p_t denotes the price, $x_{t,b}$ is the quantity of energy taken from the battery to sell to the grid, and $x_{t,w}$ is the analogous amount of energy sold to the grid from wind power, all at time t. For the variant examined in Section 5.2.1, we take $p_t \triangleq p$ for all t; in Section , we model it as a stochastic process instead.

In both problems, our decision variable x_t is decomposed into $(x_{t,b}, x_{t,w}, x_{t,c})$, where $x_{t,b}$ and $x_{t,w}$ are as defined above, and where $x_{t,c}$ denotes the amount of energy from excess wind power used to charge the battery.[3] Given this decision vector and the current state R_t of the battery, we can easily calculate $R_{t+\Delta t}$ using the transition function:

$$R_{t+\Delta t} = min(R_t + \rho_{in} * x_{t,c} - x_{t,b}, R_{max}). \tag{5.5}$$

Having established our problem framework, we can now examine our two specific test cases.

[3]Other variants, including work by Warren Scott at Princeton University's CASTLE Lab, allow one to buy energy from the grid in order to charge the battery. We simplify the decision slightly by assuming that the underage penalty c_u is sufficiently large that we would never buy energy from the grid unless our minimum obligation is met.

5.2.1 Energy Storage with a Stochastic Supply Process

In our first test problem, we apply approximate Q-learning with discount factor $\gamma = 0.8$. We choose a time step of ten minutes, or $\delta t = 600s$, and set $p_t = 0.20$ dollars per kilowatt-hour for all t. Additionally, we set the underage penalty $c_u = 0.50$ dollars per kilowatt-hour. We take $D_{min} = 250$ kilowatt-hours and $D = 500$ kilowatt-hours as the minimum obligation and maximum demand per δt, respectively. Finally, we choose $\rho_{in} = \rho_{out} = 0.9$ as the charging and discharging efficiencies of the battery. We define our state at time t as $S_t = (R_t, v_t)$.

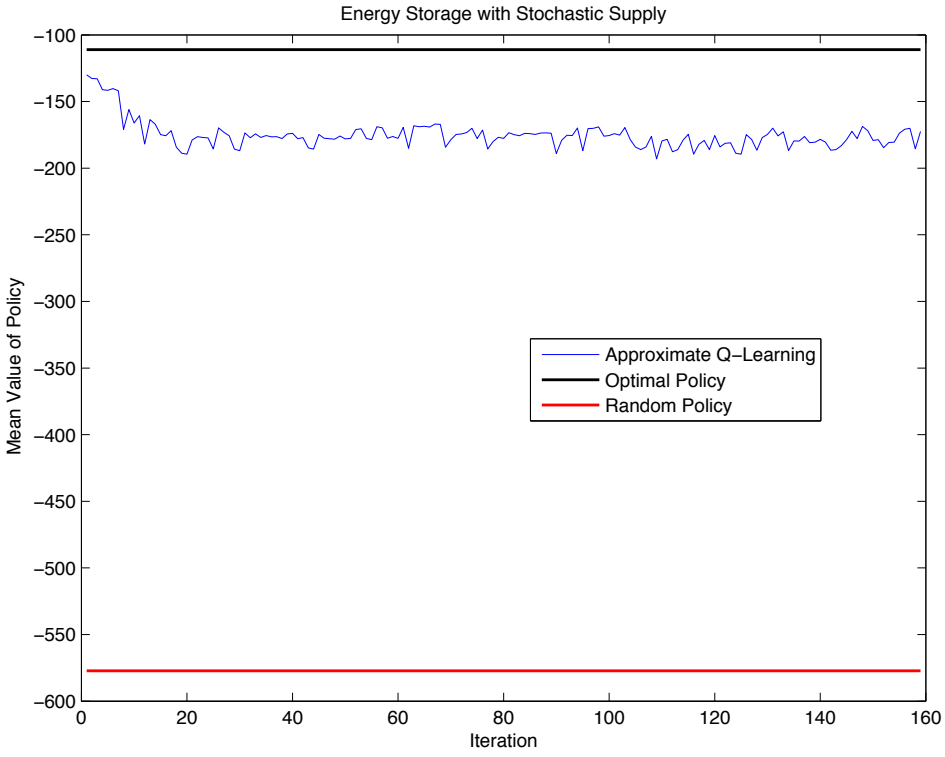

Figure 5.1: The values of sequential policies generated by applying Algorithm 5.1 to energy storage with stochastic supply. The solid red line represents the value of a policy of random actions, which serves as our baseline. The solid black line represents the value of the optimal policy.

Figures 5.1 and 5.2 show the sequential values of the policy generated by Algorithm 5.1 at each of 160 iterations. Figure 5.1 benchmarks these policies against both the

optimal policy, which is obtained by exact value iteration, and the mean value of 100 policies composed of uniformly random actions. Approximate Q-learning does quite well, averaging 86.2% of the optimal value in comparison to the baseline.[4]

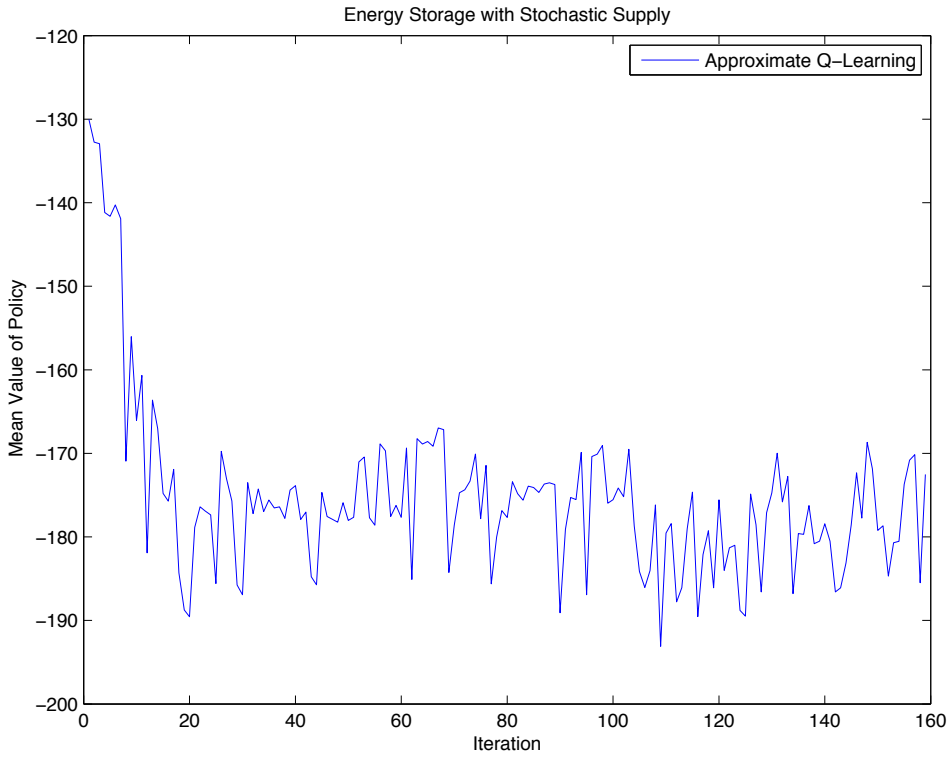

Figure 5.2: A closer view of the values of policies generated on successive iterations by applying Algorithm 5.1 to energy storage with stochastic supply.

We observe — particularly in the closer viewing of Figure 5.2 — that approximate Q-learning obtains its best value on the first iteration and declines thereafter. We postulate that this is because of bias introduced by the sampling policy that is not present in the initialization phase. During initialization, we sample a large number of states with purely random actions in the pursuit of a reasonably good initial estimate. In the subsequent phases, we use an ϵ-greedy policy with harmonically declining

[4]Benchmarking against the value of the worst possible policy (-1125.0) seems less informative — not to mention, slightly disingenuous — when compared to our more moderate baseline of choice.

epsilons to sample actions, which is inherently biased towards actions that it "believes" to be best at the time. In conjunction with the smoothing inherent in the regression model, this bias can cause us a degradation of our estimator with time. In accordance with our proposition in Section 5.3, however, Algorithm 5.1 appears to converge to a fixed point starting around the twentieth iteration, after which the value stabilizes in the range of $(-190, -170)$.

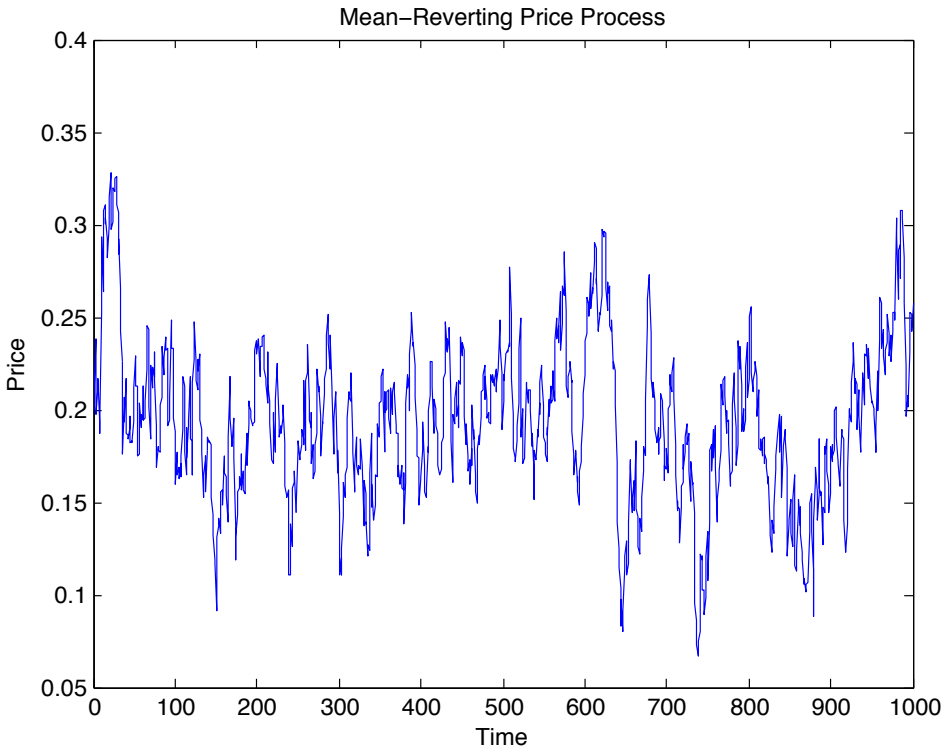

Figure 5.3: Sample realization of our mean-reverting price process.

5.2.2 Energy Storage with Stochastic Supply & Price Processes

Our second test case is nearly identical to the first, with the exception of the price process p_t. Now, rather than being a predetermined constant, we take p_t to be an $\mathcal{F}(t)$-measurable random variable following an Ornstein-Uhlenbeck process, which is

mean-reverting and can be described by the stochastic differential equation:

$$dp_t = \kappa(\mu - p_t)dt + \sigma dW_t, \tag{5.6}$$

where κ is the reversion rate, μ is the long-term mean, σ is the volatility, and W_t is a Wiener process (i.e., zero-mean Brownian motion). For our present experiment, we set $\mu = 0.2$, $\sigma = 0.04$, and $\kappa = 0.5$. A sample of the resulting process is shown in Figure 5.3 above. Our state variable is now (R_t, v_t, p_t). The decision vector x_t is identical to before.

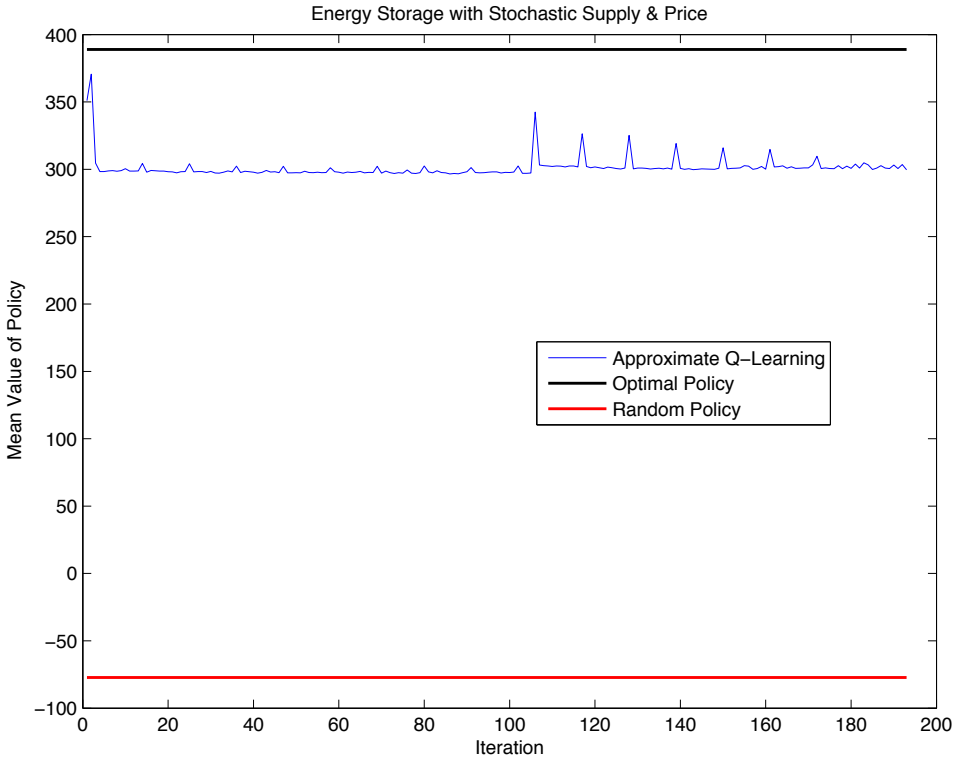

Figure 5.4: Value of policies generated by approximate Q-learning for energy storage with stochastic supply and prices. The solid red line represents the value of a policy of uniformly random actions, which serves as our baseline. The solid black line represents the value of the optimal policy.

The results of applying Algorithm 5.1 in such a context is strikingly similar to

those of Section 5.2.1, although an analogously discretized model of the state space is now exponentially larger than in the preceding case. We were able to take advantage of the fact that the expectation of the Ornstein-Uhlenbeck process is precisely its mean parameter μ in order to determine the true optimal policy; similar models lacking the possibility of such an analytical sleight of hand would be considerably more difficult to solve exactly.

Though the variance of our sequence of estimators has visibly increased — as we would expect with the addition of random contributions — the quality of the average policy is only mildly diminished. Evaluating them once again with respect to our baseline calculated from policies of random actions, the average value of the policies generated by our approximate Q-learning algorithm is 81.16% of the value of the optimal policy. The robustness of the algorithm under the increase to a six-dimensional, continuous state-action space is impressive and suggests potential scalability to higher-dimensional spaces, though doing so is beyond the scope of this work.

5.3 Convergence to a Fixed Point

We conclude this chapter with a discussion of the theoretical properties of our approximate Q-learning algorithm. Specifically, we desire a theoretical guarantee of some sort of convergence. Figure 5.5 shows empirical convergence of the optimal action estimates, which is promising. We conjecture that our approximate Q-learning algorithm converges to a unique fixed point in a similar fashion to the class of kernel-based reinforcement learning algorithms derived by Ormoneit and Sen (2002):

Conjecture 5.1. *Algorithm 5.1 converges to a unique fixed point with probability 1.*

A potential proof of this proposition would involve proving that the DP-GLM operator Λ forms a contraction mapping with respect to the maximum norm $\|\cdot\|_\infty$; convergence

would follow under the Banach fixed point theorem.[5]

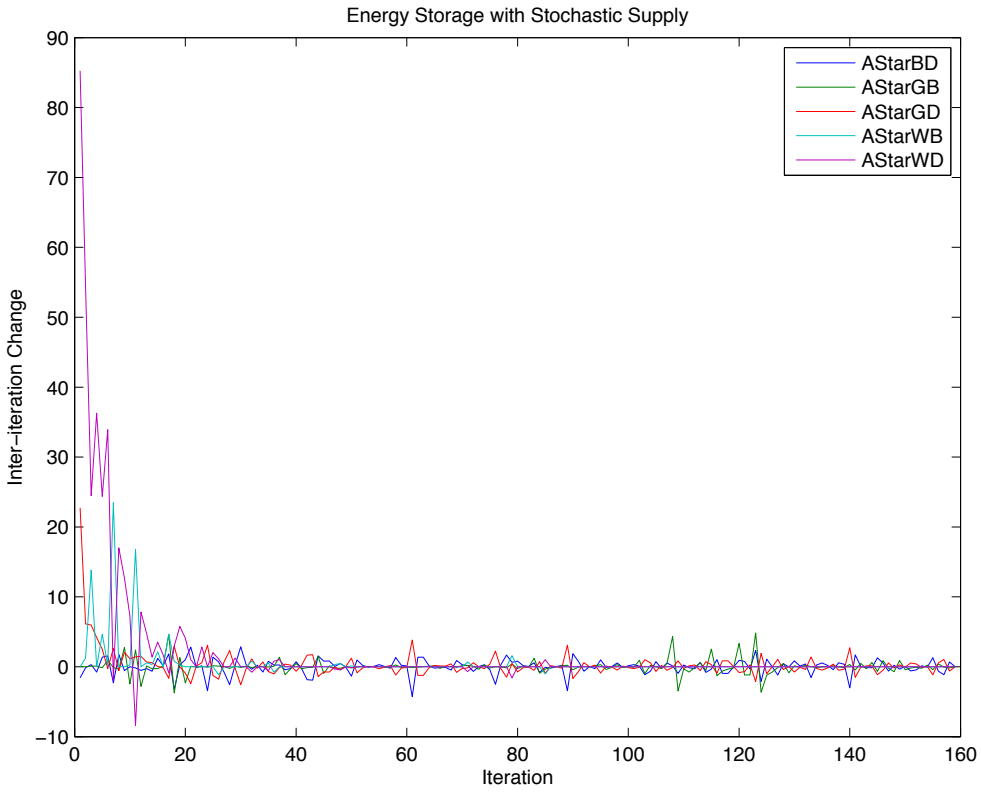

Figure 5.5: Plots of the between-iteration changes in the estimates of the optimal actions. Approximate Q-learning appears to be converging to a fixed point.

Moving beyond convergence to a fixed point, we might seek to prove convergence to the optimum, though doing so may be impossible. However, since DP-GLM is weakly consistent and asymptotically unbiased, its limiting properties are quite similar to those of kernel regression; thus, the techniques for proving the convergence of the latter may provide guidance in an endeavor to do so for our algorithm.

[5]I may be wrong, but I doubt it.

Chapter 6

Concluding Remarks

This thesis explored two algorithms, Dirichlet process mixtures of generalized linear models and Q-learning, and made a significant contribution with respect to each. After presenting the relevant underlying theory of DP-GLMs, we derived a novel mean field variational algorithm as an alternative to the existing Markov chain Monte Carlo methods for posterior inference. This approach deterministically finds the tractable approximation that minimizes the Kullback-Leibler divergence from the target posterior, and both the empirical properties of this variational algorithm and its extension to the full class of generalize linear models are very promising areas for potential research.

Additionally, we proposed an algorithm for approximate Q-learning in continuous state and action spaces, which obtained empirically sound results in our two test cases. Applications of this algorithm to higher-dimensional problems — as well as to the domain of energy storage, in particular — could also be of interest for further study. Theoretical properties of this algorithm were conjectured, but the exact nature of the approximation remains an open question.

Bibliography

Antoniak, C. E. (1974). Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174.

Bertsekas, D. P. (1999). *Nonlinear Optimization*. Athena Scientific, Nashua, N.H.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson Distributions Via Polya Urn Schemes. *The Annals of Statistics*, 1(2):353–355.

Blei, D. M. and Jordan, M. I. (2006). Variational Inference for Dirichlet Process Mixtures. *Bayesian Analysis*, 1(1):121–144.

Braun, M. and McAuliffe, J. D. (2010). Variational Inference for Large-Scale Models of Discrete Choice. *Journal of the American Statistical Association*, 105(489):324–335.

Brown, B. G., Katz, R. W., and Murphy, A. H. (1984). Time series models to simulate and forecast wind speed and wind power. *Journal of Applied Meteorology and Climatology*, 23(8):1184–1195.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Duane, S. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230.

Frazier, P. I., Powell, W. B., and Dayanik, S. (2009). The Knowledge-Gradient Policy for Correlated Normal Beliefs. *INFORMS Journal on Computing*, 21(4):599–613.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman & Hall.

Ghahramani, Z. and Beal, M. J. (2001). Propagation algorithms for variational Bayesian learning. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Neural Information Processing Systems*, number 13, pages 507–513. MIT Press.

Hannah, L. A. (2010). *Stochastic Search, Optimization and Regression with Energy Applications*. Dissertation, Princeton University.

Hannah, L. A., Blei, D. M., and Powell, W. B. (2009). Dirichlet Process Mixtures of Generalized Linear Models.

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193.

Manning, L. W. (2010). Bayesian Nonparametric Regression with a Dirichlet Process Mixture of Generalized Linear Models.

Manning, L. W. (2011). COS 513 Final Project : Stochastic Search Variable Selection for Dirichlet Process mixtures of Generalized Linear Models.

Neal, R. M. (1993). Probabilistic Inference using Markov Chain Monte Carlo Methods.

Neal, R. M. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265.

Neal, R. M. (2010). *MCMC using Hamiltonian dynamics*. Chapman & Hall / CRC Press.

Ormoneit, D. and Sen, S. (2002). Kernel-Based Reinforcement Learning. *Machine Learning*, 49(2):161–178.

Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, second edition.

Powell, W. B. and Frazier, P. I. (2008). Optimal Learning. *INFORMS*, pages 213–246.

Ross, S. M. (2006). *Simulation, Fourth Edition (Statistical Modeling and Decision Science)*. Academic Press.

Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw Hill, New York, New York, USA, 3rd edition.

Rudin, W. (1986). *Real and Complex Analysis (International Series in Pure and Applied Mathematics)*. McGraw-Hill.

Ryzhov, I. O., Frazier, P. I., and Powell, W. B. (2009). Stepsize Selection for Approximate Value Iteration and a New Optimal Stepsize Rule. *Journal of Machine Learning Research*, 1(1).

Sethuraman, J. (1994). A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 4(2):639–650.

Shahbaba, B. and Neal, R. (2009). Nonlinear models using Dirichlet process mixtures. *The Journal of Machine Learning Research*, 10:1829–1850.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press.

Tsitsiklis, J. (1994). Asynchronous Stochastic Approximation and Q-Learning. *Machine Learning*, 202:185–202.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Watkins, C. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103.

Xing, E., Jordan, M. I., and Russell, S. (2003). A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, volume 19, pages 583–591, San Fransisco, CA. Morgan Kaufmann Publishers.