Notes: Determinantal Point Processes

Theodore O. Brundage

Researched under the guidance of my advisor,
Barbara Engelhardt

Princeton University,
Princeton, New Jersey

November, 2015 - Present

Contents

1 Meeting Notes **2**
- 1.1 Meeting with Barbara 11/06/15 . . . 2
  - 1.1.1 Questions to ask . . . 2
  - 1.1.2 Notes from Barbara, 11/06/15 . . . 2
- 1.2 Meeting with Barbara 11/09/15 . . . 3
  - 1.2.1 Questions to ask . . . 3
  - 1.2.2 Notes from Barbara 11/09/15 . . . 3
- 1.3 Metting wtih Greg 11/10/15 . . . 3
  - 1.3.1 Questions to ask . . . 3
  - 1.3.2 Notes from Greg 11/10/15 . . . 4
- 1.4 Meetings about approaching §2.4.1 . . . 5
  - 1.4.1 Notes from Barbara 12/8/15 . . . 5
  - 1.4.2 Notes from Barbara 12/18/15 . . . 5
- 1.5 Notes from Barbara 3/8/16 . . . 6

2 Problem Description and Notes **7**
- 2.1 Parameterized DPPs . . . 7
  - 2.1.1 Motivation . . . 7
  - 2.1.2 Definitions . . . 7
  - 2.1.3 Probabilities . . . 8
- 2.2 Linear Regression with Parameterized DPP Sparsity . . . 8
  - 2.2.1 Overview . . . 8
  - 2.2.2 Definitions of Terms . . . 8
  - 2.2.3 Problem Description . . . 10
- 2.3 Marginal and Conditional Distributions . . . 11
  - 2.3.1 Conditional Distribution of $\boldsymbol{\beta}$ Given σ^2 . . . 11
  - 2.3.2 Full Marginal Distribution of $\boldsymbol{\beta}$ . . . 11
  - 2.3.3 Full Conditional Distribution of $\boldsymbol{y}$: . . . 11
  - 2.3.4 Conditional Distribution of $p(\boldsymbol{y}|\sigma^2, \boldsymbol{\gamma})$ Marginalizing $\boldsymbol{\beta}$ . . . 12
  - 2.3.5 Conditional Distribution of $p(\boldsymbol{y}|\boldsymbol{\gamma})$ Marginalizing $\boldsymbol{\beta}$ and σ^2 . . . 12
- 2.4 Prior Distributions . . . 12
  - 2.4.1 Analytic Attempts at Solving for $\boldsymbol{\gamma}$ . . . 12
  - 2.4.2 Ideas for Priors on $\boldsymbol{\theta}$ . . . 14
  - 2.4.3 Uninformative Priors on Hyperparameters . . . 15

3 Solutions to the DPP-Sparse Linear Regression Problem 18
3.1 An Ordered Systematic Approach . . . 18
3.1.1 Computation of $\boldsymbol{\theta}$ Maximum Likelihood Estimation, Marginalizing $\boldsymbol{\gamma}$ 19
3.1.2 A word on concavity . . . 19
3.1.3 Problems with this approach . . . 20
3.2 A Variational Inference Approach: Partial Pre-Marginalization . . . 20
3.2.1 Likelihood and Gradient of σ^2 . . . 20
3.2.2 Likelihood and Gradient of $\boldsymbol{\theta}$ . . . 21
3.2.3 Likelihood and Optimization of $\boldsymbol{\gamma}$ . . . 21
3.2.4 Making Predictions . . . 21
3.3 A Variational Inference Approach: Maximal Pre-Marginalization . . . 22
3.3.1 Likelihood and Gradient of a_0 . . . 22
3.3.2 Likelihood and Gradient of b_0 . . . 23
3.3.3 Likelihood and Gradient of c . . . 24
3.3.4 Likelihood and Gradient of $\boldsymbol{\theta}$ . . . 24
3.3.5 Optimization of $\boldsymbol{\gamma}$ . . . 24
3.3.6 Eliminating b_0 . . . 25
3.3.7 Making Predictions . . . 26
3.4 Optimizing Hyperparameters With Partial Marginalization of $\boldsymbol{\gamma}$ . . . 26
3.4.1 Approximation . . . 27
3.4.2 Marginalizing With Bornn's Parameterization . . . 27
3.4.3 Pre-Estimation Solution . . . 28
3.5 Computational Adjustments for the Large-p Small-n Setting . . . 30
3.5.1 Reduction of $\hat{\boldsymbol{\beta}}$ . . . 30
3.5.2 Reduction of the "Difference Projection:" $b_N - b_0$ . . . 31
3.5.3 Computing DPP Likelihoods in $\mathbb{R}^{n\times n}$ . . . 31
3.5.4 Sampling from the DPP Dual . . . 32

4 Code 33
4.1 Synthetic Data Generation . . . 33
4.1.1 `LinearDataGenerator.py` . . . 33
4.1.2 `KojimaKomakiDataGen.py` . . . 34
4.2 Utility Functions: `DPPutils.py` . . . 35
4.3 Bayesian Network Variable: `BNV.py` . . . 36
4.3.1 BNVs for Maximal Pre-Marginalization Variational Inference Experiment . . . 36
4.4 Prior Distributions, $\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{X})$, `theta_priors.py` . . . 37
4.5 Variational Inference Control Object: `VI.py` . . . 37
4.5.1 Structure and Instance Variables . . . 37
4.5.2 Utilities . . . 37

5 Experiments 39
5.1 Regression Comparisons with Fake Data . . . 39
5.1.1 Small p with Bornn's Parameterization . . . 39
5.1.2 Pre-Optimization of Hyperparameters with Partial Marginalization of $\boldsymbol{\gamma}$ with Bornn's Parameterization . . . 41

6 Paper Reviews **46**
6.1 Bornn et al., 2014 - Diversifying Sparsity Using Variational Determinantal Point Processes . . . 46
6.1.1 Overview . . . 46
6.1.2 Bayesian Variable Selection . . . 46
6.1.3 Priors: . . . 46
6.1.4 Joint Likelihood . . . 47
6.1.5 Marginal Likelihood . . . 47
6.1.6 Posterior Probability . . . 47
6.1.7 Bornn's Algorithm . . . 48
6.1.8 A word on Φ . . . 48
6.2 Affandi, Fox, Adams, Taskar, 2014 - Learning the Parameters of DPP Kernels 49
6.2.1 Overview . . . 49
6.2.2 Notation . . . 49
6.2.3 Optimization Methods . . . 50
6.2.4 A Bayesian Approach . . . 50
6.3 Gillenwater, Kulesza, Fox, Taskar, 2014 - Expectation-Maximization for Learning DPPs . . . 51
6.3.1 Overview . . . 51
6.3.2 Notation . . . 51
6.3.3 Projected Gradient Ascent . . . 52
6.3.4 Eigendecomposition . . . 52
6.4 Kojima, Komaki, 2014 - DPP Priors for Bayesian Variable Selection in Linear Regression . . . 53
6.4.1 Overview . . . 53
6.4.2 Model . . . 53
6.4.3 Review of Bayesian Variable Selection Methods . . . 53
6.4.4 DPP Models . . . 54
6.4.5 Numerical Methods . . . 54
6.4.6 Applying models to real data . . . 55
6.5 Kulesza, Taskar, 2011 - Learning Determinantal Point Processes . . . 55
6.5.1 Overview . . . 56
6.5.2 Notation . . . 56
6.5.3 Comparison of DPPs and MRFs . . . 56
6.5.4 MLE Learning . . . 57
6.5.5 MAP Inference . . . 58
6.6 Kulesza, Taskar, 2013 - Determinantal Point Processes for Machine Learning 58
6.6.1 Overview . . . 59
6.6.2 Sampling from a DPP . . . 59
6.6.3 Learning Quality Parameters . . . 59
6.6.4 Approximating the MAP summary . . . 61
6.7 Engelhardt, Adams, 2014 - Bayesian Sparsity with Gaussian Fields . . . 61

Appendices

A Derivations of Distribution Closed Forms **63**
A.1 Regressor Coefficient Marginal Distribution, $p(\boldsymbol{\beta})$ 63
A.1.1 Marginalizing out σ^2 63
A.1.2 Implications for the Hyperparameters 64
A.2 Marginalizing Regressors: An Analytic Solution for $p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2)$ 65
A.2.1 Conditional distribution of $\boldsymbol{y}$ 65
A.2.2 Conditional distribution of $\boldsymbol{\beta}$ 66
A.2.3 Solution 66
A.3 Marginalizing Regressors and Variance: An Analytic Solution for $p(\boldsymbol{y}|\boldsymbol{\gamma})$. 66
A.3.1 Solution 67
A.3.2 Implications for the Hyperparameters 68
A.3.3 Computational Note 69

B Derivations of Likelihoods and Gradients **70**
B.1 Partial Pre-Marginalization 70
B.1.1 Partial Pre-Marginalization: Likelihood of σ^2 70
B.1.2 Partial Pre-Marginalization: Likelihood of $\boldsymbol{\gamma}$ 71
B.1.3 Likelihood of $\boldsymbol{\theta}$ 71
B.2 Maximal Pre-Marginalization 72
B.2.1 Maximum Pre-Marginalization: Likelihood of a_0 72
B.2.2 Maximum Pre-Marginalization: Likelihood of b_0 72
B.2.3 Maximum Pre-Marginalization: Likelihood of c 73
B.2.4 Maximum Pre-Marginalization: Likelihood of $\boldsymbol{\gamma}$ 74

Chapter 1

Meeting Notes

1.1 Meeting with Barbara 11/06/15

1.1.1 Questions to ask

- Address 6.1 section on $\mathbf{\Phi}$ and the distinction between rows of similarity features and columns of diversity features. Column vs row obviously not a big deal, but are similarity and diversity features inverses?
- What is the shape of $\boldsymbol{X} \in \mathbb{R}^{N \times M}$ (M gives number of features, N gives number of samples.) Using L-ensemble standard representation is advantageous for $M < N$, while the dual representation, C is advantageous for $M > N$. Currently assuming $M \in O(N)$.
- What is parameterized and what is not (w.r.t. posterior distributions, q in Bornn). When we augment vectors, do we simply extract $\boldsymbol{\gamma}$ later? What about exponentiation of Z_θ?

1.1.2 Notes from Barbara, 11/06/15

- In general, similarity *is* different from diversity, though we for the purposes of implementing Bornn's work, we need not worry and treat L as it is given, with Φ as the equivalent of $B^\top$ in [9].
- Wait to actually get data for point two.
- Barbara suggests to NOT learn on any intercepts, and just use $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, ignoring all extensions like $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}, \theta_0]$.
- Barbara agrees that the normalization term $U(\boldsymbol{\theta})$ is incorrect in [2], that it should have a logarithm.
- Don't be confused by the language in Bornn – columns of $\boldsymbol{X}$ are features, and each row is an observation.

1.2 Meeting with Barbara 11/09/15

1.2.1 Questions to ask

- Data generation: If I generate $\boldsymbol{\gamma}$ from a Bernoulli prior, how do I calculate the "correct" $\boldsymbol{\theta}$? This parameter should only inform the distribution (DPP) that is most likely to have produced $\boldsymbol{\gamma}$ given the data.
  - Should I brute force find the $\boldsymbol{\theta}$ that maximizes the probability that my $\boldsymbol{\gamma}$ is selected? This seems intractable if not really inefficient.
  - Correctness? (Is this a Bayesian/Frequentist issue?)
- Marginal likelihood parameters.
  - I believe that there's a typo in Bornn here.
  - I got a correct value from [14]. Is it correct? Are those exponents? on b_0 and b_N?
  - How should I verify? Should I be doing this math? Is there a canonical source?

1.2.2 Notes from Barbara 11/09/15

- Asking about the "true" $\boldsymbol{\theta}$ is sort of a meaningless question. Here, Bornn is just minimizing KL-divergence, and sort of forcing $q(\boldsymbol{\theta})$ to be a DPP, without making any assertions about its prior nature. (Or the opposite, for `DPP-Bernoulli`). Regardless, we should ignore this method (maybe reproduce on data to test code??) and re-implement with a DPP prior *and* posterior.
- Yes, the marginal likelihood is a typo. Walli [14] is correct, but take a look t Hoff's book on Bayesian Stats as a reference.
- For this new implementation of Born, I will need to find $p(\boldsymbol{y}|\boldsymbol{\gamma})$. Look into how to do this.
  - Maybe contact Alison Cheney? (Talk to Greg first).
  - ELBO and KL-Divergence minimization

1.3 Metting wtih Greg 11/10/15

1.3.1 Questions to ask

- How do I perform Bayesian Inference?
  1. First, what is the general, theoretical structure. I believe it's choosing a prior, calculating the marginals, and evaluating, typically iteratively, the parameterization of the posterior.

2. Second, how do we do that? I believe there are a ton of methods and Barbara has suggested minimizing KL-divergence using this ELBO thing. What is that? Where should I look? also, this is one method. Where's a good place to learn about other methods? Good examples that I can practice with? How does Greg think about choosing different methods? Am I just approaching this totally incorrectly?

- Is my understanding of Bayes Rule as it applies to latent variables and fixed parameters correct? My immediate problem and thinking is below

My goal is to find $p(\boldsymbol{y}|\boldsymbol{\gamma})$. I know from Bornn [2]

$$p(\boldsymbol{y}, \boldsymbol{\beta}, \sigma, \boldsymbol{\gamma}|\boldsymbol{X}, \Lambda_0, a_0, b_0, \alpha) = p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma; \boldsymbol{X})\, p(\boldsymbol{\gamma}|\alpha)\, p(\boldsymbol{\beta}|\sigma; \Lambda_0)\, p(\sigma|a_0, b_0) \tag{1.1}$$

Further, given the regression

$$\boldsymbol{y} = \boldsymbol{X}(\boldsymbol{\gamma} \odot \boldsymbol{\beta}) + \varepsilon \tag{1.2}$$

$$\varepsilon \sim \mathcal{N}(\cdot; 0, \sigma) \quad \sigma \sim \Gamma^{-1}(a_0, b_0) \quad \boldsymbol{\beta} \sim \mathcal{N}(\cdot; \mathbf{0}, \sigma^2 \Lambda_0^{-1}) \quad \boldsymbol{\gamma} \sim \text{DPP}(\boldsymbol{\theta}) \text{ or Bernoulli}(\alpha)$$

We should be able to say that

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma; \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|\boldsymbol{X}_n(\boldsymbol{\gamma} \odot \boldsymbol{\beta}), \sigma) \tag{1.3}$$

Is it true that:

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma; \boldsymbol{X})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta})\, p(\boldsymbol{\beta}|\sigma; \Lambda_0)\, p(\sigma|a_0, b_0) \tag{1.4}$$

1.3.2 Notes from Greg 11/10/15

- Great conversation.
- Basic theoretical structure, as I presented it, is good. We're just using priors and marginals to calculate posteriors, and everything we do is either estimating integrals, or maximizing the posterior, basically.
- Yes, KL-divergence is just one method. Didn't get into too many specifics of other algorithms, but (relevant later) Bayesian approximation of integrals will be relevant to solving some parts of this problem.
- My understanding of the full marginal in equation 1.3 is correct. The distribution over $\boldsymbol{y}$ is just a bunch of Gaussians, since we're assuming some gaussian noise, ε.
- My understanding in equation 1.4 is incorrect as written, but needs only be integrated over the "extra" variables.
  - My logic had been that the probability of $\boldsymbol{y}$ given $\boldsymbol{\gamma}$ must be the probability of $\boldsymbol{y}$ given $\boldsymbol{\gamma}$ *and everything else that we need to determine* $\boldsymbol{y}$, so we must then multiply by the probability that we actually have those values.
  - Trouble is, they must be general, so we must sum over all possible values for them, since our original query, $p(\boldsymbol{y}|\boldsymbol{\gamma})$ is not dependent on anything else.

  - Thus, you *can* rewrite any marginal as dependent on many other variables, if you multiply by those variables' priors, and then integrate that product over all values of those variables.

- Thus, the correct expressions are:

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \iiint p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma, \boldsymbol{X})\, p(\boldsymbol{\beta}|\sigma; \Lambda_0)\, p(\sigma|a_0, b_0)\, d\boldsymbol{\beta}\, d\sigma \tag{1.5}$$

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}) \tag{1.6}$$

- Equation 1.5 should be a tractable, analytically solvable integral, given the gaussian and inverse gamma distributions of each component. Equation **??** however, will very likely need to be evaluated numerically.

- Greg didn't have any precise suggestions for a numerical method, but suggested I look into Bayesian approximations to integrals and choose the best method I can find. Before this though, take a look, and see if it can be solved analytically.

1.4 Meetings about approaching §2.4.1

1.4.1 Notes from Barbara 12/8/15

- Discussed the Taylor Expansion and Eigenvalue Methods for analytically solving $\int p(\boldsymbol{\gamma}|\boldsymbol{\theta})\, d\boldsymbol{\theta}$ from §2.4.1.

- She agreed that it looked likely intractable, but suggested the following approximations of the integral:

  - In the Taylor Expanded form, the integral takes on a sort of Gaussian-like form. Can we model this whole thing as a Guassian if we write the correct $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$?
  - Perhaps a Laplace transformation is then best?
  - Bianca has experience with Laplace transformations – chat with her.

- Alternatively, she said to solve it numerically, and that MC Integration was the best bet. However, this is also intractable unless I use a reasonable prior over $\boldsymbol{\theta}$, which is so far undefined for this problem. We've just always left it as $\boldsymbol{\theta} \in \mathbb{R}^M$ without specifying any distribution. Think about what a good distribution would be.

1.4.2 Notes from Barbara 12/18/15

- Very brief meeting. I presented my findings on treating the whole exponentiated form of θ as a quality term and arbitrarily defining some closed range, like $\theta_i \in [0, 1]$ for each q_i.

- Barbara seemed to think this was reasonable and not breaking any inference rules.

- I realize later that $\theta_i = 0$ yields $p(i \in \boldsymbol{Y} \subseteq \mathcal{Y}) = 0$ by properties of the determinant. Perhaps this is an interesting possibility, but since it is explicitly not allowed by the definition of $\boldsymbol{\theta}$, I will at least initially forbid it in the definition of $\boldsymbol{\theta^1}$.

- I will drop the parameter $\boldsymbol{\theta}$ from the current definition of L

$$L(\boldsymbol{\theta}) = \text{Diag}(e^{\boldsymbol{\theta}/2})\boldsymbol{\Phi}\boldsymbol{\Phi}^\top \text{Diag}(e^{\boldsymbol{\theta}/2}) \tag{1.7}$$

  And instead, define a new parameter, $\boldsymbol{\theta^1}$ with a relation to the old parameter

$$\boldsymbol{\theta^1} = \begin{cases} e^{\boldsymbol{\theta}/2} & \boldsymbol{\theta} \in (0,1]^M \\ \text{D.N.E.} & \text{o/w} \end{cases} \tag{1.8}$$

- In keeping with Bayesian Inference conventions, I want to use θ to represent my discovered parameterization, however, I do feel the need to include the superscript in order to avoid a notational nightmare in my notes.

1.5 Notes from Barbara 3/8/16

- My old approach had been to learn the MLE of all hyperparameters and $\boldsymbol{\theta}$. However, this requires a lot of integrating out over other things. No good. Instead, be much clearer with the model, and do variational inference.

- Basic idea is to do a handful of updates of each variable, either MLE or MAP (dependent on whether or not we have a prior available) and cycle through all of the variables until we reach some overall convergence.

- Great thing is that we're only concerned with the immediately relevant variables. I.e., updating $\boldsymbol{\gamma}$ is independent of a_0.

- Idea: Get rid of c. Just set $c = 1$ and eliminate it from all calculations.

- Instead of a real prior on $\boldsymbol{\gamma}$, just do some sort of l_0 norm.

Chapter 2

Problem Description and Notes

2.1 Parameterized DPPs

2.1.1 Motivation

Nearly every marginalized distribution has some parameter, be it $\mathcal{N}(\mu, \sigma^2)$, $\Gamma(a, b)$, $B(\alpha)$, etc. This is convenient, because it is significantly easier to infer a small finite number of parameters than it is an entire probability distribution, when solving the canonical

$$\theta_{\text{MLE}} = \arg\max_{\theta \in \mathbf{\Theta}} p(\mathcal{D}|\theta)$$

In this section, I will select a more specific parameterization to be used in my experiments, and then solve some of the typical probability calculations and algorithms for this particular parameterization.

2.1.2 Definitions

I will determine my DPP with an L-ensemble, defined in terms of some parameter θ. A typical choice for this definition would be the covariance of the observed data after some scaling of the features [2][ADD MORE CITATIONS]. Using the notation from [9], I will define my L-ensemble as the inner product of a matrix $B \in \mathbb{R}^{n \times p}$ with itself, where the columns B_i are $q_i \phi_i$, where $q_i \in [0, 1]$ is a quality scaling of the feature vector $\phi_i \in \mathbb{R}^n$.

$$L(\theta) = B(\theta)^\top B(\theta) = \exp(\theta/2) X^\top X \exp(\theta/2) \tag{2.1}$$

$$L \in \mathbb{R}^{p \times p}, \;\; \theta \in \mathbb{R}^p, \;\; X \in \mathbb{R}^{n \times p}, \;\; \boldsymbol{\gamma} \in \{0, 1\}^p \triangleq \mathcal{Y}$$

I will also use the following notation to indicate L indexed on particular rows and columns, indicated by $\boldsymbol{\gamma}$:

$$L(\theta, \boldsymbol{\gamma}) = L(\theta)_{\boldsymbol{\gamma}} \tag{2.2}$$

Thus, we have defined an L-ensemble in terms of the covariance of the feature-weighted data matrix. I will assume this form in all of the following particular solutions. For the purposes of these derivations, I will use the variable $\boldsymbol{\gamma} \in \{0, 1\}^p$ to represent the random variable drawn from the DPP. Note, the space of all possible draws was defined as $\mathcal{Y}$. Further, I will assume a uniform prior distribution on θ, which here will be the most general, allowing for any arbitrary weighting of the features, ϕ_i. Note, in all of the following derivations, I have an implicit condition on X, assuming that all data is known prior to any inference calculations.

2.1.3 Probabilities

If we draw $\boldsymbol{\gamma}$ from our DPP, determined by $L(\theta)$, we can define the conditional probability as follows [9].

$$p(\boldsymbol{\gamma}|\theta) \propto \det\left[L(\theta, \boldsymbol{\gamma})\right] \tag{2.3}$$

$$p(\boldsymbol{\gamma}|\theta) = \frac{\det\left[L(\theta, \boldsymbol{\gamma})\right]}{\det\left[L(\theta) + I_p\right]} \tag{2.4}$$

ADD MORE FORMULAE HERE, OR MOVE THE SECTION TO THE KULESZA, TASKAR 2013 REVIEW.

2.2 Linear Regression with Parameterized DPP Sparsity

2.2.1 Overview

When performing regression on datasets with $p \gg n$, the need for sparsity is clear [4] [2] [13][CITEMORE]. Here, I will solve the sparse linear regression problem by using a feature-selection binary vector, $\boldsymbol{\gamma} \in \mathcal{Y} \triangleq \{0, 1\}^p$ as done in [2]. However, I will assume that both the prior and posterior on $\boldsymbol{\gamma}$ are DPPs, as defined and discussed in §2.1.

As further motivation, in the Bornn paper [2], they assume a Bernoulli prior on an inclusion vector $\boldsymbol{\gamma}$, and use Variational Inference to learn the parameters $\boldsymbol{\theta}$ for a DPP posterior on $\boldsymbol{\gamma}$ that minimizes the KL-divergence. They then repeat the process assuming a DPP prior and a Bernoulli posterior. This *can* robustify sparsity, but it's not really committing to any claim about the true nature of $\boldsymbol{\gamma}$. I will assert that $\boldsymbol{\gamma}$ really has a DPP prior, treating this as standard inference.

2.2.2 Definitions of Terms

Taking our lead from Bornn, [2], we assume a noisy linear regression model with a binary feature selection variable. I assume that we are given the data and their labels in a dictionary, $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$. The parameters, their priors, and hyperparameters are given below.

$$\boldsymbol{y} = \boldsymbol{X}(\boldsymbol{\gamma} \odot \boldsymbol{\beta}) + \varepsilon \tag{2.5}$$

Variable Description: $\boldsymbol{X}$

The given data is set of regressors, $\{x_1, \ldots, x_n\}$, collected into a design matrix, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. No relevant priors or hyperparameters.

Variable Description: $\boldsymbol{y}$

Each datapoint is given a label, y_i, which are collected into the label vector $\boldsymbol{y} \in \mathbb{R}^n$. No relevant priors or hyperparameters.

Variable Description: $\boldsymbol{\beta}$

The regression coefficients are given by the vector $\boldsymbol{\beta} \in \mathbb{R}^p$. We assume that these coefficients are drawn from a zero-mean multivariate gaussian.

$$\boldsymbol{\beta} \sim \mathcal{N}(\cdot; 0, \sigma^2 \Lambda_0^{-1}) \tag{2.6}$$

The first part of the covariance, σ^2, is given two hyperparameters, a_0, b_0 and the inverse-Gamma prior distribution:

$$p(\sigma^2 | a_0, b_0) = \Gamma^{-1}(a_0, b_0) = \frac{b_0^{a_0}}{\Gamma(a_0)} (\sigma^2)^{-a_0 - 1} \exp\left(-\frac{b_0}{\sigma^2}\right) \tag{2.7}$$

The second part of the covariance, Λ_0 is given by $\Lambda_0 = c\boldsymbol{I}_p$, where c is a scalar hyperparameter and $\boldsymbol{I}_p$ is the identity matrix in $\mathbb{R}^{p \times p}$.

Variable Description: $\boldsymbol{\gamma}$

The variable selection vector $\boldsymbol{\gamma} \in \{0, 1\}^p$ is assumed to be drawn from a DPP, defined by the L-ensemble parameterized on θ, $L(\theta)$. Details of the priors and probabilities associated with $\boldsymbol{\gamma}$ are given in §2.1 and §2.4.1. Throughout these notes I will use the subscript $\boldsymbol{A}_{\boldsymbol{\gamma}} \in \mathbb{R}^{|\boldsymbol{\gamma}| \times |\boldsymbol{\gamma}|}$ to indicate the subset of matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ as indexed by $\boldsymbol{\gamma}$. The notation $\boldsymbol{v}_{\boldsymbol{\gamma}}$ can indicate either the vector $\boldsymbol{v} \odot \boldsymbol{\gamma} \in \mathbb{R}^n$ or it may refer to only those elements of $\boldsymbol{v}$ as indexed by $\boldsymbol{\gamma}$, concatenated into a new vector $\boldsymbol{v}_{\boldsymbol{\gamma}} \in \mathbb{R}^{|\boldsymbol{\gamma}|}$. Context should clarify the distinction between these cases.

Variable Description: ε

We add noise to the regression with $\varepsilon \sim \mathcal{N}(\cdot; 0, \sigma^2)$. 160219 Note: Bornn actually writes the variance for ε as "σ" [2]. I believe this is simply a typo, because making the variance on the noise the same as the coefficient on the covariance of $\boldsymbol{\beta}$ drastically simplifies the math, while including the hyperparameter c maintains full generality. Further, given Bornn's simple definition, the dimensionality of the regression does not work, so we take ϵ as the noise on each observation, thus performing n draws from the 1-Dimensional Gaussian described here.

Bayesian Network:

Thus we arrive at the full model, shown in figure 2.1. Circled variables of interest that we must learn or estimate. Greyed variables indicate known data. Non-circled variables are either hyperparameters or values we will not be estimating. Light arrows indicate paths of influence not yet affected by variables of interest. Heavy arrows indicate paths of influence originating from a variables of interest. In the following sections, we will marginalize out parts of this network in an effort to reduce the complexity of our problem. Also, we will add hyperparameters as variables of interest in order to find MLEs of them.

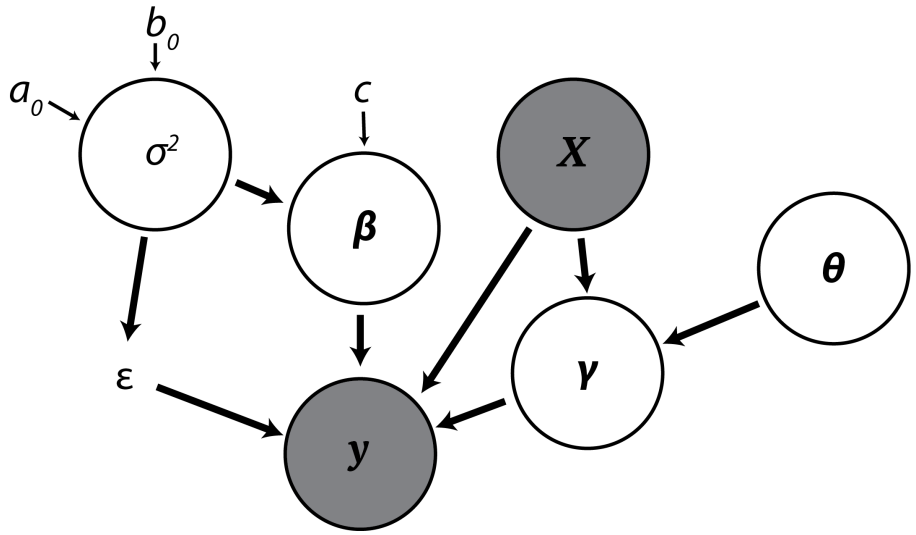

Figure 2.1: The full Bayesian Network for Linear Regression with DPP-sparsity.

2.2.3 Problem Description

We are posing this as a linear regression problem. Thus, there is no need to infer a complete joint distribution of all of our parameters ($\boldsymbol{\beta}$, $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$, σ^2). In fact, if possible, we don't even need point estimates of our parameters as long as we can make predictions $\hat{y}$ for new data points $\boldsymbol{x} \in \mathbb{R}^p$.

In fact, as I show in §3.2.4 and §3.3.7, I need only a MAP estimate of $\boldsymbol{\gamma}$ in order to make predictions.

Full Joint Distribution

I did look into solving for the full joint distribution (I abandoned that course) but the problem description is as follows.

To estimate $\boldsymbol{\theta}$, the DPP parameters of $\boldsymbol{\gamma}$, we can use the MLE, which is found by solving the following problem.

$$\boldsymbol{\theta}^{\star}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{y}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \left(\sum_{\boldsymbol{\gamma}\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}) \right) \tag{2.8}$$

Then, to learn the optimal sparsity vector in the regression problem, we can solve the MAP problem:

$$\boldsymbol{\gamma}^{\star}_{\text{MAP}} = \arg\max_{\boldsymbol{\gamma}\in\{0,1\}^p} p\left(\boldsymbol{\gamma}|\boldsymbol{y}, \boldsymbol{\theta}^{\star}_{\text{MLE}}\right) = \arg\max_{\boldsymbol{\gamma}} p(\boldsymbol{y}|\boldsymbol{\gamma})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}^{\star}_{\text{MLE}}) \tag{2.9}$$

2.3 Marginal and Conditional Distributions

2.3.1 Conditional Distribution of $\boldsymbol{\beta}$ Given σ^2

As discussed in §2.2, we take $\boldsymbol{\beta}$ to be distributed normally. Given σ^2 and the hyper-parameter c (coefficient in Λ_0, defined in §2.2), we found the value of the multivariate normal density function in equation A.5, reproduced here:

$$p(\boldsymbol{\beta}|\sigma^2, \Lambda_0) = \mathcal{N}\left(\boldsymbol{\beta}; \mathbf{0}, \frac{\sigma^2}{c}\boldsymbol{I}_p\right) = \left(\frac{c}{2\pi\sigma^2}\right)^{\frac{p}{2}} \exp\left[\frac{-1}{\sigma^2}\left(\frac{c\boldsymbol{\beta}^\top\boldsymbol{\beta}}{2}\right)\right] \tag{2.10}$$

2.3.2 Full Marginal Distribution of $\boldsymbol{\beta}$

As discussed in §2.2.2, $\boldsymbol{\beta}$ has a p-dimensional gaussian prior, and σ^2 has an inverse gamma prior. Thus, the marginal distribution for $\boldsymbol{\beta}$ is given by the following.

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \int_{\mathbb{R}^+} p(\boldsymbol{\beta}|\sigma^2, c)p(\sigma^2|a_0, b_0)\, d\sigma^2 \tag{2.11}$$

This probability is given in closed form by

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \left(\frac{c}{2\pi}\right)^{p/2} \frac{b_0^{a_0}}{\Gamma(a_0)} \left(\frac{c}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta} + b_0\right)^{a_0+p/2} \Gamma(-a_0 - p/2) \tag{2.12}$$

Where we have the following constraints:

$$a_0, b_0, c > 0 \qquad \frac{\lceil a_0 + p/2 - 1\rceil}{2} \equiv 0 \mod 2 \qquad a_0 + p/2 \notin \mathbb{Z} \tag{2.13}$$

$$c^{p/2} b_0^{a_0} \left(\frac{c}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta} + b_0\right)^{a_0+p/2} \Gamma\left(\frac{\lceil a_0 + p/2\rceil}{a_0 + p/2}\right) \leq (2\pi)^{p/2}\, \Gamma(a_0) \prod_{i=0}^{\lceil a_0+p/2-1\rceil} (a_0 + p/2 - i) \tag{2.14}$$

Details of the derivation for both the closed form expression and the constraints are provided in §A.1.

2.3.3 Full Conditional Distribution of $\boldsymbol{y}$:

Given that each y_i is a linear function of $\boldsymbol{x}_i$ with normally distributed, noise, each y_i is distributed normally, and the conditional probability of the full vector $\boldsymbol{y}$ is given by the product of each elemental gaussian.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) = \prod_{i=1}^{n} \mathcal{N}\left(y_i; x_i^\top(\boldsymbol{\gamma} \odot \boldsymbol{\beta}), \sigma^2\right) \tag{2.15}$$

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{y}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\beta}}{2\sigma^2}\right) \tag{2.16}$$

Details connecting these steps are presented in §A.2.1.

2.3.4 Conditional Distribution of $p(\boldsymbol{y}|\sigma^2, \boldsymbol{\gamma})$ Marginalizing $\boldsymbol{\beta}$

As detailed in §A.2, marginalizing out $\boldsymbol{\beta}$ gives us the following conditional distribution of $\boldsymbol{y}$.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2) = \frac{c^{p/2} \exp\left(\frac{-1}{2\sigma^2} \boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}\right)}{(2\pi\sigma^2)^{n/2} \sqrt{\det\left[\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right]}} \tag{2.17}$$

2.3.5 Conditional Distribution of $p(\boldsymbol{y}|\boldsymbol{\gamma})$ Marginalizing $\boldsymbol{\beta}$ and σ^2

For part of the problem of solving for the optimal sparsity vector, $\boldsymbol{\gamma}$, we are interested in finding a simple expression for $p(\boldsymbol{y}|\boldsymbol{\gamma})$. Given my discussion with Greg §1.3.2, I set up the following integral, which itself includes a number of conditional probabilities.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \iint p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X})\, p(\boldsymbol{\beta}|\sigma^2; \Lambda_0)\, p(\sigma^2|a_0, b_0)\, d\boldsymbol{\beta}\, d\sigma^2 \tag{2.18}$$

Combining the distributions given in equations 2.16 and 2.10 with the prior on σ^2, we find a full expression for the integrand in equation 2.18. Solving, we find the closed form solution:

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \frac{b_0^{a_0} c^{p/2} \Gamma\left(\frac{n}{2} + a_0\right)}{\Gamma(a_0)(2\pi)^{n/2} \sqrt{\left|\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right|} \cdot \left(b_0 + \frac{1}{2} \boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}\right)^{a_0 + n/2}} \tag{2.19}$$

The derivation of the solution is provided in §A.3.1. As part of the solution, we infer the following constraints:

$$b_0 \geq \left(\frac{\lambda_{\max} - 1}{2}\right) \boldsymbol{y}^\top \boldsymbol{y} \tag{2.20}$$

$$c \geq 1 \tag{2.21}$$

Where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}^\top$. A discussion of this constraint, its validity, and a computational trick are also given in §A.3

2.4 Prior Distributions

2.4.1 Analytic Attempts at Solving for $\boldsymbol{\gamma}$

In the linear regression setting, §2.2, we estimate the DPP parameters $\boldsymbol{\theta}$ with an MLE, and then use that to obtain the MAP estimate of $\boldsymbol{\gamma}^\star$, using $p(\boldsymbol{\gamma}|\boldsymbol{\theta}_{\text{MLE}})$ as the prior on $\boldsymbol{\gamma}$. In some prior work, I was attempting to avoid estimating $\boldsymbol{\theta}$, and instead use the prior $p(\boldsymbol{\gamma}) = \int p(\boldsymbol{\gamma}|\boldsymbol{\theta})\, d\boldsymbol{\theta}$. This integral is intractable, but I did some work on it. My notes on attempts and methods of solving this integral are below.

Integral over $\boldsymbol{\theta}$:

We have only the following term, and its value in terms of the DPP $L_{\boldsymbol{\theta}}$ as given by [9].

$$p(\boldsymbol{\gamma}|\boldsymbol{\theta}) = p_L(\boldsymbol{\gamma}) = \frac{\det\left(L_{\boldsymbol{\gamma}}\right)}{\det\left(L+I\right)} \tag{2.22}$$

We know that $\det\left(\boldsymbol{AB}\right) = \det\left(\boldsymbol{A}\right)\cdot\det\left(\boldsymbol{B}\right)$ and that $\det\left(e^{\boldsymbol{A}}\right) = e^{\mathrm{tr}(\boldsymbol{A})}$. Thus, we find:

$$\det\left(L_{\boldsymbol{\gamma}}\right) = e^{\mathrm{tr}(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2)}\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\top}\boldsymbol{X}_{\boldsymbol{\gamma}}\right)e^{\mathrm{tr}(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2)} \tag{2.23}$$

Which could be integrated nicely:

$$\int \det\left(L_{\boldsymbol{\gamma}}\right)\,d\boldsymbol{\theta} = \det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\top}\boldsymbol{X}_{\boldsymbol{\gamma}}\right)\int\prod_{i\in\boldsymbol{\gamma}}e^{\theta_i}\,d\theta_i \tag{2.24}$$

However, the pesky denominator creates a much more substantive problem. I have tried a few different ways of approaching this integral, but have found no way to write $\det\left(L+I\right)$ in terms of simply $\det\left(L\right)$. Here are my notes for later reference, but ultimately, I will have to pursue numeric solutions.

Taylor Expansion Methods:

If we use the Taylor expansion of the determinant, we can "reduce" the denominator.

$$\det\left(L(\boldsymbol{\theta})+I\right) = \exp\left(\mathrm{tr}\left(\log\left(L(\boldsymbol{\theta})+I\right)\right)\right) = \exp\left(\mathrm{tr}\left(-\sum_{i=1}^{\infty}\frac{(-1)^i}{i}L(\boldsymbol{\theta})^i\right)\right) \tag{2.25}$$

Because trace is a linear operation we can bring it within the sum. Thus we have the full integral as

$$\int\frac{\det\left(L(\boldsymbol{\theta})_{\boldsymbol{\gamma}}\right)}{\det\left(L(\boldsymbol{\theta})+I\right)} = \det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\top}\boldsymbol{X}_{\boldsymbol{\gamma}}\right)\int\prod_{i\in\boldsymbol{\gamma}}e^{\theta_i}\exp\left(\sum_{i=1}^{\infty}\frac{(-1)^i}{i}\,\mathrm{tr}\left[L(\boldsymbol{\theta})^i\right]\right)\,d\boldsymbol{\theta} \tag{2.26}$$

However, here, we would still need to compute many powers of L, which cannot be done while separating $\boldsymbol{\theta}$ terms and $\boldsymbol{X}$ terms. Thus, we are computationally, no better off.

Eigenvalue Methods:

We can instead use the eigenvalue representation of a determinant:

$$\det\left(L(\boldsymbol{\theta})\right) = \prod_i\lambda_i \;\Rightarrow\; \det\left(L(\boldsymbol{\theta})+I\right) = \prod_i(\lambda_i+1) \tag{2.27}$$

This requires finding eigenvalues for all $\boldsymbol{\theta}$ of

$$L(\boldsymbol{\theta}) = \mathrm{Diag}\left(e^{\boldsymbol{\theta}/2}\right)\boldsymbol{X}^{\top}\boldsymbol{X}\,\mathrm{Diag}\left(e^{\boldsymbol{\theta}/2}\right) \tag{2.28}$$

For which, there is no separable, analytic solution.

Numerical Estimation of $\boldsymbol{\theta}$ conditionals

First let us comment on some notational differences between Bornn [2] and Kulesza and Taskar [9]. First, in Bornn [2], we have

$$L = \text{Diag}\left(e^{\boldsymbol{\theta}/2}\right) \boldsymbol{\Phi}\boldsymbol{\Phi}^\top \text{Diag}\left(e^{\boldsymbol{\theta}/2}\right), \ \boldsymbol{\Phi} \in \mathbb{R}^{M\times d} \tag{2.29}$$

And they describe $\boldsymbol{\Phi}$ as "a matrix of similarity features whose row m, $\phi(m)$ is the similarity feature vector for item m. They then suggest $\boldsymbol{\Phi} = \boldsymbol{X}$. However, we know from the problem statement that $\boldsymbol{X}$ has features as columns, not rows. Thus, I believe that this is a type, and Bornn intended to say that $\boldsymbol{\Phi} = \boldsymbol{X}^\top$. In this case, we have the following description for L:

$$L = \text{Diag}\left(e^{\boldsymbol{\theta}/2}\right) \boldsymbol{X}^\top\boldsymbol{X} \text{Diag}\left(e^{\boldsymbol{\theta}/2}\right) = \begin{bmatrix} e^{\theta_1/2}\boldsymbol{x}_1^\top \\ e^{\theta_2/2}\boldsymbol{x}_2^\top \\ \vdots \\ e^{\theta_M/2}\boldsymbol{x}_M^\top \end{bmatrix} \cdot \begin{bmatrix} e^{\theta_1/2}\boldsymbol{x}_1 & e^{\theta_2/2}\boldsymbol{x}_2 & \dots & e^{\theta_M/2}\boldsymbol{x}_M \end{bmatrix} \tag{2.30}$$

Which, if we define $B \triangleq \begin{bmatrix} e^{\theta_1/2}\boldsymbol{x}_1 & e^{\theta_2/2}\boldsymbol{x}_2 & \dots & e^{\theta_M/2}\boldsymbol{x}_M \end{bmatrix}$ we can rewrite as

$$L = B^\top B \qquad B_i = q_i\phi_i \tag{2.31}$$

As Kulesza and Taskar define it in section 3.1 of their overview paper, treating q_i as a scalar "quality" term and ϕ_i as a vector "diversity feature" and $\phi_i^\top\phi_j$ as a signed similarity measure of items i and j [9]. They define a shorthand for similarity of

$$S_{ij} \triangleq \phi_i^\top\phi_j = \frac{L_{ij}}{\sqrt{L_{ii}L_{jj}}} \tag{2.32}$$

Thus in this problem, we are in effect integrating over quality values. Barbara thus approved reformulating this problem with a new parameterization, $\boldsymbol{\theta^1}$, were each $\boldsymbol{\theta}_i^{\mathbf{1}}$ is exactly the quality term, and is constrained to be in the range $(0, 1]$. (See §1.4.2).

Ideas for priors on $\boldsymbol{\gamma}$

- Uniform
- l_0 norm
- Some Gamma shaped prior, placing the mode above zero.

2.4.2 Ideas for Priors on $\boldsymbol{\theta}$

The parameterization vector, $\boldsymbol{\theta} \in \mathbb{R}^p$, serves to scale the ijth element of the inner product $\boldsymbol{X}^\top\boldsymbol{X}$ by $\exp\left((\boldsymbol{\theta}_i + \boldsymbol{\theta}_j)/2\right)$. The inner product matrix should already capture the similarity between features, which is the real measure by which samples are drawn from the DPP. Thus, $\boldsymbol{\theta}$ serves as a mechanism to adjust the relative similarities captured by $\boldsymbol{X}^\top\boldsymbol{X}$ to better reflect the information propagated from the rest of the network (i.e. $\boldsymbol{y}$). We do not necessarily want to restrict the ability of $\boldsymbol{\theta}$ to re-scale the relationships in $\boldsymbol{X}^\top\boldsymbol{X}$, but at the same time, we want L to actually reflect the structure of $\boldsymbol{X}^\top\boldsymbol{X}$. Thus, the proposed priors below will try to balance these two concerns.

Uniform Prior

The most simple (and perhaps obvious) prior we can place on $\boldsymbol{\theta}$ is the uniform prior. This very explicitly gives no prior preference to any values of θ_i. However, it can allow for values of θ_i to become arbitrarily large or small.

Other PDF Priors

While it may seem as if something more complex is necessary in order to properly balance large and small coefficients, note that our distribution for $\boldsymbol{\theta}$ will be equivalent to the distribution of $\exp(\boldsymbol{\theta})$ in log-space, which is really what we care about; we are interested in how *scaling* factors are distributed. To that end, we merely consider symmetric distributions that will balance compounding and fractional factors. The two priors that come to mind are the Gaussian and Laplace distributions, with the primary difference being how tightly centered they are around $\boldsymbol{\theta}_i = 0$, corresponding to a factor of $\exp(\boldsymbol{\theta}_i/2) = 1$.

Gaussian Prior:

$$\mathcal{L}(\boldsymbol{\theta}) = \log\left(\mathcal{N}(\boldsymbol{\theta}; \mathbf{0}_p, \boldsymbol{I}_p)\right) \propto -\frac{1}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} \tag{2.33}$$

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{\theta} \tag{2.34}$$

Laplace Prior:

$$\mathcal{L}(\boldsymbol{\theta}) = \log\prod_i f(\boldsymbol{\theta}_i;, 0, 1) \propto -\sum_i |\boldsymbol{\theta}_i| \tag{2.35}$$

$$\nabla\mathcal{L}(\boldsymbol{\theta})i) = -\operatorname{sign}(\boldsymbol{\theta}_i) \tag{2.36}$$

Expectation Prior:

$$\mathcal{L}(\boldsymbol{\theta}) = \log\operatorname{tr}(\exp(\boldsymbol{\theta}/2)\boldsymbol{I}_p\exp(\boldsymbol{\theta}/2)) = \log\left(\sum_i \frac{\boldsymbol{\theta}_i^2}{\boldsymbol{\theta}_i^2+1}\right) \tag{2.37}$$

$$\nabla\mathcal{L}(\boldsymbol{\theta})_i = \left(\frac{1}{\sum_j \frac{\boldsymbol{\theta}_j^2}{\boldsymbol{\theta}_j^2+1}}\right)\frac{2\boldsymbol{\theta}_i}{(\boldsymbol{\theta}_i^2+1)^2} \tag{2.38}$$

2.4.3 Uninformative Priors on Hyperparameters

In the case that I decide not to set a_0, b_0, and c, but instead attempt to optimize them as well, with some sort of MLE, it will benefit me to use an uninformative prior on each of them. The obvious approach, implemented by both Bayes and Laplace, is the uniform distribution [6]. However, this is not optimal, and may yield disproportionately many poor results. Using a Jeffreys prior is a way of *actually* asserting no prior knowledge on the hyperparameters [15].

Prior Distribution for a_0 and b_0

The hyperparameters a_0 and b_0 give the shape and scale parameters of the inverse gamma prior on σ^2. The Jeffreys prior on these parameters is given by the following [15].

$$\pi\left(a_0, b_0\right) = b_0 \sqrt{a_0 \psi^{(1)}\left(a_0\right) - 1} \tag{2.39}$$

Where $\psi^{(1)}$ is the polygamma function of the first order. However, we will be performing these calculations iteratively, and thus require not the joint probability of a_0 and b_0, but rather their conditional probabilities. We learn from [15] that in the setting where a_0 is known, the Jeffreys prior on b_0 is simply itself, which is to say that they are independent

$$\pi(b_0|a_0) = \pi(b_0) = b_0 \tag{2.40}$$

Using these two relationships, we can also find the conditional prior for a_0.

$$\pi(a_0|b_0) = \frac{\pi(a_0, b_0)}{\pi(b_0)} = \sqrt{a_0 \psi^{(1)}(a_0) - 1} = \pi(a_0) \tag{2.41}$$

Again, confirming their independence.

Prior Distribution for c

In our setting, c defines the ratio of the coefficients on the noise variance and the regression coefficient covariance constant. While this does not have a standard Jeffreys prior, we do know the prior for the full covariance. Since we will be computing this in an iterative setting, we can treat σ^2 as known. Thus, in effect, we can say

$$\pi(c = c_t) = \pi(\Sigma = \sigma_t^2 \boldsymbol{I}_p / c_t) \tag{2.42}$$

Since the regression coefficients are normally distributed and $\mu = \mathbf{0}$ is known, we can give the prior with the following [15].

$$\pi\left(\Sigma\right) = \frac{1}{\det(\Sigma)^{\frac{p+1}{2}}} \tag{2.43}$$

Which, in terms of c, is

$$\pi\left(c\right) = \left(\frac{c}{\sigma^2}\right)^{p\frac{p+1}{2}} \tag{2.44}$$

Of course however, this is dependent on σ^2. In all likelihood, if we are concerned with the uninformative prior on c, we have already marginalized over σ^2. Let us do the same for this distribution.

$$\pi\left(c|a_0, b_0\right) = \int_{\mathbb{R}^+} \pi\left(c|\sigma^2\right)\, p\left(\sigma^2|a_0, b_0\right)\, d\sigma^2 \tag{2.45}$$

Using the inverse gamma distribution on σ^2 and letting $k \triangleq p(p+1)/2$, we find

$$\pi\left(c|a_0, b_0\right) = \int_{\mathbb{R}^+} \left(\frac{c}{\sigma^2}\right)^k \frac{b_0^{a_0}}{\Gamma\left(a_0\right)} \left(\frac{1}{\sigma^2}\right)^{a_0+1} \exp\left(-\frac{b_0}{\sigma^2}\right) d\sigma^2 \tag{2.46}$$

Defining the following constant, we rewrite the integral as an inverse gamma distribution with a new shape parameter $a_0' = a_0 + k$.

$$\alpha \triangleq \frac{\Gamma(a_0 + k)}{\Gamma(a_0)\, b_0^k} \tag{2.47}$$

$$\pi(c|a_0, b_0) = \alpha c^k \int_{\mathbb{R}^+} \frac{b_0^{a_0+k}}{\Gamma(a_0 + k)} \left(\frac{1}{\sigma^2}\right)^{a_0+k+1} \exp\left(-\frac{b_0}{\sigma^2}\right) d\sigma^2 \tag{2.48}$$

The integral of $\Gamma^{-1}(\sigma^2; a_0 + k, b_0)$ is simply 1. Substituting back in our expressions for k and α, we arrive at the following.

$$\pi(c|a_0, b_0) = \left(\frac{c}{b_0}\right)^{p\frac{p+1}{2}} \frac{\Gamma\left(a_0 + p\frac{p+1}{2}\right)}{\Gamma(a_0)} \tag{2.49}$$

Chapter 3

Solutions to the DPP-Sparse Linear Regression Problem

3.1 An Ordered Systematic Approach

The point of linear regression is to create a model of some given data in the hope that it can serve as a predictor for future data. The following steps are all in relation to model creation. They should be applied with standard machine learning practices, with respect to cross-validation and the like.

My first approach is to build a predictor using the following three sequential steps:

1. Estimate hyperparameters $\boldsymbol{\theta}, a_0, b_0, c$ with the MLE by maximizing the type II likelihood:

$$\boldsymbol{\theta}^{\text{MLE}} = \arg\max_{\boldsymbol{\theta}\in\Theta} \mathcal{P}\left(\boldsymbol{y}|\boldsymbol{\theta}\right) \tag{3.1}$$

$$a_0^{\text{MLE}} = \arg\max_{a_0\in\mathbb{R}^+} \mathcal{P}\left(\boldsymbol{y}|a_0\right) \tag{3.2}$$

$$b_0^{\text{MLE}} = \arg\max_{b_0\in\mathbb{R}^+} \mathcal{P}\left(\boldsymbol{y}|b_0\right) \tag{3.3}$$

$$c^{\text{MLE}} = \arg\max_{c\in\mathbb{R}^+} \mathcal{P}\left(\boldsymbol{y}|c\right) \tag{3.4}$$

2. Select the MAP estimate of $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma}^{\text{MAP}} = \arg\max_{\boldsymbol{\gamma}\in\{0,1\}^p} \mathcal{P}\left(\boldsymbol{\gamma}|\boldsymbol{y}; \boldsymbol{\theta}^{\text{MLE}}, a_0^{\text{MLE}}, b_0^{\text{MLE}}, c^{\text{MLE}}\right) \tag{3.5}$$

3. Choose $\boldsymbol{\beta}$ as its expected value:

$$\boldsymbol{\beta}^{\star} = \mathbb{E}\left[\boldsymbol{\beta}|\boldsymbol{\gamma}^{\text{MAP}}, a_0^{\text{MLE}}, b_0^{\text{MLE}}, c^{\text{MLE}}\right] \tag{3.6}$$

With these results, we can build the predictor:

$$\hat{y} = x^{\top}\left(\boldsymbol{\gamma}^{\text{MAP}} \odot \boldsymbol{\beta}^{\star}\right) \tag{3.7}$$

3.1.1 Computation of $\boldsymbol{\theta}$ Maximum Likelihood Estimation, Marginalizing $\boldsymbol{\gamma}$

As discussed in §3.1, we must calculate the maximum likelihood estimation of all hyperparameters. For the DPP-parameterization, $\boldsymbol{\theta}$, that equation looks like this:

$$\boldsymbol{\theta}^{\text{MLE}} = \arg\max_{\boldsymbol{\theta}\in\Theta} \mathcal{P}\left(\boldsymbol{y}|\boldsymbol{\theta}\right) \tag{3.8}$$

Since $\boldsymbol{y}$ is not directly dependent on $\boldsymbol{\theta}$, we can write this in a more meaningful way as

$$\boldsymbol{\theta}^{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \left(\sum_{\boldsymbol{\gamma}\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}) \right) \tag{3.9}$$

Where the expression for $p(\boldsymbol{y}|\boldsymbol{\gamma})$ is given by equation 2.19, and $p(\boldsymbol{\gamma}|\boldsymbol{\theta})$ is simply the standard DPP probability expression, given in equation 2.4. Noting that $p(\boldsymbol{y}|\boldsymbol{\gamma})$ is independent of $\boldsymbol{\theta}$, we can use the result from §6.6.3, equation 6.56, to find a simple algorithm to solve equation 3.9.

We note that in this setting, $\boldsymbol{f}_i(X) = e_i$ where e_i is the ith elementary vector, that their variable Y is our variable $\boldsymbol{\gamma}$, and that their matrix $S(X)$ is our $\boldsymbol{X}^\top\boldsymbol{X}$. Thus, we can make the following reduction.

$$\nabla\left(\log\mathcal{P}(\boldsymbol{\gamma}|\boldsymbol{\theta})\right) = \sum_{i\in\boldsymbol{\gamma}} \boldsymbol{f}_i(\boldsymbol{X}) - \sum_i K_{ii}\boldsymbol{f}_i(\boldsymbol{X}) = \boldsymbol{\gamma} - \text{diag}(K) \tag{3.10}$$

Thus, for our problem, we can define the following likelihood function and gradient.

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}|\boldsymbol{\theta}) \qquad \nabla\mathcal{L}(\boldsymbol{\theta}) = \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma})\cdot\nabla p(\boldsymbol{\gamma}|\boldsymbol{\theta}) \tag{3.11}$$

Recall the following.

$$\nabla f(x) = f(x)\cdot\nabla\left[\log(f(x)\right] \tag{3.12}$$

Using the identity in equation 3.12, we can substitute the result from equation 3.10 into the gradient in equation 3.11. Thus, we obtain the closed form expression for the gradient of the likelihood.

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}|\boldsymbol{\theta})\left(\boldsymbol{\gamma} - \text{diag}(K)\right) \tag{3.13}$$

3.1.2 A word on concavity

Kulesza and Taskar show that $\log\mathcal{P}(\boldsymbol{\gamma}|\boldsymbol{\theta})$ is concave [9] page 48. Thus $\mathcal{P}(\boldsymbol{\gamma}|\boldsymbol{\theta})$ is log-concave. However, the sum of log-concave functions is not necessarily log-concave. This arises from the fact that while $\log f(x)$ being convex implies that $f(x)$ is convex, $\log g(x)$ being concave does not imply that $g(x)$ is concave. Therefore, we cannot be certain that the likelihood in equation 3.11 is actually a concave function. Thus, when using gradient descent, it will be prudent to run the algorithm a number of times with different, random initializations.

Irrespective of the concavity, the full algorithm for computing the gradient of the log-likelihood is given in §**??**.

3.1.3 Problems with this approach

Given the solutions we have to conditional distributions like 2.18, our model is now only dependent on hyperparameters, $\boldsymbol{\gamma}$, which we want to learn, and $\boldsymbol{\theta}$. Given the $\vee$ structure of our model and the determinantal form of $\boldsymbol{\gamma}$'s distribution, marginalizing out these other parameters is all but impossible. Therefore, the above MLE estimates will *have* to be iterative. Thus, (inspired by my discussion with Barbara §1.5), it is far simpler to use an approach as outlined in §3.3 and §3.2.

3.2 A Variational Inference Approach: Partial Pre-Marginalization

Using results from §2.3, we can marginalize out $\boldsymbol{\beta}$ from our model. This yields a network much like the one given in figure 3.1. In §3.2.1, §3.2.2, and §3.2.3, I give likelihood functions and their gradients for each of the variables of interest, or those circled in the network in figure 3.1. Note, greyed variables are known data. Then, using variational inference, I will arrive at a point estimate of the optimal values for all of the variables of interest. Using those estimates, I will create a predictor that can serve to yield estimates, $\hat{y}$, for new data points, $x \in \mathbb{R}^p$. These results, presented in §3.2.4 actually indicate that $\boldsymbol{\gamma}$ is the only variable necessary for making predictions.

In this section, I operate under the assumption that the likelihood functions can be any function $\mathcal{L}(\phi)$ for general variable ϕ, such that the following holds.

$$\arg\max_{\phi \in \boldsymbol{\Phi}} \mathcal{P}(\phi | \operatorname{MB}(\phi)) = \arg\max_{\phi \in \boldsymbol{\Phi}} \mathcal{L}(\phi) \tag{3.14}$$

Where $\operatorname{MB}(\phi)$ is the Markov Blanket of ϕ in the network. Note that in this setting, which I call the "Partial Pre-Marginalization" setting, I use a fixed a_0, b_0, and c as hyperparameters. I will infer values for σ^2, $\boldsymbol{\gamma}$, and $\boldsymbol{\theta}$, thus, using only the marginalization of $\boldsymbol{\beta}$. Details of my derivations for the following likelihoods and gradients are presented in §B.1. The results are summarized here.

3.2.1 Likelihood and Gradient of σ^2

Both are defined for $\sigma^2 \in \mathbb{R}^+$ and defined in §B.1.1

$$\mathcal{L}(\sigma^2) = -\frac{k}{\sigma^2} - (a_0 + 1 + n/2) \log\left(\sigma^2\right) \tag{3.15}$$

$$\nabla\mathcal{L}(\sigma^2) = \frac{k}{\left(\sigma^2\right)^2} - \frac{a_0 + 1 + n/2}{\sigma^2} \tag{3.16}$$

Where we have defined the constant

$$k = b_0 + \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y} \tag{3.17}$$

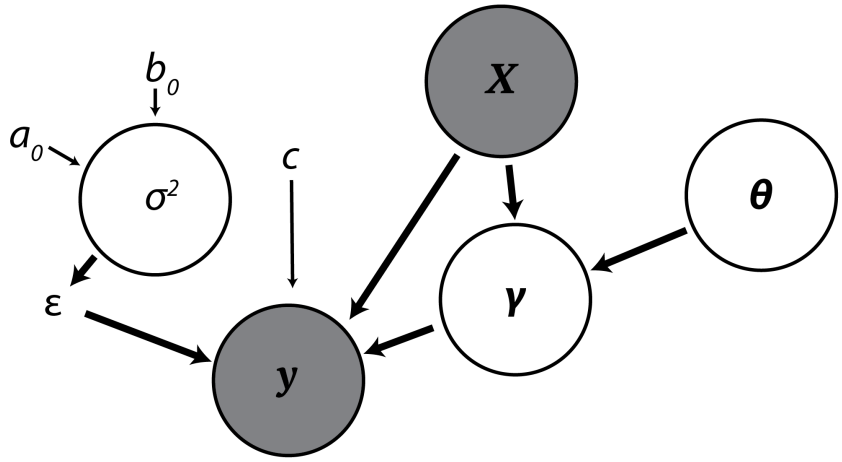

Figure 3.1: The reduced Bayesian Network for Linear Regression with DPP-sparsity with marginalized regression coefficients, $\boldsymbol{\beta}$.

3.2.2 Likelihood and Gradient of $\boldsymbol{\theta}$

Both are defined for $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$, where Θ, and any prior for $\boldsymbol{\theta}$ over Θ is left up to the user at the time of operation. In this setting, the markov blanket of $\boldsymbol{\theta}$ is identical to that in §3.3, so we repeat the results from §B.1.3.

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i \in \boldsymbol{\gamma}} \boldsymbol{\theta}_i - \log\left(\det\left[\text{Diag}\left(e^{\boldsymbol{\theta}/2}\right) \boldsymbol{X}^\top \boldsymbol{X} \,\text{Diag}\left(e^{\boldsymbol{\theta}/2}\right)\right]\right) + \log p\left(\boldsymbol{\theta}|\boldsymbol{X}\right) \tag{3.18}$$

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\gamma} - \text{diag}(K) + \nabla \log p(\boldsymbol{\theta}|\boldsymbol{X}) \tag{3.19}$$

3.2.3 Likelihood and Optimization of $\boldsymbol{\gamma}$

Much like the optimization of $\boldsymbol{\gamma}$ for the maximal pre-marginalization experiment, we will need to use an approximation algorithm here. The likelihood function though is a little different. Its derivation is given in §B.1.2. For now, I will use the same greedy algorithm given in algorithm 1.

$$\mathcal{L}(\boldsymbol{\gamma}) = \frac{-1}{2\sigma^2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y} - \frac{1}{2}\log\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right) + \sum_{i \in \boldsymbol{\gamma}} \boldsymbol{\theta}_i + \log\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right) \tag{3.20}$$

3.2.4 Making Predictions

Here we consider the predictive distribution. For a single new datapoint $x \in \mathbb{R}^p$, we are interested in predicting its label, $\hat{y} \in \mathbb{R}$. This can be given by the following equation, where we assume $\boldsymbol{\gamma}$, $\boldsymbol{\theta}$, and σ^2 have already been optimized.

$$\hat{y} = \arg\max_{y \in \mathbb{R}} \mathcal{P}\left(y|x, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \sigma^2\right) \tag{3.21}$$

$$\mathcal{P}\left(y|x,\boldsymbol{y},\boldsymbol{X},\boldsymbol{\gamma},\boldsymbol{\theta},\sigma^2\right)=\int_{\mathbb{R}^p}\mathcal{P}\left(y|x,\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2\right)\mathcal{P}\left(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\gamma},\sigma^2\right)d\boldsymbol{\beta} \tag{3.22}$$

Following the derivation in [3], we see that the result is the convolution of two gaussians. First, we know that

$$\mathcal{P}\left(y|x,\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2\right)=\mathcal{N}\left(y|x_{\boldsymbol{\gamma}}\left(\boldsymbol{\beta}\odot\boldsymbol{\gamma}\right),\sigma^2\right) \tag{3.23}$$

Where $x_{\boldsymbol{\gamma}}\left(\boldsymbol{\beta}\odot\boldsymbol{\gamma}\right)$ is the target variable that would be predicted by the original regression problem (equation 2.5). We also know from [3] that the distribution on $\boldsymbol{\beta}$ is given by

$$\mathcal{P}\left(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\gamma},\sigma^2\right)=\mathcal{N}\left(\boldsymbol{\beta}|\mathbf{m_N},\mathbf{S_N}\right) \tag{3.24}$$

$$\mathbf{m_N}\triangleq\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}+c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{y}\qquad\mathbf{S_N}\triangleq\sigma^2\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}+c\boldsymbol{I}_p\right)^{-1} \tag{3.25}$$

The convolution of these gaussians is then given by

$$\mathcal{P}\left(y|x,\boldsymbol{y},\boldsymbol{X},\boldsymbol{\gamma},\boldsymbol{\theta},\sigma^2\right)=\mathcal{N}\left(y|\mathbf{m_N}^\top x_{\boldsymbol{\gamma}},\boldsymbol{\sigma}_{\mathbf{N}}^2(x)\right) \tag{3.26}$$

$$\boldsymbol{\sigma}_{\mathbf{N}}^2(x)\triangleq\sigma^2+x_{\boldsymbol{\gamma}}^\top\mathbf{S_N}x_{\boldsymbol{\gamma}} \tag{3.27}$$

Thus, we will predict with the mode of $\mathcal{P}\left(\hat{y}|x,\boldsymbol{y},\boldsymbol{X},\boldsymbol{\gamma},\boldsymbol{\theta},\sigma^2\right)$, or the mean of the above distribution.

$$\hat{y}(x)=\boldsymbol{y}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}+c\boldsymbol{I}_p\right)^{-1}x_{\boldsymbol{\gamma}} \tag{3.28}$$

3.3 A Variational Inference Approach: Maximal Pre-Marginalization

Here, we use the results from §2.3 to further reduce the network in figure 2.1 by marginalizing out σ^2 as well as $\boldsymbol{\beta}$. At this point, the variables of interest are simply $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$. One possible path would be to simply estimate those with variational inference. However, we would also like to chose better values for the hyperparameters, selecting the MLE of a_0, b_0, and c, rather than just arbitrary values. Thus, in this setting, I place uninformative Jeffreys Priors on the hyperparameters (discussed in §2.4.3), and add them as variables of interest to the network. The result is in figure 3.2. I call this setting "Maximal Pre-Marginalization."

I use the same definition of likelihood function as described by equation 3.14. All likelihoods and gradients of variables of interest are given in §3.3.1-3.3.5, with details of the derivations provided in §B.2. As with the Partial Pre-Marginalization setting, I am interested in using my point estimates to make predictions. §3.3.7 gives the derivation of that prediction function (equation 3.52), which, you may note, is identical to the one found for Partial Pre-Marginalization (equation 3.28).

3.3.1 Likelihood and Gradient of a_0

Both are defined for $a_0\in\mathbb{R}^+$. Discussion and derivation of these equations are given in §B.2.1.

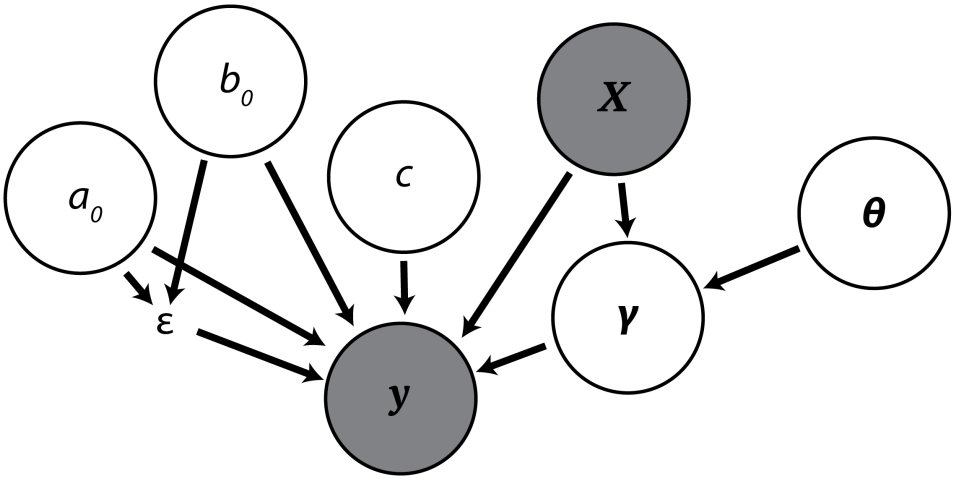

Figure 3.2: The reduced Bayesian Network for Linear Regression with DPP-sparsity with marginalized regression coefficients, $\boldsymbol{\beta}$ and variance, σ^2.

$$\mathcal{L}(a_0) = a_0 k + \log\left[\Gamma(n/2 + a_0)\right] - \log\Gamma(a_0) + \frac{1}{2}\log\left(a_0\psi^{(1)}(a_0) - 1\right) \tag{3.29}$$

$$\nabla\mathcal{L}(a_0) = k + \psi^{(0)}\left(\frac{n}{2} + a_0\right) - \psi^{(0)}(a_0) + \frac{1}{2}\left(\frac{\psi^{(1)}(a_0) + a_0\psi^{(2)}(a_0)}{a_0\psi^{(1)}(a_0) - 1}\right) \tag{3.30}$$

Where $\psi^{(i)}$ is the ith polygamma function and k is defined as follows.

$$k \triangleq \log\left(\frac{b_0}{b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}}\right) \tag{3.31}$$

3.3.2 Likelihood and Gradient of b_0

Both are defined for $b_0 \in \mathcal{B} = \{x \in \mathbb{R} | x \geq \max(0, \frac{1}{2}(\lambda_{\max} - 1)\boldsymbol{y}^\top\boldsymbol{y})\}$. Discussion and derivation of these equations are given in §B.2.2.

$$\mathcal{L}(b_0) = (a_0 + 1)\log b_0 - (a_0 + n/2)\log\left(b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}\right) \tag{3.32}$$

$$\nabla\mathcal{L}(b_0) = \frac{a_0 + 1}{b_0} - \left(\frac{a_0 + \frac{n}{2}}{b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}}\right) \tag{3.33}$$

3.3.3 Likelihood and Gradient of c

Both are defined for $c \in \mathbb{R}$, though the actual domain of c in this problem is $\mathbb{R}^+$. Discussion and derivation of these equations is given in §B.2.3.

$$\mathcal{L}(c) = \left(\frac{p^2}{2} + p\right)\log(c) - \frac{1}{2}\log\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right) - \left(a_0 + \frac{n}{2}\right)\log\left(b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}\right) \tag{3.34}$$

$$\nabla\mathcal{L}(c) = \frac{p\left(p+\frac{1}{2}\right)}{c} - \frac{1}{2}\sum_{i=1}^{p}\frac{1}{\lambda_i + c} - \left(a_0 + \frac{n}{2}\right)\frac{\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{Q}\left(\boldsymbol{\Lambda} + c\boldsymbol{I}_p\right)^{-2}\boldsymbol{Q}^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{y}}{b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}} \tag{3.35}$$

Where we have taken the eigendecomposition $\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$ and λ_i is the ith eigenvalue on the diagonal of $\boldsymbol{\Lambda}$.

3.3.4 Likelihood and Gradient of $\boldsymbol{\theta}$

Both are defined for $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$, where Θ, and any prior for $\boldsymbol{\theta}$ over Θ is left up to the user at the time of operation. Discussion and derivation of these equations are given in §B.1.3.

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i\in\boldsymbol{\gamma}}\boldsymbol{\theta}_i - \log\left(\det\left[\mathrm{Diag}\left(e^{\boldsymbol{\theta}/2}\right)\boldsymbol{X}^\top\boldsymbol{X}\,\mathrm{Diag}\left(e^{\boldsymbol{\theta}/2}\right)\right]\right) + \log p\left(\boldsymbol{\theta}|\boldsymbol{X}\right) \tag{3.36}$$

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\gamma} - \mathrm{diag}(K) + \nabla\log p(\boldsymbol{\theta}|\boldsymbol{X}) \tag{3.37}$$

3.3.5 Optimization of $\boldsymbol{\gamma}$

As noted in §B.2.4, the gradient of any function with respect to a binary vector is not well defined. Such is the case for $\mathcal{L}(\boldsymbol{\gamma})$. However, inspired by the success of Kulesza and Taskar, I will use a greedy algorithm to optimize $\boldsymbol{\gamma}$, modeled after their algorithms, reproduced below (algorithms 5 and 7) [9] [8]. In these algorithms, they essentially perform a greedy optimization of $\mathcal{P}(\boldsymbol{\gamma}|\boldsymbol{y},\boldsymbol{X})$, scaled by a constraint on each element. For my purposes, I think a greedy maximization of $\mathcal{P}(\boldsymbol{\gamma}|\,\mathrm{MB}(\boldsymbol{\gamma}))$ is more appropriate. As another adjustment, I will halt the greedy algorithm when no $i \in \mathcal{Y}(X)$ increases the likelihood. (Note, in the algorithm given in [9], there is no constraint preventing the maximal value in line 5 to be negative). To perhaps encourage earlier halting (and thus sparser $\boldsymbol{\gamma}$), I will add an l_0 regularization term. My definition of the likelihood and the modified algorithm are given below.

$$\mathcal{L}(\boldsymbol{\gamma}) = \sum_{i\in\boldsymbol{\gamma}}\boldsymbol{\theta}_i + \log\left(\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\right)\right) - \frac{1}{2}\log\left(\det\left[\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + \boldsymbol{I}\right]\right) - \left(a_0 + \frac{n}{2}\right)\log\left(b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + \boldsymbol{I}\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}\right) - \lambda_{\boldsymbol{\gamma}}\|\boldsymbol{\gamma}\|_0 \tag{3.38}$$

Algorithm 1 Greedy Estimation of $\boldsymbol{\gamma}^{\text{MAP}}$

Input: input $\boldsymbol{X}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$, a_0, b_0
Output: estimate of $\boldsymbol{\gamma}^{\text{MAP}}$
$U \leftarrow \mathcal{Y}(X)$; $\boldsymbol{\gamma} \leftarrow \emptyset$
while $U \neq \emptyset$ **do**
    $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} \cup \arg\max_{i \in U} \left(\mathcal{L}(\boldsymbol{\gamma} + \boldsymbol{e}_i) - \mathcal{L}(\boldsymbol{\gamma}) \right)$;
    $U \leftarrow U \setminus \{i\}$;
end while
return $\boldsymbol{\gamma}$;

3.3.6 Eliminating b_0

In an effort to reduce the number of hyperparameters in this problem, we can eliminate b_0 from every equation by writing it in terms of a_0 and c. Of course, in general, b_0 is a hyperparameter and is free to take on any positive value, but we can decide to always chose that value that maximizes its likelihood function, defined in equation 3.32.

To do so, note that $\nabla \mathcal{L}(b_0)$ is a linearly separable function of b_0. First, let us define the following constants.

$$x \triangleq a_0 + 1 \qquad y \triangleq a_0 + \frac{n}{2} \qquad z \triangleq \frac{1}{2} \boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c \boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y} \tag{3.39}$$

Then, we solve for optimal b_0.

$$\nabla \mathcal{L}\left(b_0\right) = \frac{x}{b_0} - \frac{y}{b_0 - z} = 0 \quad \Rightarrow \quad b_0 = \frac{xz}{y - x} \tag{3.40}$$

To verify that this is a maximum point, we show that the second derivative is negative for this choice of b_0.

$$\nabla^2 \mathcal{L}\left(b_0\right) = -\frac{x}{b_0^2} + \frac{y}{\left(b_0 - z\right)^2} \tag{3.41}$$

$$\nabla^2 \mathcal{L}\left(\frac{xz}{y-x}\right) = \frac{(y-x)^2}{z^2} \left(\frac{1}{y} - \frac{1}{x} \right) \tag{3.42}$$

Recalling our definitions of x, y, and z, equation 3.42 is clearly positive $\forall\, n > 2$, which is a very reasonable assumption. Thus, the optimal choice of b_0, once given data $\boldsymbol{X}$, labels $\boldsymbol{y}$, any feature selection, $\boldsymbol{\gamma}$, and other hyperparameters, a_0 and c, can be optimized by the following expression.

$$b_0^\star = \frac{\left(a_0 + 1\right) \boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c \boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y}}{n - 2} \tag{3.43}$$

Clearly, this is not really in the spirit of a hyperparameter as prior knowledge, since we're setting its value based on the data observed. However, this conception aligns with our conception of hyperparameters like regularization coefficients that can be tuned with validation sets.

3.3.7 Making Predictions

To make predictions, we must consider the following predictive distribution.

$$\hat{y} = \arg\max_{y\in\mathbb{R}} \mathcal{P}\left(y|x, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}, \boldsymbol{\theta}\right) = \arg\max_{y\in\mathbb{R}} \int_{\mathbb{R}^+} \mathcal{P}\left(y|x, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \sigma^2\right) \mathcal{P}\left(\sigma^2|a_0, b_0\right)\, d\sigma^2 \tag{3.44}$$

Thus, we can simply use the result from §3.2.4 and marginalize out σ^2. Recall the results:

$$\mathcal{P}\left(y|x, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \sigma^2\right) = \mathcal{N}\left(y|\mathbf{m_N}^\top x_{\boldsymbol{\gamma}}, \boldsymbol{\sigma}^2_{\mathbf{N}}(x)\right) \tag{3.45}$$

Where we use the same definitions of $\mathbf{m_N}$ and $\boldsymbol{\sigma}^2_{\mathbf{N}}(x)$. Including the inverse gamma prior on σ^2, the integral becomes

$$\int_{\mathbb{R}^+} \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2_{\mathbf{N}}(x)}} \exp\left(-\frac{\left(y - \mathbf{m_N}^\top x_{\boldsymbol{\gamma}}\right)^2}{2\boldsymbol{\sigma}^2_{\mathbf{N}}(x)}\right) \frac{b_0^{a_0}}{\Gamma\left(a_0\right)} \left(\frac{1}{\sigma^2}\right)^{a_0+1} \exp\left(-\frac{b_0}{\sigma^2}\right)\, d\sigma^2 \tag{3.46}$$

Substituting in for $\mathbf{m_N}$ and $\boldsymbol{\sigma}^2_{\mathbf{N}}(x)$, we can define the following constants, make the variable substitution $u \triangleq \sigma^2$, and rewrite our integral.

$$k \triangleq a_0 + 3/2 \qquad \alpha_1 \triangleq b_0 + \frac{1}{2}\left(y - \boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(c\boldsymbol{I}_p + \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right)^{-1} x_{\boldsymbol{\gamma}}\right)^2 \tag{3.47}$$

$$\alpha_0 \triangleq \frac{b_0^{a_0}}{\Gamma\left(a_0\right)\sqrt{2\pi} \cdot \sqrt{1 + x_{\boldsymbol{\gamma}}^\top \left(c\boldsymbol{I}_p + \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right)^{-1} x_{\boldsymbol{\gamma}}}} \tag{3.48}$$

$$\alpha_0 \int_0^\infty \frac{1}{u^k} \exp\left(-\frac{\alpha_1}{u}\right)\, du \tag{3.49}$$

Note, this is precisely the same integral as in equation A.31, which is known to converge for positive α_1. For now, assume that the constraint holds; upon finishing our solution, we will see that our choice of y justifies this assumption. The solution (equation A.34) is reproduced here.

$$\frac{\alpha_0}{\alpha_1^{k-1}} \Gamma\left(k-1\right) \tag{3.50}$$

Noting that only α_1 is dependent on y, we can rewrite the original problem as the following

$$\hat{y} = \arg\max_{y\in\mathbb{R}} \mathcal{P}\left(y|x, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}, \boldsymbol{\theta}\right) = \arg\min_{y\in\mathbb{R}} b_0 + \frac{1}{2}\left(y - \boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(c\boldsymbol{I}_p + \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right)^{-1} x_{\boldsymbol{\gamma}}\right)^2 \tag{3.51}$$

Which gives us the same prediction as that found in §3.2.4.

$$\hat{y} = \boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(c\boldsymbol{I}_p + \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right)^{-1} x_{\boldsymbol{\gamma}} \tag{3.52}$$

3.4 Optimizing Hyperparameters With Partial Marginalization of $\boldsymbol{\gamma}$

In light of the results presented in §5.1.1, I realized that it is not enough to vary $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ and hope they find a maximum, as their updates reinforce one another. However, given the results in §3.1, it is infeasible to completely marginalize $\boldsymbol{\gamma}$ once p is much more than about 10.

3.4.1 Approximation

As a potential approximation, we can pre-select the k most likely features to be selected by $\boldsymbol{\gamma}$, call them $\mathcal{S} = \{i_1, \ldots, i_k\}$. Then, we can optimize the DPP hyperparameters by marginalizing over those $\boldsymbol{\gamma}$ that are composed from elements of $\mathcal{S}$, or $\boldsymbol{\gamma} \subseteq \mathcal{S}$. This approximation is discussed in [7]. Without good reason, Kojima and Komaki chose LARS as the algorithm for selecting the most important features. Given my results in §5.1.1, this decision is better justified, as we see that without any DPP prior on $\boldsymbol{\gamma}$, the variational algorithm just returns the LARS solution. They also always set $k = 10$, which is fair, given computational constraints. I reproduce their equation below that describes how we will set hyperparameters ξ given $\mathcal{S}$.

$$\sum_{\gamma_{i1}=\{0,1\}} \sum_{\gamma_{i2}=\{0,1\}} \cdots \sum_{\gamma_{jk}=\{0,1\}} p(y_n|\gamma, \xi)p(\gamma|\xi) \tag{3.53}$$

3.4.2 Marginalizing With Bornn's Parameterization

Given the model in figure 3.2 we see that the Markov Blanket of $\boldsymbol{y}$ implies that $p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{\gamma})$. Thus, we can view equation 3.53 in this setting as a linear sum of terms $k(\boldsymbol{\gamma}) \cdot \det(L(\boldsymbol{\theta})_{\boldsymbol{\gamma}}) / \det(L(\boldsymbol{\theta}) + \boldsymbol{I}_p)$. Thus, in an effort to optimize $\boldsymbol{\theta}$, we will be most interested in the partial derivatives of $\det(L(\boldsymbol{\theta})_{\boldsymbol{\gamma}})$ and $\det(L(\boldsymbol{\theta}) + \boldsymbol{I}_p)$. To find these, first recall from [2] that the Bornn Parameterization of L is given by

$$L(\boldsymbol{\theta}) = \operatorname{Diag}\left[\exp\left(\boldsymbol{\theta}/2\right)\right] \boldsymbol{X}^\top \boldsymbol{X} \operatorname{Diag}\left[\exp\left(\boldsymbol{\theta}/2\right)\right] \tag{3.54}$$

For simplicity, we will assume that $\exp(\boldsymbol{\theta}/2)$ carries an implicit Diag operator and take $\boldsymbol{L}_{\boldsymbol{\gamma}} = L(\boldsymbol{\theta})_{\boldsymbol{\gamma}}$. We examine the first partial derivative using standard vector/matrix calculus.

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}} \det\left(\boldsymbol{L}_{\boldsymbol{\gamma}}\right) &= \frac{\partial}{\partial \boldsymbol{\theta}} \det\left(\exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right) \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right)\right) \\
&= 2 \det\left(\exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right) \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right)\right) \exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right)^{-1} \left(\frac{\exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right)}{2}\right) \operatorname{Diag}\left(\boldsymbol{\gamma}\right) \\
&= \det\left(\exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right) \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \exp\left(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2\right)\right) \operatorname{Diag}\left(\boldsymbol{\gamma}\right) \\
&= \det\left(\boldsymbol{L}_{\boldsymbol{\gamma}}\right) \operatorname{Diag}\left(\boldsymbol{\gamma}\right)
\end{aligned} \tag{3.55}$$

If instead we rewrite the expression for $\det(\boldsymbol{L}_{\boldsymbol{\gamma}})$ as a scalar expression first, we confirm this result. We know that the derivative of a scalar, ϕ, with respect to a vector, $\boldsymbol{x}$, gives a vector where the ith component gives the $\partial\phi/\partial\boldsymbol{x}_i$. We see that the above derivative with respect to the ith component is

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \det\left(\boldsymbol{L}_{\boldsymbol{\gamma}}\right) = \begin{cases} \det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right) \frac{\partial}{\partial \boldsymbol{\theta}_i} \prod_{j \in \boldsymbol{\gamma}} \exp\left(\boldsymbol{\theta}_j\right) & i \in \boldsymbol{\gamma} \\ 0 & i \notin \boldsymbol{\gamma} \end{cases} \tag{3.56}$$

Noting that the expression for the first case in equation 3.56 is simply $\det(L(\boldsymbol{\theta})_{\boldsymbol{\gamma}})$, we can write the full derivative as the following.

$$\frac{\partial}{\partial \boldsymbol{\theta}} \det\left(\boldsymbol{L}_{\boldsymbol{\gamma}}\right) = \det\left(\boldsymbol{L}_{\boldsymbol{\gamma}}\right) \boldsymbol{\gamma} \tag{3.57}$$

To find the partial derivative of $\det(L(\boldsymbol{\theta}) + \boldsymbol{I}_p)$, we cannot use the component-wise approach as easily, given that we cannot separate $\boldsymbol{\theta}$ from the sum of matrices. However, applying standard rules of vector/matrix calculus, we find the following partial. In the work below, I take $\boldsymbol{L} = L(\boldsymbol{\theta})$ for visual simplification.

$$\frac{\partial}{\partial \boldsymbol{\theta}} \det(\boldsymbol{L} + \boldsymbol{I}_p) = \det(\boldsymbol{L} + \boldsymbol{I}_p)(\boldsymbol{L} + \boldsymbol{I}_p)^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{L} \tag{3.58}$$

Noting the following,

$$\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{L} = \frac{\partial}{\partial \boldsymbol{\theta}} \exp(\boldsymbol{\theta}/2)\, \boldsymbol{X}^\top \boldsymbol{X} \exp(\boldsymbol{\theta}/2) = 2 \exp(\boldsymbol{\theta}/2)\, \boldsymbol{X}^\top \boldsymbol{X} \left(\frac{\exp(\boldsymbol{\theta}/2)}{2} \right) \mathbf{1}_p = \boldsymbol{L}\mathbf{1}_p \tag{3.59}$$

Combining this result with the fact that $L = L^\top$ and the identity from [9] that $K = L(L+I)^{-1}$, we can write the full derivative as the following.

$$\frac{\partial}{\partial \boldsymbol{\theta}} \det(\boldsymbol{L} + \boldsymbol{I}_p) = \det(\boldsymbol{L} + \boldsymbol{I}_p) K \mathbf{1}_p \tag{3.60}$$

Recall that we are trying to optimize equation 3.53 for Bornn's setting. Using the results from equations 3.57 and 3.60, we can find the gradient of equation 3.53 in this setting. After simplifying the result of the quotient rule, we find the following.

$$\frac{\partial}{\partial \boldsymbol{\theta}} \sum_{\gamma_{i1}=\{0,1\}} \cdots \sum_{\gamma_{jk}=\{0,1\}} p(y_n|\boldsymbol{\gamma}, \boldsymbol{X}) \frac{\det(\boldsymbol{L}_{\boldsymbol{\gamma}})}{\det(\boldsymbol{L} + \boldsymbol{I}_p)} = \sum_{\gamma_{i1}=\{0,1\}} \cdots \sum_{\gamma_{jk}=\{0,1\}} p(y_n|\boldsymbol{\gamma}, \boldsymbol{X}) \frac{\det(\boldsymbol{L}_{\boldsymbol{\gamma}})}{\det(\boldsymbol{L} + \boldsymbol{I}_p)} (\boldsymbol{\gamma} - K\mathbf{1}_p) \tag{3.61}$$

This result allows us to use a gradient ascent algorithm to solve for

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \sum_{\boldsymbol{\gamma}} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}) \tag{3.62}$$

3.4.3 Pre-Estimation Solution

Here I present an algorithm that provides a full solution to the sparse regression problem utilizing the partial marginalization of $\boldsymbol{\gamma}$. By the nature of this setup, we will be looking for MLEs of all parameters before selecting a value for $\boldsymbol{\gamma}^\star$. In this case, we will estimate a_0, b_0, and c before performing optimization of $\boldsymbol{\theta}$.

Estimation of c:

Recall that the parameter c is the ratio of the variance of the noise on $\boldsymbol{y}$ and the covariance of $\boldsymbol{\beta}$. This is precisely the Bayesian description of the regularization factor, λ, in Ridge Regression (CITE). Equations 3.28 and 3.52, and their similarity to the closed form solution to the Ridge Regression problem should confirm this. Therefore, we can estimate c by performing a quick optimization of the regularization coefficient for the Ridge Regression problem with $\boldsymbol{\gamma} = \mathbf{1}_p$.

Estimation of a_0 and b_0:

Recall the result from §3.3.6. We found that we could optimize $\mathcal{L}(b_0)$ by choosing

$$b_0 = \frac{(a_0+1)\,\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}}{n-2} \tag{3.63}$$

We do not have enough other constraints on a_0 and b_0 to determine them beyond this equation, but as long as they satisfy the above relation, we're all set. Thus, we will select some value for a_0 as an *actual* hyperparameter, and choose b_0 to satisfy equation 3.63.

One interesting note. Recall that a_0 and b_0 parameterize the inverse gamma prior on σ^2. The mode of $\Gamma^{-1}(a_0, b_0)$ is given by $b_0/(a_0+1)$. Thus, equation 3.63 reduces to the following.

$$\sigma^2 = \frac{\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}}{n-2} \propto \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{y}^\top \hat{\boldsymbol{y}} \tag{3.64}$$

Which, as noted, is proportional to the difference between the inner product of $\boldsymbol{y}$ and the Ridge Regression prediction, $\hat{\boldsymbol{y}}$ with regularization coefficient c.

Choosing $\boldsymbol{\gamma}$:

Once we have found $\boldsymbol{\theta}^\star$, we must select a value for $\boldsymbol{\gamma}$. We can use the greedy MAP estimate discussed above. Another option will be to randomly sample values of $\boldsymbol{\gamma}$, calculate $\mathcal{P}(\boldsymbol{\gamma}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\theta}^\star)$. We can sample $\boldsymbol{\gamma}$ from both the LARS setting (similar to the greedy MAP estimate method) as well as from $L(\boldsymbol{\theta})$.

Algorithm:

Using the concepts discussed above, the algorithm for selecting $\boldsymbol{\gamma}^\star$, and thus making predictions (using equation 3.52 again), is given below.

Algorithm 2 Selection of $\boldsymbol{\gamma}^{\text{MAP}}$ with pre-optimized hyperparameters

1: **Input:** input $\boldsymbol{X}$, $\boldsymbol{y}$, a_0, $\boldsymbol{\theta}_0$, threshold τ, functions $\mathcal{L}(\boldsymbol{\gamma})$, $\mathcal{L}(\boldsymbol{\theta}) \approx \mathcal{P}(\boldsymbol{\theta}|\boldsymbol{y})$, and $\nabla\mathcal{L}(\boldsymbol{\theta})$, learning rate α, iteration maximum T, sampling function $G(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\theta}, a_0, b_0, c)$.
2: **Output:** estimate of $\boldsymbol{\gamma}^{\text{MAP}}$
3: $c \leftarrow \arg\max_\lambda \|\boldsymbol{y} - \hat{\boldsymbol{y}}_{\text{RR}}(\lambda)\|_2^2$
4: $b_0 \leftarrow \frac{a_0+1}{n-2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}$
5: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
6: **while** $\nabla\mathcal{L}(\boldsymbol{\theta}) > \tau$ **do**
7: $\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\nabla\mathcal{L}(\boldsymbol{\theta})$
8: **end while**
9: $t \leftarrow 0$, $\boldsymbol{\gamma}^\star \leftarrow \mathbf{0}_p$
10: **while** $t < T$ **do**
11: $\quad\boldsymbol{\gamma} \leftarrow G(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\theta}, a_0, b_0, c)$
12: $\quad$**if** $\mathcal{L}(\boldsymbol{\gamma}) < \mathcal{L}(\boldsymbol{\gamma}^\star)$ **then**
13: $\quad\quad\boldsymbol{\gamma}^\star \leftarrow \boldsymbol{\gamma}$
14: $\quad$**end if**
15: **end while**
16: **return** $\boldsymbol{\gamma}^\star$

3.5 Computational Adjustments for the Large-p Small-n Setting

The computations discussed in these notes primarily rely on inner products of $\boldsymbol{X}$ with itself, thus requiring computations on $\boldsymbol{X}^\top\boldsymbol{X} \in \mathbb{R}^{p\times p}$. This is clearly preferred over computations in $\mathbb{R}^{n\times n}$ when $p < n$. However, in the case of large-p small-n datasets (biological/genetic datasets), computations in $\mathbb{R}^{p\times p}$ are to be favored. Here, I examine the primary equations used in §3.4 and offer alterations that allow us to perform computations in $\mathbb{R}^{n\times n}$.

3.5.1 Reduction of $\hat{\boldsymbol{\beta}}$

We make predictions $f(x) \mapsto \hat{y}$ both in the estimation of c (step 3 of algorithm 3) and in the computation of $\hat{\boldsymbol{\beta}}$ after the training of our regression. In both cases, we can rewrite the equation, which, as given computes matrices in $\mathbb{R}^{p\times p}$, to use matrices in $\mathbb{R}^{n\times n}$ instead.

The given equation is

$$\hat{y} = \boldsymbol{y}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\left(c\boldsymbol{I}_p + \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\right)^{-1}x_{\boldsymbol{\gamma}} \tag{3.65}$$

Recall the matrix identity

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1}DA^{-1} \tag{3.66}$$

Setting $A \triangleq c\boldsymbol{I}_p$, $B \triangleq \boldsymbol{X}_{\boldsymbol{\gamma}}^\top$, $C \triangleq \boldsymbol{I}_n$, and $D \triangleq \boldsymbol{X}_{\boldsymbol{\gamma}}$, we can rewrite the inverse function as follows.

$$\left(c\boldsymbol{I}_p + \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\right)^{-1} = \frac{1}{c}\left(\boldsymbol{I}_p - \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\left(c\boldsymbol{I}_n + \boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}\right) \tag{3.67}$$

Thus, we can reduce the full expression, using only inner products in the feature space between observations in $\boldsymbol{X}$ and/or test data points x.

$$\hat{y} = \frac{1}{c}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \left(c\boldsymbol{I}_n + \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right)^{-1} \right) \boldsymbol{X}_{\boldsymbol{\gamma}} x_{\boldsymbol{\gamma}} \tag{3.68}$$

3.5.2 Reduction of the "Difference Projection:" $b_N - b_0$

In several computations, including $p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2)$ and $p(\boldsymbol{y}|\boldsymbol{\gamma})$, we compute a quantity I have sometimes been referring to as the difference projection, or $\frac{1}{2}\|\boldsymbol{y}\|^2_{\boldsymbol{P}}$, where the projection matrix is given by the following difference.

$$\boldsymbol{P} = \boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \tag{3.69}$$

Comparing my results to the literature, it is clear that this value is the difference between b_0, the prior estimation of the scale parameter to the variance inverse Gamma prior, and b_N, the posterior value of the same parameter. We can rewrite this value to include only calculations in $\mathbb{R}^{n\times n}$ by using the result in equation 3.67.

$$\begin{aligned} &\frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y} \\ &\quad = \frac{1}{2c}\boldsymbol{y}^\top \left(c\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top + \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \left(c\boldsymbol{I}_n + \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y} \end{aligned} \tag{3.70}$$

3.5.3 Computing DPP Likelihoods in $\mathbb{R}^{n\times n}$

In the bulk of the work discussed in these notes, I use the following notation for an L-ensemble used in Kulesza and Taskar [9].

$$L = B^\top B = \left(\boldsymbol{X} \exp\left[\boldsymbol{\theta}/2\right] \right)^\top \left(\boldsymbol{X} \exp\left[\boldsymbol{\theta}/2\right] \right) \tag{3.71}$$

In this setting, we know that the likelihood of drawing a particular item from a DPP is given by

$$\mathcal{P}\left(\boldsymbol{\gamma}\right) = \frac{\det\left(L_{\boldsymbol{\gamma}}\right)}{\det\left(L + I\right)} \tag{3.72}$$

And that a random draw from L can be found by using algorithm 6. Clearly, these computations rely on calculations in $\mathbb{R}^{p\times p}$. However, if we represent our DPP in the dual form presented in [9], $C = BB^\top$, then we can restrict our computations of $\mathbb{R}^{n\times n}$.

$$C = BB^\top = \boldsymbol{X} \exp\left[\boldsymbol{\theta}\right] \boldsymbol{X}^\top \tag{3.73}$$

As shown in Kulesza and Taskar, [9], proposition 3.1, the nonzero eigenvalues of C and L are identical, and their eigenvectors are related by the following.

$$C = \sum_{i=1}^{n} \lambda_i \hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^\top \tag{3.74}$$

is an eigendecomposition of C iff the following is an eigendecomposition of L.

$$L = \sum_{i=1}^{n} \lambda_i \left(\frac{1}{\sqrt{\lambda_i}} B^\top \hat{\boldsymbol{v}}_i \right) \left(\frac{1}{\sqrt{\lambda_i}} B^\top \hat{\boldsymbol{v}}_i \right)^\top \tag{3.75}$$

Given an eigendecomposition $L = (Q^L)\Lambda(Q^L)^\top$, we see that the following holds.

$$\det(L_{\boldsymbol{\gamma}}) = \left(\det(Q^L_{\boldsymbol{\gamma}})\right)^2 \prod_{i \in \boldsymbol{\gamma}} \lambda_i \tag{3.76}$$

We know from the above result from [9] that the ith column of Q^L is given by $B^\top \hat{\boldsymbol{v}}_i / \sqrt{\lambda_i}$, and that the nonzero eigenvalues of C are the same as those of L. We need not worry about the case of eigenvalues equal to zero, since these features are sampled with probability 0. Thus, $L_{\boldsymbol{\gamma}}$ will never be singular.

Again, since C and L share the same nonzero eigenvalues, we know that $\det(L + I) = \det(C + I)$; a fact that is confirmed in [9].

Thus, we can compute any probability that $\boldsymbol{\gamma}$ is drawn from the DPP using only the eigenvectors of the dual representation, $C = BB^\top$.

3.5.4 Sampling from the DPP Dual

In addition to calculating probabilities, we will need to sample from our DPP. Fortunately, Kulesza and Taskar offer an algorithm for doing so with the dual representation, $C = BB^\top$, [9].

Algorithm 3 Sampling from the DPP Dual, $C = BB^\top$

Input: $\{(\hat{\boldsymbol{v}}_i, \lambda_i)\}_{i=1}^n$ of $C = \boldsymbol{X} \exp[\boldsymbol{\theta}] \boldsymbol{X}^\top$
$J \leftarrow \emptyset$
for $i = 1, 2, \ldots, n$ **do**
    $J \leftarrow J \cup \{i\}$ with prob. $\frac{\lambda_i}{\lambda_i + 1}$
end for
$\hat{V} \leftarrow \left\{ \frac{\hat{\boldsymbol{v}}_i}{\sqrt{\hat{\boldsymbol{v}}_i^\top C \hat{\boldsymbol{v}}_i}} \right\}_{i \in J}$
$\boldsymbol{\gamma} \leftarrow \mathbf{0}$
while $|\hat{V}| > 0$ **do**
    Select i from $\mathcal{Y}$ with $\mathcal{P}(i) = \frac{1}{|\hat{V}|} \sum_{\hat{\boldsymbol{v}} \in \hat{V}} \left(\hat{\boldsymbol{V}}^\top B_i \right)^2$
    $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} + \boldsymbol{e}_i$
    Let $\hat{\boldsymbol{v}}_0$ be a vector in $\hat{V}$ with $B_i^\top \hat{\boldsymbol{v}}_0 \neq 0$
    Update $\hat{V} \leftarrow \left\{ \hat{\boldsymbol{v}} - \frac{\hat{\boldsymbol{v}}^\top B_i}{\hat{\boldsymbol{v}}_0^\top B_i} \hat{\boldsymbol{v}}_0 | \hat{\boldsymbol{v}} \in \hat{V} - \{\hat{\boldsymbol{v}}_0\} \right\}$
    Orthonormalize $\hat{V}$ with respect to the dot product $\langle \hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2 \rangle = \hat{\boldsymbol{v}}_1^\top C \hat{\boldsymbol{v}}_2$
end while
return $\boldsymbol{\gamma}$

Chapter 4

Code

4.1 Synthetic Data Generation

4.1.1 `LinearDataGenerator.py`

Used to create synthetic data sets based on the linear model described in §2.2.

Functionality: Primary functionality is in creating the `LinearDataGenerator` object. The constructor produces the training data, `ldg.X` with labels `ldg.y`. Hyperparameters can be accessed in a similar way for comparison after MLE estimations. Test data sets of size n can be generated by calling `Xtest,test = ldg.getUniformTestData(n)`.

Optional Arguments and Default Values:

Parameter	Value	Comment
`p`	`6;`	Number of features in the dataset. Keeping this small allows for absolute solutions.
`n`	`1000;`	Number of data points in the training set.
`gam_p`	`2;`	Desired cardinality of $\boldsymbol{\gamma}$.
`a0`	`1.0;`	"Actual" value of hyperparameter a_0.
`b0`	`1.0;`	"Actual" value of hyperparameter b_0.
`width`	`1.0;`	Variance for datapoint generation.

Notes:

- The data points in $\boldsymbol{X}$ are drawn from a zero-mean, `p`-dimensional gaussian with `width`$\cdot I_p$ variance.
- The hyperparameter $\boldsymbol{\theta}$ is never computed, making verification a little tricky. However, if I could easily take the mode of a DPP, I could define a "true" $\boldsymbol{\theta}$ and then produce a $\boldsymbol{\gamma}$ from that.

Further Work:

- Type checks on constructor.
- Work out expression for MLE of DPP and define $\boldsymbol{\gamma}$ from a $\boldsymbol{\theta}$-parameterized L-ensemble instead of picking a random $\boldsymbol{\gamma}$ with specified cardinality.

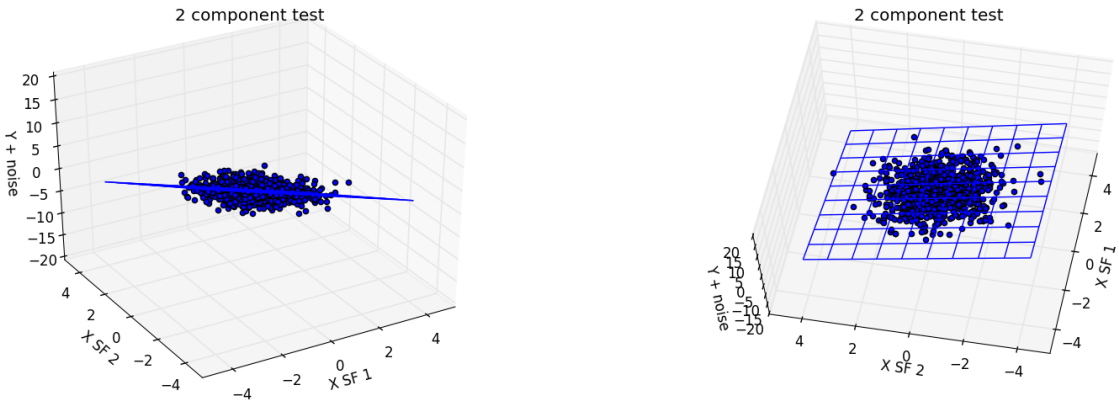

Figure 4.1: **Sample Results:** With all the default values, the data points plotted against the perfect predictor is plotted in three dimension.

4.1.2 `KojimaKomakiDataGen.py`

This class exactly reproduces the numerical dataset discussed in the 2014 paper by Kojima and Komaki [7]. For full details, see their paper, section 4. Below is a figure as evidence of my results.

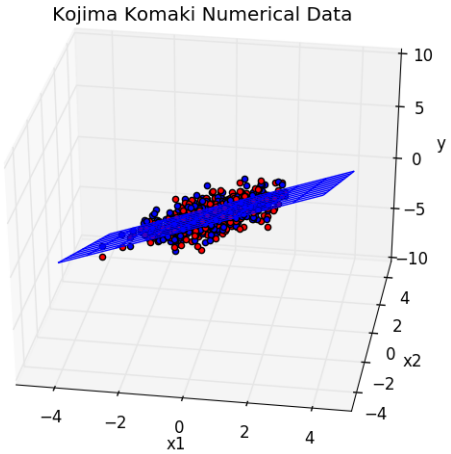

Figure 4.2: **Sample Results:** Using a single data set of 400 data points and the same optimal $\beta^\star$, I generated two labelings, differing only in their random noise. One labeling is plotted in blue, the other in red.

4.2 Utility Functions: DPPutils.py

This file does not define a class or object, but rather is simply a list of functions that perform common operations that occur when working with DPPs. Note, these are all *functions* that accept numpy.ndarray arguments, not any generalized experiment or variable objects.

columnGammaZero(X,gamma):

X can be a non-square matrix (numpy.ndarray). gamma indexes which columns remain untouched. The rest go to zero.

gammaZero2D(X,gamma):

X must be a square matrix (numpy.ndarray). Zero entries of gamma index the rows and columns of X that will become zero. We do not remove these rows and columns in the result.

gammaRM2D(X,gamma):

X must be a square matrix (numpy.ndarray). Zero entries of gamma index the rows and columns of X that will be zeroed and removed.

getKDiag(L):

L must be a square matrix (numpy.ndarray). It should be symmetric, but the method does not check for that. This function returns the diagonal of the kernel DPP K corresponding to the L-ensemble L. No memoization is used here, since L is thought to be a function of $\boldsymbol{\theta}$, which is changing with the same regularity that the diagonal of K must be calculated. Should this change in the future, move this function to the state-tracking object's class (VI?).

getK(L):

L must be a square matrix (numpy.ndarray). It should be symmetric, but the method does not check for that. This function returns the kernel DPP K corresponding to the L-ensemble L. No memoization is used here, since L is thought to be a function of $\boldsymbol{\theta}$, which is changing with the same regularity that the diagonal of K must be calculated. Should this change in the future, move this function to the state-tracking object's class (VI?).

greedyMapEstimate(p,L):

p gives the dimension of the inclusion vector, and L gives whatever likelihood function the greedy estimate is trying to maximize. This function returns the greedy MAP estimate of $\boldsymbol{\gamma}$, as given by algorithm 1.

4.3 Bayesian Network Variable: BNV.py

Bayesian Nnetwork Varable is an abstract data type representing a variable in a Bayesian Network. The primary methods are `likelihood(self, state)` and `update(self, state)`, which take in some experimental state object and compute the likelihood of or update the current value of `this BNV`, respectively. The `state` was inspired by the `VI` object, but in reality needs only be an experiment object that contains a dictionary of its network variables `state.bnv`, and a dictionary of any hyperparameters their calculations require, `state.hp`.

Based on the property `isiterative`, the `BNV`'s update function will either use the learning rate `defaultAlpha` and likelihood gradient `gradLikelihood(self, state)` or some user-specified method for updating the `BNV`'s current value. For instance, a continuous variable like $\boldsymbol{\theta}$ can be updated iteratively with a gradient, but a discrete binary vector like $\boldsymbol{\gamma}$ cannot, and must be updated with a uniquely defined method.

Other methods include getters, setters, and checkers to retrieve values, set values, and verify values.

NOTE: All notions of network dependence and connectivity are left to the user to bake into the likelihood and update calculations. This allows the user to include any relevant shortcuts due to particular variable construction, constraints, etc.

4.3.1 BNVs for Maximal Pre-Marginalization Variational Inference Experiment

Below are notes on the `BNV` objects made for the bayesian network in §3.3.

BNV for a_0: `BN_a0.py`

This class models a_0, the shape parameter for the Inverse Gamma prior on the variance. It is iterative, and has a default value of 1.0, and a default learning rate of 1.0e−2. The likelihood and its gradient are computed using the functions found in §3.3. Currently, checks on the value of a_0 simply verify that it's positive.

BNV for b_0: `BN_b0.py`

This class models b_0, the scale parameter for the Inverse Gamma prior on the variance. It is iterative, and has a default value of 5.0 and a default learning rate of 1.0e−2. The likelihood and its gradient are computed using the functions found in §3.3. Currently, checks on the value of b_0 simply verify that it satisfies equation A.43.

BNV for $\boldsymbol{\theta}$ `BN_theta.py`

This class models $\boldsymbol{\theta}$, the parameter for the L-ensemble, governing the DPP prior on $\boldsymbol{\gamma}$. It is iterative and a default learning rate of 1.0e−2. Its default value is the p-dimensional zero vector, which corresponds to setting $L = \boldsymbol{X}_{\boldsymbol{\gamma}}^{\top}\boldsymbol{X}_{\boldsymbol{\gamma}}$. The likelihood and its gradient are

computed using the functions found in §3.3. The likelihood and gradient include a term for a prior distribution on $\boldsymbol{\theta}$, dependent only on $\boldsymbol{X}$. This prior is not a specified part of our model, so I have left this as a general term. The constructor for the `BN_theta` object takes a parameter specifying which prior to use. Priors are defined in the separate file `theta_priors.py`, discussed in §4.4.

BNV for $\boldsymbol{\gamma}$ `BN_gamma.py`

This class models $\boldsymbol{\gamma}$, the inclusion vector on the regression coefficients. Due to the fact that it is a binary vector, its gradient is not well-defined. Thus, it is non iterative. Its update function instead implements the greedy MAP estimate algorithm defined in algorithm 1. Its initial value is a random binary vector with at least one non-zero entry. Checks on `BN_gamma` verify that it is of type `numpy.ndarray`, shape `(p,1)`, and its elements are all either `0.0` or `1.0`. CHANGE THIS TO BE A MORE EVEN DISTRIBUTION.

4.4 Prior Distributions, $\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{X})$, `theta_priors.py`

4.5 Variational Inference Control Object: `VI.py`

4.5.1 Structure and Instance Variables

The role of the `VI` object is to maintain and log the "current" state of a variational inference experiment. This includes keeping all hyperparameters, all training data, and all current point-estimates of variables in the bayesian network. However, it is *not* the responsibility of the `VI` object to maintain the actual structure of the bayesian network. This structure is baked into the `BNV` object definitions.

4.5.2 Utilities

Despite having a separate file for my list of utility functions §4.2, some repetitive calculations are best handled by the experiment object (here, the `VI` object), in order to handle experiment logging and verification settings, as well as fully utilize memoization in iterative experiments. Each of the computationally heavy functions (determinants and inverses) uses a dictionary to memoize past computations, indexed on values of $\boldsymbol{\gamma}$. While I will be using this code for iterative solutions where $\boldsymbol{\gamma}$ is not necessarily likely to remain constant or return to some already visited value, I may well be computing these values in optimizations of multiple other variables, i.e., that of a_0, b_0, and $\boldsymbol{\theta}$.

`getL(self):`

Not using memoization, this method returns the full (unindexed) L-ensemble corresponding to the current state's value of $\boldsymbol{\theta}$. Note, this implicitly requires that `theta` be a valid key in `self.bnv`.

`FdetSI(self,gamma):`

`gamma` is a binary inclusion vector, expected to be of length `self.p`. Using memoization to log the full result, it returns $\det(S_{\boldsymbol{\gamma}} + I)$ where $S = \boldsymbol{X}^\top \boldsymbol{X}$.

`FdetL(self,gamma,theta):`

`gamma` is a binary inclusion vector, expected to be of length `self.p`, and `theta` is the parameterization of the L-ensemble, also of length `self.p`. Using memoization to log $|S_{\boldsymbol{\gamma}}|$, this computes the numerator of L-ensemble probability expression, or $|\exp(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2) S_{\boldsymbol{\gamma}} \exp(\boldsymbol{\theta}_{\boldsymbol{\gamma}}/2)|$. This method utilizes the trick noted in §A.3.3.

`FDifferenceProjection(self,gamma):`

`gamma` is a binary inclusion vector, expected to be of length `self.p`. This method utilizes the `X` and `y` variables in the `VI` instance. Using memoization to log the final result, it returns $\frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + I_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y}$. I call it the difference projection, since it is the projective norm of $\boldsymbol{y}$ onto the difference of two spaces in $\mathbb{R}^{n \times n}$.

Chapter 5

Experiments

5.1 Regression Comparisons with Fake Data

5.1.1 Small p with Bornn's Parameterization

Dataset: I used the exact fake dataset used in [7]. Six features, all normally distributed. The last three are reduced by a factor of 0.1, and have a combination of two of the untouched columns added to them. I used $\boldsymbol{\beta} = \boldsymbol{e}_1 - \boldsymbol{e}_2$, and added noise to $\boldsymbol{y}$.

Models: I use DPP1 to refer to the DPP solution in §3.3, or the model that we are completely optimizing, and DPP2 to refer to the solution in §3.2, where hyperparameters are given. I perform a grid search for $\lambda_{\boldsymbol{\gamma}}$ with only 500 iterations at each step to choose the best regularization parameter (for the l_0 norm in $\mathcal{L}(\boldsymbol{\gamma})$), and then optimize with that choice of $\lambda_{\boldsymbol{\gamma}}$ for 2,500 iterations. I compare this with Ordinary Least Squares Regression, Ridge (also with a quick grid search), LASSO (also with a quick grid search), and the Oracle, which has knowledge of $\boldsymbol{\gamma}^\star$, but uses the OLSR $\boldsymbol{\beta}$.

Findings:

- Hyperparameters for DPP1 converged to be very close to the MLEs for OLSR. In fact, they converged exactly to the MLEs for the oracle.
- Fixing the hyperparameters to incorrect values (DPP2) did not have much affect on the prediction accuracy.
- All values in DPP1 converged, but $\boldsymbol{\gamma}$ never changed. DPP1 and DPP2 performed well, because with the correct choice of $\lambda_{\boldsymbol{\gamma}}$, the optimization chose the correct value the first time.
- LASSO often did not converge with the given number of iterations, while DPP1 (and 2) did.

Hypotheses, Quick Checks, and Future Experiments:

- I believe that $\boldsymbol{\gamma}$ does not ever change due to its relations with $\boldsymbol{\theta}$. I start $\boldsymbol{\theta}$ at a neutral value of $\mathbf{0}$. Once a $\boldsymbol{\gamma}$ is selected, the update on $\boldsymbol{\theta}$ is $\boldsymbol{\gamma} - \text{diag}(K)$, which leads to a $\boldsymbol{\theta}$ that reinforces that choice of $\boldsymbol{\gamma}$. Right now, $\boldsymbol{\theta}$ only serves to reinforce

a given choice of $\boldsymbol{\gamma}$. I will need to have its setting be determined by other values in the network, i.e., $p(\boldsymbol{y}|\boldsymbol{\theta})$.

- Given that $\boldsymbol{\theta}$ is not playing a substantive role in the selection of $\boldsymbol{\gamma}$ at the beginning, and $\boldsymbol{\gamma}$ doesn't change, $\boldsymbol{\theta}$ is irrelevant. Check: Removing $\boldsymbol{\theta}$ from all other calculations leaves us again choosing the same $\boldsymbol{\gamma}$. $\boldsymbol{\theta}$ is not involved with any other optimizations, so this gives the same results.

- The only remaining DPP influence in this problem is the $\log\det(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\top}\boldsymbol{X}_{\boldsymbol{\gamma}})$ term in $\mathcal{L}(\boldsymbol{\gamma})$. Is it relevant at all? Check: No, it is not. After I remove that term from the likelihood, I see no change. Since I am optimizing the hyperparameters to MLEs of a known regression problem, my setting is actually identical to LARS with an l_0 penalty on the number of variables added.

Figures:

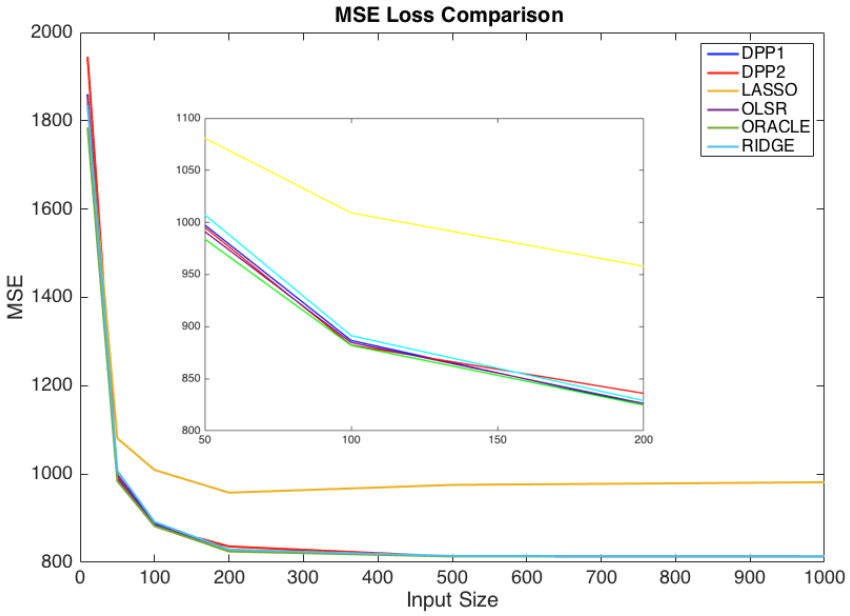

Figure 5.1: Total MSE loss over 1000 test points for each model.

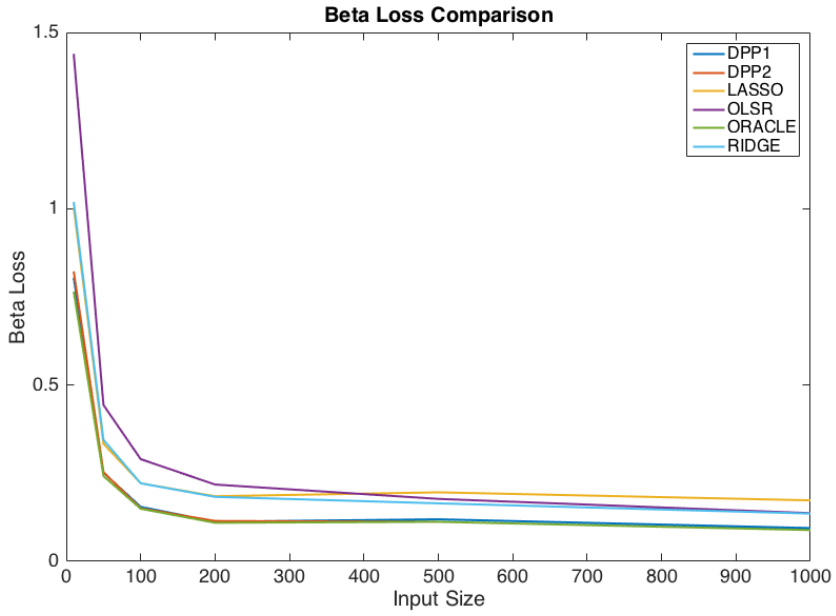

Figure 5.2: Beta loss is the given as $\|\boldsymbol{\beta}^\star - \hat{\boldsymbol{\beta}}\|_\infty$

5.1.2 Pre-Optimization of Hyperparameters with Partial Marginalization of γ with Bornn's Parameterization

Dataset: I randomly generated 35 datasets inspired by the fake dataset used in [7]. They were generated with about half of the columns drawn from a normal distribution. The other half were normal noise plus a unique combination of two of the original columns. Features were then shuffled randomly. I chose $\boldsymbol{\gamma}^\star$ to have sparsity ~ 0.90. (Closer to ~ 0.50 for $p < 10$). I generated these datasets for all $6 < p < 40$.

Models: I used the model described in §3.4. I ran four different instances of this model over the datasets above, testing all combinations of $\{500, 1000\}$ iterations to optimize $\boldsymbol{\theta}$ and $\{500, 1000\}$ draws from our DPP.

Findings: From the results, it appears that optimizing $\boldsymbol{\theta}$ is most influential in correctly estimating $\boldsymbol{\gamma}^\star$. Further, it appears that over-sampling from the DPP can actually decrease the performance of the algorithm, perhaps allowing us to find a local, overfit maximum.

Figures: Below are histograms of the results. The first set show us how the predicted $\hat{\boldsymbol{\gamma}}$ compare in magnitude to the actual $\boldsymbol{\gamma}^\star$. Overall, we see that my technique tends to predict more non-zero entries than actually exist. A further test to examine how this is affected when $\boldsymbol{\gamma}^\star$ is less sparse ($\|\boldsymbol{\gamma}^\star\|_0 > 10$) would be interesting here. The second set shows the number of errors made (sum of excess features included and necessary features left out, $\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^\star\|_0$ as a percentage of the total number of features. The third set shows

the error as a percentage of the number of nonzero features, or $\|\boldsymbol{\gamma}^\star\|_0$. In all of these experiments, $(N_{\boldsymbol{\theta}}, N_{\boldsymbol{\gamma}}) = (1000, 500)$ performed best.

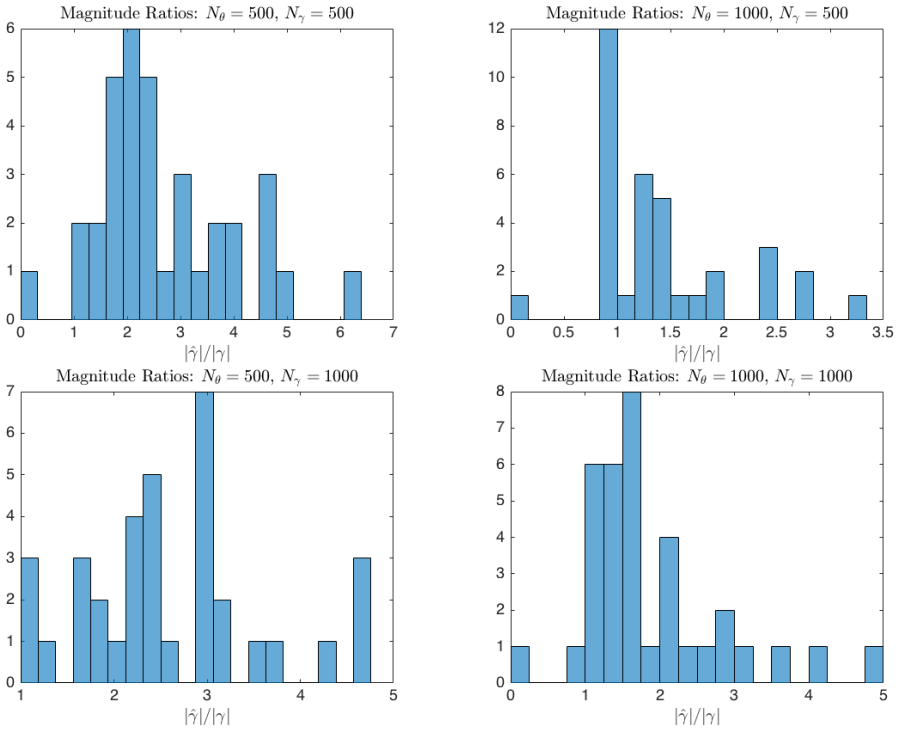

Figure 5.3: Histograms of the relative magnitudes of predicted and actual $\boldsymbol{\gamma}$.

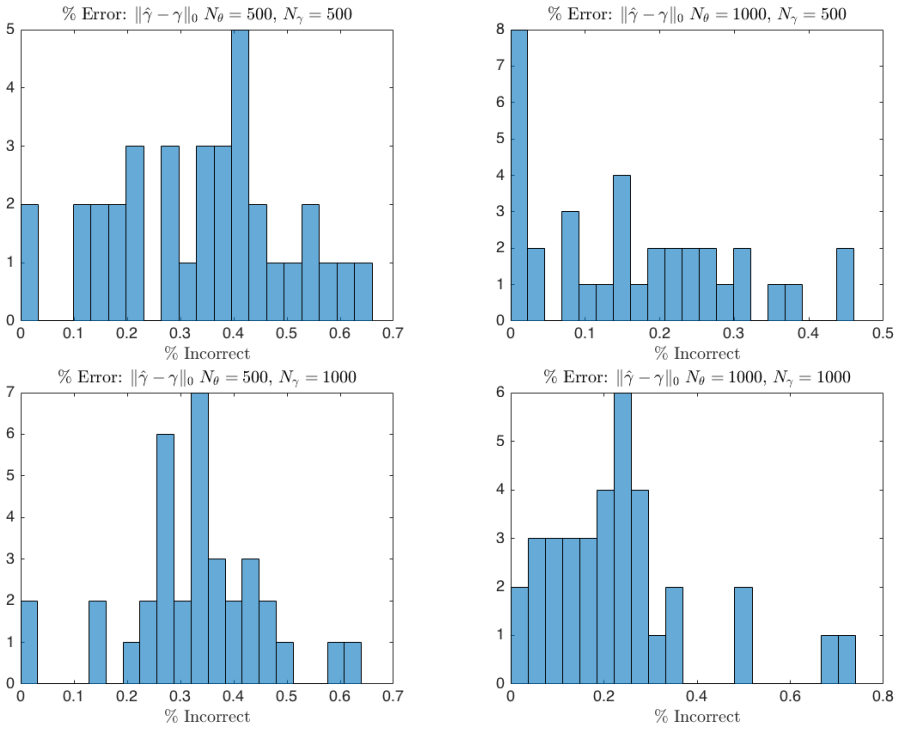

Figure 5.4: Histograms of the percent of mistakes in $\hat{\boldsymbol{\gamma}}$ relative to the total number of features.

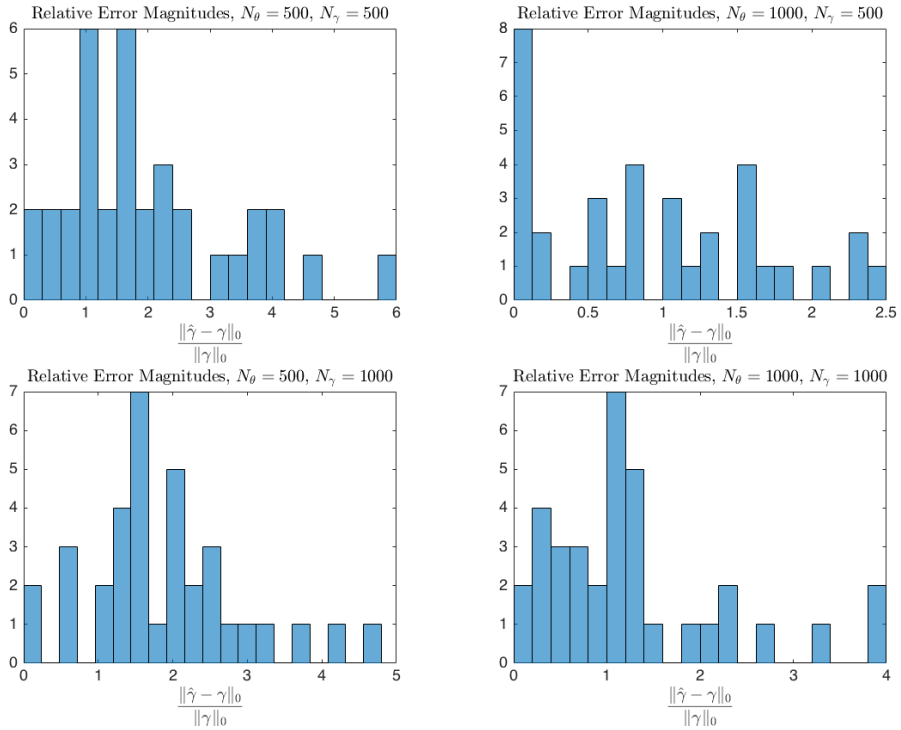

Figure 5.5: Histograms of the percent of mistakes in $\hat{\boldsymbol{\gamma}}$ relative to the total number of features included in $\boldsymbol{\gamma}^\star$.

Chapter 6

Paper Reviews

6.1 Bornn et al., 2014 - Diversifying Sparsity Using Variational Determinantal Point Processes

Paper: L. Bornn, *et al.*, *Diversifying Sparsity Using Variational Determinantal Point Processes*. arXiv:1411.6307v1, Nov. 23, 2014. [2]

6.1.1 Overview

Performing Bayesian Variable Selection using DPPs to introduce sparsity.

6.1.2 Bayesian Variable Selection

Section 3 of the paper
Standard regression model with regressors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ combined into a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M] \in \mathbb{R}^{N \times M}$, learned coefficients $\boldsymbol{\beta} \in \mathbb{R}^M$ and residual noise $\varepsilon \sim \mathcal{N}(\cdot; 0, \sigma)$.

$$\boldsymbol{y} = \sum_{m=1}^{M} \boldsymbol{x}_m \beta_m + \varepsilon \tag{6.1}$$

If we note those features included in the regression with $\boldsymbol{\gamma} \in \{0,1\}^M$, we rewrite equation 6.1 as

$$\boldsymbol{y} = \boldsymbol{X}(\boldsymbol{\gamma} \odot \boldsymbol{\beta}) + \varepsilon \tag{6.2}$$

6.1.3 Priors:

- We assign an exchangeable Bernoulli prior to $\boldsymbol{\gamma}$, and a conjugate prior for the regression coefficients:
$$\boldsymbol{\beta} \sim \mathcal{N}(\cdot; 0, \sigma^2 \Lambda_0^{-1}) \tag{6.3}$$
- The random variable $\gamma_m \beta_m$ defines the "spike-and-slab" prior, being drawn from the spike w.p. α and from the slab w.p $1 - \alpha$.
- Assign an inverse Gamma prior for the variance: $\sigma^2 \sim \Gamma^{-1}(a_0, b_0)$.

6.1.4 Joint Likelihood

Define $\pi = \{\boldsymbol{\beta}, \sigma, \boldsymbol{\gamma}\}$ as the set of latent random variables and $\rho = \{\Lambda_0, a_0, b_0, \alpha\}$ is the set of fixed parameters, with $\Lambda_0 = c\boldsymbol{I}$. We have the joint likelihood:

$$p(\boldsymbol{y}, \pi; \boldsymbol{X}, \rho) = p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma; \boldsymbol{X}) p(\boldsymbol{\beta}|\sigma; \gamma\Lambda_0) p(\sigma; a_0, b_0) p(\boldsymbol{\gamma}; \alpha) \tag{6.4}$$

6.1.5 Marginal Likelihood

Let us define

$$\Lambda_N = \boldsymbol{X}^\top \boldsymbol{X} + \Lambda_0, \in \mathbb{R}^{M \times M} \qquad \boldsymbol{\mu}_N = \Lambda_N^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \in \mathbb{R}^M$$

$$a_N = a_0 + N/2 \qquad b_N = b_0 + \frac{1}{2}\left(\boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{\mu}_N^\top \Lambda_N\right)$$

Born gives the closed-form marginal likelihood is given by

$$\begin{aligned} \log p(\boldsymbol{Y}|\boldsymbol{\gamma}; \boldsymbol{X}, \rho) \neq -\frac{N}{2}\log 2\pi + \frac{1}{2}\left(\log\det(\Lambda_0) - \log\det(\Lambda_N)\right) \\ +(a_0 - a_N)(\log b_0 - \log b_N) + \log\Gamma(a_N) - \log\Gamma(a_0) \end{aligned} \tag{6.5}$$

Bornn has a typo here n his definition of b_N. Note that $\boldsymbol{\mu}_N^\top \Lambda_N \notin \mathbb{R}$. Actually, the marginal is given by

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \frac{1}{(2\pi)^{N/2}} \frac{|\boldsymbol{\Lambda}_N^{\boldsymbol{\gamma}}|^{1/2}}{|\boldsymbol{\Lambda}_0^{\boldsymbol{\gamma}}|^{1/2}} \frac{\Gamma(a_N) b_0^{a_0}}{\Gamma(a_0)(b_N^{\boldsymbol{\gamma}})^{a_N}} \tag{6.6}$$

Where the constants are given by

6.1.6 Posterior Probability

Marginalizing $\boldsymbol{\beta}$ out, we get the posterior probability

$$q(\boldsymbol{\gamma}; \boldsymbol{\theta}) = \prod_{m=1}^{M} q(\gamma_m; \theta_m) = \prod_{m=1}^{M} \theta_m^{\gamma_m}(1 - \theta_m)^{1-\gamma_m} \tag{6.7}$$

Though this does not account for any interaction between regressors, namely, sparsity. Instead, consider the posterior

$$q(\boldsymbol{\gamma}; \boldsymbol{\theta}) = \frac{1}{Z_\theta} \det[\boldsymbol{L}]_{\boldsymbol{\gamma}} = \frac{1}{Z_\theta} e^{\boldsymbol{\theta}^\top \boldsymbol{\gamma}} \det\left[\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\right]_{\boldsymbol{\gamma}} \tag{6.8}$$

Or, we can sample from the DPP defined by the L-ensemble

$$\boldsymbol{L} = e^{\mathrm{Diag}\,\boldsymbol{\theta}/2} \boldsymbol{\Phi}\boldsymbol{\Phi}^\top e^{\mathrm{Diag}\,\boldsymbol{\theta}/2} \tag{6.9}$$

With parameters $\boldsymbol{\theta}$, latent random variables $\boldsymbol{\gamma}$ and $Z_{\boldsymbol{\theta}} = \det(\boldsymbol{I} + \boldsymbol{L})$ is the normalization constant.

Their given algorithm is derived from Salimans and Knowles [12] work on variational learning. The algorithm is reproduced here in algorithm 4. First, some notation:

- We take $\boldsymbol{\theta}$ as the parameters of our posterior DPP, where the defining L-ensemble is given by

$$\boldsymbol{L} = \text{Diag}\left(e^{\frac{\boldsymbol{\theta}}{2}}\right)\boldsymbol{\Phi}\boldsymbol{\Phi}^\top \text{Diag}\left(e^{\frac{\boldsymbol{\theta}}{2}}\right) \tag{6.10}$$

- During the iterative solution for Variational Learning, we use the following augmented vectors to account for normalization and base measures:

$$\tilde{\boldsymbol{\theta}}^\top = \left[\boldsymbol{\theta}^\top, \theta_0\right] \qquad \tilde{\boldsymbol{\gamma}}^\top = \left[\boldsymbol{\gamma}^\top, 1\right] \tag{6.11}$$

- ~~How do we define the posteriors? Bornn 5) is not equivalent to the exponential form described before 7). (Exponentiated normalizer???)~~ Questions resolved in meeting (§1.1). Barbara suggests we ignore the augmented vectors (no need to worry about intercepts) and just deal with $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$.

6.1.7 Bornn's Algorithm

Algorithm 4 Variational Learning for Diverse VS, reproduced from [2].

Input: Similarity features $\boldsymbol{\Phi}$, a function to compute/approximate the restricted marginal likelihood $p(y|\gamma)$, $p(\gamma)$, initial cardinality of DPP κ, number of iterations, N.
Output: Parameters of the posterior, (q): $\boldsymbol{\theta}$.
Adjust expected cardinality of the DPP by solving for θ_0 in $\sum_i \frac{e^{\theta_0}\lambda_i}{1+\lambda_i e^{\theta_0}} = \kappa$;
Initialize $\boldsymbol{\theta} = \theta_0 \mathbb{1}$, $\boldsymbol{L} = e^{\theta_0/2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top e^{\theta_0/2}$, and $\boldsymbol{K} = \boldsymbol{L}(\boldsymbol{L}+\boldsymbol{I})^{-1}$;
Set $(\boldsymbol{C}_1)_{ii} = K_{ii}$, $\boldsymbol{g}_1 = \boldsymbol{C}_1\tilde{\boldsymbol{\theta}}$, and $\bar{\boldsymbol{C}}, \bar{\boldsymbol{g}} = 0$;
for $t \leftarrow 1 \dots N$ **do**
    Draw a set from the current posterior approximation DPP: $\boldsymbol{\gamma}_t^* \sim q_{\tilde{\boldsymbol{\theta}}_t}$;
    Set $\hat{\boldsymbol{g}}_t = \tilde{\boldsymbol{\gamma}}_t^* \log p(y|\boldsymbol{\gamma}_t^*)p(\boldsymbol{\gamma}_t^*)$;
    Set $\hat{\boldsymbol{C}}_t = \tilde{\boldsymbol{\gamma}}_t^*\tilde{\boldsymbol{\gamma}}_t^{*\top}$ or current estimate of $\boldsymbol{K}_{\boldsymbol{\theta}_t}$;
    Set $\boldsymbol{g}_{t+1} = (1-w)\boldsymbol{g}_t + w\hat{\boldsymbol{g}}_t$;
    Set $\boldsymbol{C}_{t+1} = (1-w)\boldsymbol{C}_t + w\hat{\boldsymbol{C}}_t$;
    Solve $\tilde{\boldsymbol{\theta}}_{t+1} = \boldsymbol{C}_{t+1}^{-1}\boldsymbol{g}_{t+1}$;
    if $t > N/2$ **then**
        Set $\bar{\boldsymbol{g}} = \bar{\boldsymbol{g}} + \hat{\boldsymbol{g}}_t$;
        Set $\bar{\boldsymbol{C}} = \bar{\boldsymbol{C}} + \hat{\boldsymbol{C}}_t$;
    end if
end for
return $\boldsymbol{\theta} = \bar{\boldsymbol{C}}^{-1}\bar{\boldsymbol{g}}$;

6.1.8 A word on Φ

We take $\boldsymbol{\Phi} \in \mathbb{R}^{M\times d}$ as the matrix of similarity features, where row m, $\phi(m)$ is the similarity feature vector for item m. The expression for the L-ensemble in 6.9 is a little different from the canonical L-ensemble definition in [9], $L = B^\top B$, and instead looks like their dual representation, $C = BB^\top$. However, they define columns of B as $B_i = q_i\phi_i$ where q_i is the ith quality scalar and ϕ_i is the ith normalized diversity feature vector,

$\phi_i \in \mathbb{R}^D$, $\|\phi_i\| = 1$. However, in this case, Bornn [2] treats $\mathbf{\Phi}$ differently, saying that row m of $\mathbf{\Phi}$ corresponding to similarity feature vectors for item m. Thus, $\mathbf{\Phi} \approx B^\top$. Address distinction between *similarity* and *diversity* feature vectors with Barbara (see §1.1).

This paper seems to have a number of inconsistencies in it, w.r.t. notation as well as some seemingly arbitrary decisions in methodology. I.e., why is there necessarily a Bernoulli prior (posterior) and a DPP posterior (prior)? If we think that $\boldsymbol{\gamma}$ is inherently sparse, then why don't we make its prior *and* posterior DPPs? Simple answer is that the math is annoying, but using some methods from [1] and [5], we should be able to come up with something.

6.2 Affandi, Fox, Adams, Taskar, 2014 - Learning the Parameters of DPP Kernels

Paper: Raja Hafiz Affandi, Emily B. Fox, Ryan P. Adams, and Ben Taskar. *Learning the Parameters of Determinantal Point Process Kernels.* In **ICML**, 2014. [1]

6.2.1 Overview

They present Bayesian methods for learning DPP kernel parameters even for large scale DPPs where ordinary calculations are intractable.

6.2.2 Notation

They define the discrete base set $\Omega = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the PSD kernel matrix L, and subset A, giving the canonical

$$\mathcal{P}_L(A) = \frac{\det(L_A)}{\det(L+I)}$$

Given some finite value $0 \leq k \leq N$, we define the k-DPP, which gives positive mass only to subsets of size k as

$$\mathcal{P}_L^k(A) = \frac{\det(L_A)}{e_k(\lambda_1, \ldots, \lambda_N)}$$

Where $e_k(\cdot)$ is the kth elementary symmetric polynomial. They construct the DPP with parameters Θ as the following

$$L(\boldsymbol{x}, \boldsymbol{y}; \Theta) = q(\boldsymbol{x}; \Theta) k(\boldsymbol{x}, \boldsymbol{y}; \Theta) q(\boldsymbol{y}; \Theta) \tag{6.12}$$

Where $q(\cdot)$ is a scalar quality measure for the given feature, and $k(\cdot, \cdot)$ is a kernel giving the "similarity" between its two arguments.

6.2.3 Optimization Methods

We wish to maximize the log-likelihood:

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log\det(L_{A^t}(\Theta)) - T\log\det(L(\Theta)+I) \tag{6.13}$$

For discrete DPPs, we can compute the gradient for use with optimization techniques. However, it is a non-convex function, and thus may converge to a local optimum. Even so, computing the derivative of equation 6.13 is very inefficient when N is large, or if sets A^t are large.
Consider the log-likelihood for the k-DPP kernel parameter

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log\det(L_{A^t}(\Theta)) - T\log\sum_{|B|=k}\det(L_B(\Theta)) \tag{6.14}$$

6.2.4 A Bayesian Approach

Instead of optimizing the likelihood to get an MLE, they estimate the posterior distribution over kernel parameters:

$$\mathcal{P}(\Theta|A^1,\ldots,A^T) \propto \mathcal{P}(\Theta)\prod_{t=1}^{T}\frac{\det(L_{A^t}(\Theta))}{\det(L(\Theta)+I)} \quad \text{or for } k\text{-DPP:} \quad \propto \mathcal{P}(\Theta)\prod_{t=1}^{T}\frac{\det(L_{A^t}(\Theta))}{e_k(\lambda_1(\Theta),\ldots,\lambda_N(\Theta))} \tag{6.15}$$

Where $\mathcal{P}(\Theta)$ is the prior on Θ. Neither the posterior in equation 6.15 for DPPs or k-DPPs has a closed form solution. They approximate with MCMC, using random-walk Metropolis-Hastings and slice sampling.

Random-walk Metropolis-Hastings:

They use some proposal distribution $f(\hat{\Theta}|\Theta_i)$ to help find value of Θ on the subsequent iteration, given the value of Θ at the current iteration. The candidate, $\hat{\Theta}$ is accepted or rejected with probability $\min\{r,1\}$ (equation 6.16). They chose the proposal distribution to have mean Θ_i and hyper parameters to tune the width. The choice of the proposal is important. An aggressive one, with high rejection rates can cause a drastic increase in computation time. However, a very conservative proposal distribution will not adequately explore the parameter space.

$$r = \left(\frac{\mathcal{P}(\hat{\Theta}|A^1,\ldots,A^T)}{\mathcal{P}({}_i\Theta|A^1,\ldots,A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right) \tag{6.16}$$

Slice Sampling:

Instead of tuning the proposal distribution, they implement slice sampling [11]. The process in the single-dimensional case is given in algorithm 2 of the paper [1]. They say that they extend it to the multidimensional case by using hyperrectangles, expanding and shrinking each dimension independently, as proposed by [11]. [1] proposes extensions with coordinate-wise or random-direction approaches.

Extension for large N:

When N is large, either MH or slice sampling can take a long time; computing the normalizing determinant is very inefficient. Thus, we utilize an efficient bound-tightening procedure for $\mathcal{P}(\Theta_i|A^1, \ldots, A^T)$. Algorithm 3 and Figure 1 of the supplement of [1] describe how to apply this idea to MH and slice sampling techniques for MCMC.
They offer that the upper and lower bounds depend on the truncation of kernel eigenvalues and can be tightened by including more terms. Note, this will require calculating only a few eigenvalues at a time. Use Python `scipy.sparse.linalg.eigs` based on the Fortran77 package ARPACK, optimal for finding largest eigenvalues.

The paper finishes with extensions to the continuous DPP realm and a few experiments. It is worth noting that instead of simply using a covariance kernel (as in Bornn [2]) they use a Gaussian similarity kernel. This seems potentially more interesting. Also, they've already done out the math.

6.3 Gillenwater, Kulesza, Fox, Taskar, 2014 - Expectation-Maximization for Learning DPPs

Paper: Jennifer Gillenwater, Alex Kulesza, Emily B. Fox, and Ben Taskar. NIPs, 2014. [5]

6.3.1 Overview

They offer a method of overcoming the NP-hard issue of solving the non-convex optimization of learning L given data. Instead of assuming a parameterization of L (like [1]), they allow for an arbitrary construction, exploiting the eigendecomposition of L. They propose one advantage of this over naive gradient ascent for expectation-maximization is that the projection step tends toward learning nearly diagonal L, which under-represents the negative correlations of set elements. They also claim that it's faster than gradient ascent. They do not offer any specific comparisons to the parametric solutions offered by [1] except that it does not rely on some parameterization. This is a lot less standard for Bayesian Inference type learning.

6.3.2 Notation

Using the notation from [9], they denote the ground set of items as $\mathcal{Y} = \{1, \ldots, N\}$, noting that $\mathcal{P}(Y \subseteq \mathcal{Y}) \propto \det(L_Y)$, where L_Y is the matrix formed by intersections of rows and columns indexed by elements of Y. Recall the relationships between the L-ensemble and the marginal kernel, K, which shares eigenvectors $\boldsymbol{v}$ with L, with K's eigenvalues are

related to those of L by $\lambda_i/(1-\lambda_i)$. Definitions and probabilities are:

$$\begin{aligned} K &= L(L+I)^{-1} \\ K &= \sum_{j=1}^{N} \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^\top \\ L &= \sum_{j=1}^{N} \frac{\lambda_j}{1-\lambda_j} \boldsymbol{v}_j \boldsymbol{v}_j^\top \\ \mathcal{P}(A \subseteq Y) &= \det(K_A) \\ \mathcal{P}(Y) &= \frac{\det(L_Y)}{\det(L+I)} = |\det(K - I_{\overline{Y}})| \end{aligned}$$

In their learning algorithms, they have an interesting approach. Instead of learning an optimal sparsity set from some generic, potentially dense data, they assume that the input is in the form of n example subsets $\{Y_1, \ldots, Y_n\}$, where $Y_i \in \mathcal{Y}$.

6.3.3 Projected Gradient Ascent

As a point of comparison, they describe the gradient ascent EM method in their framework. The log-likelihood problem is

$$\max_K \sum_{i=1}^{n} \log\left(|\det(K - I_{\overline{Y}_i})|\right) \text{ s.t. } K \succeq 0,\ I - K \succeq 0 \tag{6.17}$$

Where the constraints ensure that K has eigenvalues $\lambda_i \in [0, 1]$. The gradient used for the ascent algorithm (detailed in Algorithm 1 in their paper [5]) is given by

$$\frac{\partial \mathcal{L}(K)}{\partial K} = \sum_{i=1}^{n} (K - I_{\overline{Y}_i})^{-1} \tag{6.18}$$

Problems with this method include:

- Mixed concavity/convexity can lead to local maxima solutions.
- Computation time is limited by finding matrix inverses, $O(N^3)$, and finding eigendecompositions for the projections, also $O(N^3)$. Given T_1 iterations until convergence and an average of T_2 iterations to find step sizes, the total running time is $O(T_1 n N^3 + T_1 T_2 N^3)$.

6.3.4 Eigendecomposition

Recall the generation of the hidden variable $J \subseteq \{1, \ldots, N\}$ coding eigenvectors of K during the sampling procedure as given by Kulesza and Taskar [9]. They take an eigendecomposition of $K = V \Lambda V^\top$, noting V^J as the submatrix of V indexed on elements of J.

UNFINISHED – not sure this is something I want to pursue yet. I'll focus on parameterized learning. Come back to this later perhaps?

6.4 Kojima, Komaki, 2014 - DPP Priors for Bayesian Variable Selection in Linear Regression

Paper: Mutsuki Kojima and Fumiyasu Komaki. arXiv:1406.2100, 2014. [7]

6.4.1 Overview

Very straightforward implementation of linear regression with DPP priors. They go through the results with four different kinds of DPPs and the experimental/numerical results of their implementations. Great related works list and some interesting justification for the composition of the L-ensemble. The four kinds of DPPs are the diagonal DPP or simply, Bernoulli distribution, a standard DPP, a linear mixture DPP, and a geometric mixture DPP.

6.4.2 Model

They use n observations on a dependent variable $\boldsymbol{y} \in \mathbb{R}^n$ with p predictor variables, concatenating them in $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. Their linear model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \varepsilon_n \tag{6.19}$$

Where $\varepsilon_n \in \mathbb{R}^n \sim \mathcal{N}(0, \sigma^2 I_n)$ and $\boldsymbol{\beta} \in \mathbb{R}^p$. They define $\gamma \in \{0,1\}^p$ as a feature selection parameter. They define parameterized models as

$$M_\gamma : \boldsymbol{y} = \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma + \varepsilon_n \tag{6.20}$$

which is basically identical to my model in 2.5. Thus, their problem is to select MAP estimation, maximizing $p(\gamma|\boldsymbol{y})$. They justify the need for sparsity as eliminating problems of collinearity (and thus instability of $(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$) in standard of linear regression solutions. They follow up with a great related works list. Since we want to avoid collinearity, building the L-ensemble out of the feature covariance matrix makes a lot of sense.

6.4.3 Review of Bayesian Variable Selection Methods

This paper utilizes comparisons to (George and Foster 2000) to variable selection methods. [LOOK INTO THIS PAPER AND ADD CITATION].

For a given sub model M_γ, the coefficients are β_γ, and are given the prior

$$p(\beta_\gamma|g) \sim \mathcal{N}(0, g\sigma^2 \left(X_\gamma^\top X_\gamma\right)^{-1}),\ g > 0 \tag{6.21}$$

George and Foster use a Bernoulli prior on γ:

$$p(\gamma|w) = w^{|\gamma|}(1-w)^{p-|\gamma|} \tag{6.22}$$

Which gives the MAP solution

$$\hat{\gamma} = \arg\max_\gamma \exp\left(\frac{g}{2(1+g)}(\mathrm{ss}_\gamma/\sigma^2 - F(g,w)|\gamma|)\right) = \arg\max_\gamma(\mathrm{ss}_\gamma/\sigma^2 - F(g,w)|\gamma|) \tag{6.23}$$

Where they defined

$$\text{ss}_\gamma = y_n^\top X_\gamma X_\gamma^\top y_n, \qquad F(g,w) = \frac{1+g}{g}\left(2\log\frac{1-w}{w} + \log(1+g)\right) \tag{6.24}$$

I'm a little skeptical of this setup, but I believe the expression for ss comes from the following. We take ss_γ as the sum of squares of the γth model, or $ss_\gamma = \beta_\gamma^\top X_\gamma^\top X_\gamma \beta_\gamma$. Further, in a typical regression problem, we have $\hat{\beta} = \left(X_\gamma^\top X_\gamma\right)^{-1} X_\gamma^\top Y$. Plugging this in for β_γ in the expression for ss_γ, we find that $ss_\gamma = y^\top X_\gamma \left(X_\gamma^\top X_\gamma\right)^{-1} X_\gamma^\top y$. Here, in 6.4, they list $ss = y^\top X_\gamma X_\gamma^\top y$. I'm still not sure where this came from, and it may be a typo.

6.4.4 DPP Models

A few notes:

- They use a standardized matrix, subtracting the mean and deciding by the standard deviation of each feature, pushing each feature toward $\mathcal{N}(0,1)$.
- They (really George and Foster) find MLEs of their hyperparameters. Do I need to do this? I feel like I probably should, yes?

They use four different distributions:

- Bernoulli prior as discussed above.
- DPP prior with $L = w\tilde{X}^\top \tilde{X}$. Where they are artificially weighting the likelihood of different magnitude γ. They are not strictly using k-DPPs, but are encouraging lower dimensionality.
- Linear DPP with $L = w\left(\theta\tilde{X}^\top\tilde{X} + (1-\theta)I_\gamma\right)$, where w is defined as before, and $\theta \in [0,1]$. We see that $\theta = 1$ returns the Bernoulli distribution.
- Geometric DPP with $L = w\left(\tilde{X}^\top\tilde{X}\right)^\alpha$ with $\alpha \geq 0$, where $\alpha = 1$ gives the DPP prior, and $\alpha = 1$ gives the Bernoulli distribution.

In all cases, γ is the only non-hyperparameter, and we are most interested in maximizing the type II likelihood, or the posterior probability $p(\gamma|y_n)$.

6.4.5 Numerical Methods

- They artificially generate their data. Look into this.
- They use the loss function:

$$\|\beta^\star - \hat{\beta}\|_\infty = \max_i |\beta_i^\star - \hat{\beta}_i| \tag{6.25}$$

  Justifying this use by saying that maximum loss function is more appropriate than the standard quadratic loss when investigating collinearity.

- Estimate hyperparameters by finding the MLE of the marginal $p(y_n|g, w, \theta)$. See paper for actual equation.
- They compare also with ridge regression, ordinary least squares, and the oracle (same as ordinary least squares, but with known optimal $\gamma^\star$ parametrization). They also estimate the optimal λ for ridge regression by finding the MLE of $p(y_n|\lambda)$. Again, see paper for details.

6.4.6 Applying models to real data

- They apply their DPPs to real data, a design matrix X, both relevant to Air Pollution Data and Body Fat Data. For both of these datasets they assume some non-zeroed model with

$$y_n = \mu 1_n + X\beta + \varepsilon_n \qquad \text{submodels: } M_\gamma : y_n = \mu 1_n + X_\gamma \beta_\gamma + \varepsilon_n \tag{6.26}$$

- They assume that columns of X are standardized. I.e., each feature has been standardized.
- Apply standard methods, Ridge, and OLS.
- For Bayesian variable selection, set hyperparameters, μ, σ^2, g, w, and θ by maximizing the type II likelihood. In the particular case of GDPP, they maximize $p(y_n|\alpha)$ for α over $[0, 3]$ (as opposed to $\mathbb{R}^+$) because $X^\top X$ is ill-conditioned. I.e., making this sort of concession for a suboptimal setting is accepted practice.
- To reduce computation in estimation of hyperparameters, they reduce X by selecting only the 10 most relevant features using least angle regression. More precisely, "for $l = 1, 2, \ldots, 60$, we select 10 predictors $x_{j1}, \ldots, x_{j10}$ by LARS and hyperparameters are estimated by maximizing the sum

$$\sum_{\gamma_{j1}=\{0,1\}} \sum_{\gamma_{j2}=\{0,1\}} \cdots \sum_{\gamma_{j10}=\{0,1\}} p(y_n|\gamma, \xi)p(\gamma|\xi) \tag{6.27}$$

  where ξ denotes all hyperparameters." They point out that each evaluation sums the probability product 2^p times, hence justifying their partial sum.

6.5 Kulesza, Taskar, 2011 - Learning Determinantal Point Processes

Paper: Alex Kulesza and Ben Taskar, *Learning Determinantal Point Processes*, Conference on Uncertainty in Intelligence (UAI), Barcelona Spain, July 2011. [8]
Note, this is the shorter precursor paper to [9].

6.5.1 Overview

Using a motivating example of extractive multi-document summarization, Kulesza and Taskar demonstrate how to learn the MLE of DPP parameters and then find the MAP estimate. Note, this is the exact setting of my problem in §2.2. They compare the DPP results to those of Markov Random Fields, a popular choice, though it is known to be intractable for exact solutions.

6.5.2 Notation

Using their standard notation, the full set that we draw from is $\mathcal{Y}$, and we have the symmetric kernel matrix and L-ensembel, $K, L \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ related by $K = (L+I)^{-1}L$, as well as all of the other relations presented in [9].

Here they use the Gram matrix

$$L_{ij} = q_i \phi_i^\top \phi_j q_j \tag{6.28}$$

Where $q_i \in \mathbb{R}^+$ is a quality measurement, and $\phi_i \in \mathbb{R}^n$, $\|\phi_i\|_2 = 1$ is a normalized feature vector, such that $\phi_i^\top \phi_j \in [-1, 1] \, \forall \, i, j \in \{1, \ldots, |\mathcal{Y}|\}$ measures the similarity between items i and j.

$$S_{ij} \triangleq \phi_i^\top \phi_j = \frac{L_{ij}}{\sqrt{L_{ii} L_{jj}}} \tag{6.29}$$

Note, as of 160228, in my proposed setting (§2.1), I leave ϕ_i unnormalized, but constrain $q_i \in [0, 1]$. Perhaps I will revisit this setting.

Regardless, in this setting, the probability of a draw, Y from a DPP is given by

$$\mathcal{P}_L(\mathbf{Y} = Y) = \frac{\left(\prod_{i \in Y} q_i^2\right) \det\left(S_Y\right)}{\det(L+I)} \tag{6.30}$$

Kulesza and Taskar also highlight the conditional probability

$$\mathcal{P}(\mathbf{Y} = A \cup B | A \subseteq \mathbf{Y}) = \frac{\det\left(L_{A \cup B}\right)}{\det\left(L + I_{\mathcal{Y} \setminus A}\right)} \tag{6.31}$$

For their experiments, they place the parameterization of the DPP in the calculation of q_i with

$$q_i(X) = \exp\left(\frac{1}{2} \theta^\top \boldsymbol{f}_i(X)\right) \tag{6.32}$$

where $\theta \in \mathbb{R}^p$ is the parameter vector and $\boldsymbol{f}_i(X) \in \mathbb{R}^p$ is a feature vector, used for modeling quality, which may in general be distinct from $\phi_i(X)$, the feature vector used for measuring similarity.

6.5.3 Comparison of DPPs and MRFs

As a point of comparison, Kulesza and Taskar a Markov Random Field with repulsive potentials. They set $y_i = \mathbb{I}(i \in Y)$ to represent the binary inclusion vector sampled from

the DPP. Here, the sample distribution is given by the following:

$$P(\mathbf{Y} = Y) \propto \exp\left(\sum_i w_i y_i + \sum_{i<j} w_{ij} y_i y_j\right) \tag{6.33}$$

Where they enforce the constraint that $w_{ij} < 0$. Note, this has the same number of parameters as the proposed DPP. However, as they point out, there is an important distinction here where for the MRF, $w_{ij} < 0$ is an individual constraint on the parameters, while the positive semi-definite constraint on the DPP kernel is global *and* transitive, per the triangle inequality:

$$\sqrt{1 - S_{ij}^2} + \sqrt{1 - S_{jk}^2} \geq \sqrt{1 - S_{ik}^2} \tag{6.34}$$

6.5.4 MLE Learning

The problem statement is as follows. Given T pairs, $(X^1, Y^1), \ldots, (X^T, Y^T)$, as inputs, we are interested in the corresponding output set of $Y^t \in \mathcal{Y}(X^t)$. Note, this implies that the learning set labels are themselves inclusion vectors. This is *not* a good parallel to the linear regression setting in §2.2. Learning the parameters, θ amounts to maximizing the log-likelihood of the training set:

$$\mathcal{L}(\theta) = \log \prod_t \mathcal{P}_\theta(Y^t|X^t) = \sum_t \log \mathcal{P}_\theta(Y^t|X^t) \tag{6.35}$$

The full log-likelihood is shown to be concave:

$$\log \mathcal{P}_\theta(Y|X) = \log\left(\frac{\prod_{i\in Y}\left[\exp\left(\theta^\top \boldsymbol{f}_i(X)\right)\right] \det\left(S_Y(X)\right)}{\sum_{Y'\subseteq\mathcal{Y}(X)} \prod_{i\in Y'}\left[\exp\left(\theta^\top \boldsymbol{f}_i(X)\right)\right] \det\left(S_{Y'}(X)\right)}\right) \tag{6.36}$$

$$\log \mathcal{P}_\theta(Y|X) = \theta^\top \sum_{i\in Y} \boldsymbol{f}_i(X) + \log\det(S_Y(X)) - \log \sum_{Y'} \exp\left(\theta^\top \sum_{i\in Y'} \boldsymbol{f}_i(X)\right) \det\left(S_{Y'}(X)\right) \tag{6.37}$$

With respect to θ, the first term is linear, the second is constant, and the third is the composition of a concave function (negative log-sum-exp) and a non-negative linear function, so it is also concave. Thus, with an efficient computation of $\nabla\mathcal{L}(\theta)$, standard convex optimization is feasible. The gradient is given by

$$\nabla\mathcal{L}(\theta) = \sum_{i\in Y} \boldsymbol{f}_i(X) - \sum_{Y'} \mathcal{P}_\theta(Y'|X) \sum_{i\in Y'} \boldsymbol{f}_i(X) \tag{6.38}$$

With the full derivation presented in [9], page 48. They note that the sum over Y' is exponential in $|\mathcal{Y}(X)|$, but that they can rewrite

$$\sum_{Y'} \mathcal{P}_\theta(Y'|X) \sum_{i\in Y'} \boldsymbol{f}_i(X) = \sum_i \boldsymbol{f}_i(X) \sum_{Y'\supseteq\{i\}} \mathcal{P}_\theta(Y'|X) \tag{6.39}$$

Where we see that $\sum_{Y'\supseteq\{i\}} \mathcal{P}_\theta(Y'|X)$ is the marginal probability of item i appearing in a set sampled from the θ-conditioned DPP, which is given by the diagonal of $K(X;\theta)$, recalling from [9]

$$\mathcal{P}(A \subseteq \boldsymbol{Y}) = \det\left(K_A\right) \tag{6.40}$$

If we take the eigendecomposition of the L-ensemble

$$L_\theta(X) = \sum_k \lambda_k v_k v_k^\top \tag{6.41}$$

We can find the marginals:

$$K_{ii} = \sum_k \frac{\lambda_k}{1+\lambda_k} v_{ki}^2 \tag{6.42}$$

And use them to find the gradient:

$$\nabla \mathcal{L}(\theta) = \sum_{i \in Y} \boldsymbol{f}_i(X) - \sum_i K_{ii} \boldsymbol{f}_i(X) \tag{6.43}$$

From there, any sort of gradient descent algorithm can be use to infer θ.

6.5.5 MAP Inference

For prediction, they want to pick a Y for some new unseen X. They mention sampling from the conditional distribution, but claimed better performance with a constrained MAP estimate:

$$Y^{\text{MAP}} = \arg\max_Y \mathcal{P}_\theta(Y|X) \quad \text{s.t.} \quad \sum_{i \in T} = \text{cost}(i) \leq B \tag{6.44}$$

They cite that an exact solution is NP-hard, and instead propose two approximations. First, to brute-force estimate by sampling sets Y, and selecting the one that maximizes the above expression. The other option is to apply a greedy algorithm.

Algorithm 5 Greedy Estimation of Y^{MAP}, reproduced from [8].

1: **Input:** input X, parameters θ, budget B
2: **Output:** set Y
3: $U \leftarrow \mathcal{Y}(X)$; $Y \leftarrow \emptyset$
4: **while** $U \neq \emptyset$ **do**
5: $\quad Y \leftarrow Y \cup \arg\max_{i \in U} \left(\frac{\mathcal{P}_\theta(Y \cup \{i\}|X) - \mathcal{P}_\theta(Y|X)}{\text{cost}(i)} \right)$;
6: $\quad U \leftarrow U \setminus \{i | \text{cost}(Y) + \text{cost}(i) > B\}$;
7: **end while**
8: **return** Y;

6.6 Kulesza, Taskar, 2013 - Determinantal Point Processes for Machine Learning

Paper: Alex Kulesza and Ben Taskar, *Determinantal Point Processes for Machine Learning. arXiv*, 2013. [8]
Note, this is the extended version of to [8].

6.6.1 Overview

This paper offers a collection of most of Kulesza and Taskar's work on DPPs from the last few years. I am not yet tracking notes on the full thing, but as I investigate sections in detail, I will add them here.

6.6.2 Sampling from a DPP

Algorithm 6 `sample(L, lam)`: Sampling from a DPP, Algorithm 1 from [9].

Input: Eigendecomposition $\{(\boldsymbol{v}_n, \lambda_n)\}_{n=1}^N$ of L
$J \leftarrow \emptyset$
for $n = 1, 2, \ldots, N$ **do**
    $J \leftarrow J \cup \{n\}$ with prob. $\frac{\lambda_n}{\lambda_n + 1}$
end for
$V \leftarrow \{\boldsymbol{v}_n\}_{n \in J}$
$Y \leftarrow \emptyset$
while $|V| > 0$ **do**
    Select i from $\mathcal{Y}$ with $\Pr(i) = \frac{1}{|V|} \sum_{\boldsymbol{v} \in V} (\boldsymbol{v}^\top \boldsymbol{e}_i)^2$
    $Y \leftarrow Y \cup i$
    $V \leftarrow V_\perp$, an orthonormal basis for the subspace of V orthogonal to $\boldsymbol{e}_i$.
end while
Output: Y

6.6.3 Learning Quality Parameters

Kulesza and Taskar do not directly address parameterized DPPs (at least not in the sense that [1] does). But they do offer some very useful derivations. Using the notation of $L = B^\top B$ where columns of B are given by the product of a scalar quality term, q_i and a similarity feature vector ϕ_i, they utilize the fact that this results in an L-ensemble of the following form.

$$L = \begin{bmatrix} q_1^2\langle\phi_1, \phi_1\rangle & q_1 q_2\langle\phi_1, \phi_2\rangle & \cdots & q_1 q_p\langle\phi_1, \phi_p\rangle \\ q_2 q_1\langle\phi_2, \phi_1\rangle & q_2^2\langle\phi_2, \phi_2\rangle & \cdots & q_2 q_p\langle\phi_2, \phi_p\rangle \\ \vdots & \vdots & \ddots & \vdots \\ q_p q_1\langle\phi_p, \phi_1\rangle & q_p q_2\langle\phi_p, \phi_2\rangle & \cdots & q_p^2\langle\phi_p, \phi_p\rangle \end{bmatrix} \tag{6.45}$$

Recall that the determinant of a matrix A' created by multiplying a row or column of matrix A by a constant k is given by the following [10].

$$\det(A') = k \det(A) \tag{6.46}$$

Thus, we can reduce the expression for the determinant of L.

$$\det L = \left(\prod_{i=1}^{p} q_i^2 \right) \det(S) \tag{6.47}$$

where S is the Gram matrix of all ϕ_i.

Kulesza and Taskar keep their formulation general by defining the quality terms as

$$q_i(X;\theta) = \exp\left(\frac{1}{2}\theta^\top \boldsymbol{f}_i(X)\right) \tag{6.48}$$

They assert that the generalized $\boldsymbol{f}_i(X)$ terms are distinct from $\phi_i(X)$ used to generate S, since $\boldsymbol{f}_i(X)$ are used for modeling quality, while $\phi_i(X)$ are used to model diversity. Given these definitions, Kulesza and Taskar give a very clever derivation for the gradient of the log likelihood.

We start by rewriting the probability in the form of equation 2.4

$$\mathcal{P}_\theta(Y|X) = \frac{\prod_{i\in Y}\left[\exp\left(\theta^\top \boldsymbol{f}_i(X)\right)\right]\det\left(S_Y(X)\right)}{\sum_{Y'\subseteq\mathcal{Y}(X)}\prod_{i\in Y'}\left[\exp(\theta^\top \boldsymbol{f}_i(X))\right]\det\left(S_{Y'}(X)\right)} \tag{6.49}$$

Then, the log likelihood is given by the following.

$$\mathcal{L}(\theta) = \log\mathcal{P}_\theta(Y|X) = \theta^\top\sum_{i\in Y}\boldsymbol{f}_i(X) + \log\det(S_Y(X)) - \log\sum_{Y'\subseteq\mathcal{Y}(X)}\exp\left(\theta^\top\sum_{i\in Y'}\right)\det(S_{Y'}(X)) \tag{6.50}$$

The gradient is then

$$\nabla\mathcal{L}(\theta) = \sum_{i\in Y}\boldsymbol{f}_i(X) - \nabla\left[\log\sum_{Y'\subseteq\mathcal{Y}(X)}\exp\left(\theta^\top\sum_{i\in Y'}\boldsymbol{f}_i(X)\right)\det(S_{Y'}(X))\right] \tag{6.51}$$

$$= \sum_{i\in Y}\boldsymbol{f}_i(X) - \sum_{Y'\subseteq\mathcal{Y}(X)}\frac{\exp\left(\theta^\top\sum_{i\in Y'}\boldsymbol{f}_i(X)\right)\det\left(S_{Y'}(X)\right)\sum_{i\in Y'}\boldsymbol{f}_i(X)}{\sum_{Y'}\exp\left(\theta^\top\sum_{i\in Y'}\boldsymbol{f}_i(X)\right)\det\left(S_{Y'}(X)\right)} \tag{6.52}$$

$$= \sum_{i\in Y}\boldsymbol{f}_i(X) - \sum_{Y'\subseteq\mathcal{Y}(X)}\mathcal{P}_\theta(Y'|X)\sum_{i\in Y'}\boldsymbol{f}_i(X) \tag{6.53}$$

The very clever trick that Kulesza and Taskar apply now is to change the order of summation in 6.53, to avoid the exponential sum over $i \in Y'$, and then notice that this is exactly the marginal probability of a single element appearing in a sample, which is given exactly by the basic DPP formula,

$$\mathcal{P}\left(Y \subseteq \boldsymbol{Y}\right) = \det\left(K_Y\right) \tag{6.54}$$

First, with the swapped sums:

$$\nabla\mathcal{L}(\theta) = \sum_{i\in Y}\boldsymbol{f}_i(X) - \sum_i \boldsymbol{f}_i(X)\sum_{Y'\supseteq\{i\}}\mathcal{P}_\theta\left(Y'|X\right) \tag{6.55}$$

Leaving us with the result:

$$\nabla\mathcal{L}(\theta) = \sum_{i\in Y}\boldsymbol{f}_i(X) - \sum_i K_{ii}\boldsymbol{f}_i(X) \tag{6.56}$$

6.6.4 Approximating the MAP summary

In the context of a document summarization experiment (chapter 4) Kulesza and Taskar present an algorithm for approximating the MAP summary. In fact, it is a simple greedy algorithm, but they have some success with it. In the text, it is algorithm 6. I reproduce it here.

Algorithm 7 Greedy Algorithm for Approximately computing the MAP summary, reproduced from [9].

Input: Document cluster, X, parameter θ, and character limit b
$U \leftarrow \mathcal{Y}(X)$
$Y \leftarrow \emptyset$
while $U \neq \emptyset$ **do**
    $i \leftarrow \arg\max_{i' \in U} \left(\frac{\mathcal{P}_\theta(Y \cup \{i\}|X) - \mathcal{P}_\theta(Y|X)}{\text{length}(i)} \right)$
    $Y \leftarrow Y \cup \{i\}$
    $U \leftarrow U - (\{i\} \cup \{i' | \text{length}(Y) + \text{length}(i') > b\})$
end while
return summary Y

6.7 Engelhardt, Adams, 2014 - Bayesian Sparsity with Gaussian Fields

Paper: Barbara E. Engelhardt, Ryan P. Adams, *Bayesian Structured Sparsity from Gaussian Fields*. arXiv:1407.2235, Jul. 8, 2014.

Overview: Use a Bayesian model for sparsity that utilizes a Gaussian process. Models genetic data using an MCMC algorithm. The "sparsity-inducing prior on regression coefficients is a relaxation of the canonical spike-and-slab prior that flattens the mixture model into a scale mixture of normals."

Model:

- We assume there are n samples and p predictors; $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. Encoded as $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{X} \in \mathbb{R}^{n \times p}$.
- Response variables are conditionally independent, given predictors and some parameters:
  $\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \beta_0, \nu \sim \mathcal{N}(\beta_0 \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta}, \nu^{-1}\mathbb{I}_n)$
  - $\beta \in \mathbb{R}^p$ gives weights of each predictor.
  - $\nu > 0$ is the precision of the residuals.
  - $\beta_0 \in \mathbb{R} \sim \mathcal{N}(\mathbf{0}, (\lambda\nu)^{-1})$ gives the offset.
- Priors are given by the following.
  - Conjugate prior on $\nu \sim \Gamma(a_\nu, b_\nu)$.

  - Gaussian prior on $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, (\lambda\nu)^{-1}\boldsymbol{\Gamma})$.
    * $\lambda \sim \Gamma(a_\lambda, b_\lambda)$ is an inverse squared global scale parameter for regression weights.
    * $\boldsymbol{\Gamma}$ is a degenerate diagonal covariance matrix with inclusion variables on the diagonal, $\Gamma_{j,j} = z_j$.

- Sparsity in $\boldsymbol{\beta}$ comes from replacing z_i inclusion variables (which before were Bernoulli in the spike and slab method) with *probit link*.
  - The *probit link* is $\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$
  - $\boldsymbol{\Sigma}$ is a known covariance matrix used to specify dependence structure for inclusion.
  - Implication is that the diagonal of the covariance matrix $\boldsymbol{\Gamma}$ is given by $\Gamma_{j,j} = \mathbb{I}(\gamma_j > \gamma_0)$ where $\gamma_0 \sim \mathcal{N}(\mu_\gamma, v_\gamma)$.

- Probability of inclusion of predictor j is computed directly using the normal *cnf* function, Φ.
  $P(\beta_j \neq 0) = 1 - \Phi(\gamma_0/\Sigma_{j,j})$

MCMC Sampling:

- Closed form marginalization will be

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\gamma}, \gamma_0, \nu, \lambda) &= \int\int \mathcal{N}(\boldsymbol{y}|\beta_0\mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta}, \nu^{-1}\mathbb{I}_n)\mathcal{N}(\beta_0|0, (\nu\lambda)^{-1})\, d\boldsymbol{\beta}\, d\beta_0 \\ &= \int \mathcal{N}(\boldsymbol{y}|\beta_0\mathbf{1}_n, \nu^{-1}(\lambda^{-1}\boldsymbol{X}\boldsymbol{\Gamma}\boldsymbol{X}^\top + \mathbb{I}_n))\mathcal{N}(\beta_0|0, (\nu\lambda)^{-1})\, d\beta_0 \\ &= \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \nu^{-1}(\lambda^{-1}(\mathbf{1}_n\mathbf{1}_n^\top + \boldsymbol{X}\boldsymbol{\Gamma}\boldsymbol{X}^\top) + \mathbb{I}_n)) \end{aligned}$$

- Sampling of $\boldsymbol{\gamma}$ ...
- Updating λ, γ_0, and ν ...

Utilization of DPPs: see Notes **??**

Appendix A

Derivations of Distribution Closed Forms

Here I provide details of the mathematical derivations of the closed forms of distributions used in the above notes.

A.1 Regressor Coefficient Marginal Distribution, $p(\boldsymbol{\beta})$

Recall from §2.2 that our regression coefficients, $\boldsymbol{\beta}$, are dependent on the hyperparameters, a_0, b_0, c. Here we solve for the prior on $\boldsymbol{\beta}$ and present the implied constraints on the hyperparameters a_0, b_0, and c. Details on these parameters are also given in §2.2.

A.1.1 Marginalizing out σ^2

We begin with a standard marginalization equation.

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \int_{\mathbb{R}^+} p(\boldsymbol{\beta}|\sigma^2, c)p(\sigma^2|a_0, b_0)\, d\sigma^2 \tag{A.1}$$

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \int_{\mathbb{R}^+} \mathcal{N}\left(\boldsymbol{\beta}; 0, \sigma^2\frac{1}{c}\boldsymbol{I}_p\right)\Gamma^{-1}\left(\sigma^2; a_0, b_0\right)\, d\sigma^2 \tag{A.2}$$

I reduce the normal probability density as follows. First, recall that for some vector $\boldsymbol{x} \in \mathbb{R}^k$ drawn from a normal distribution,

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{A.3}$$

Here, $\boldsymbol{\mu} = 0$ and $\boldsymbol{\Sigma} = \sigma^2\Lambda_0^{-1} = \frac{\sigma^2}{c}\boldsymbol{I}_p$, implying

$$\boldsymbol{\Sigma}^{-1} = \frac{c}{\sigma^2}\boldsymbol{I}_p \qquad |\boldsymbol{\Sigma}| \triangleq \det(\boldsymbol{\Sigma}) = \left(\frac{\sigma^2}{c}\right)^p\det(\boldsymbol{I}_p) = \left(\frac{\sigma^2}{c}\right)^p \tag{A.4}$$

Thus, we reduce the first factor of the integrand:

$$\mathcal{N}\left(\boldsymbol{\beta}; \boldsymbol{0}, \sigma^2\frac{1}{c}\boldsymbol{I}_p\right) = \frac{1}{\sqrt{(2\pi\sigma^2/c)^p}}\exp\left(-\frac{1}{2}\boldsymbol{\beta}^\top\left(\frac{c}{\sigma^2}\right)\boldsymbol{I}_p\boldsymbol{\beta}\right) = \left(\frac{c}{2\pi}\right)^{\frac{p}{2}}\left(\sigma^2\right)^{-\frac{p}{2}}\exp\left[\left(\frac{c\boldsymbol{\beta}^\top\boldsymbol{\beta}}{2}\right)\frac{-1}{\sigma^2}\right] \tag{A.5}$$

The inverse gamma probability density term is given in its standard form:

$$\Gamma^{-1}\left(\sigma^2; a_0, b_0\right) = \frac{b_0^{a_0}}{\Gamma(a_0)}\left(\sigma^2\right)^{-a_0-1}\exp\left(-\frac{b_0}{\sigma^2}\right) \tag{A.6}$$

Thus, our integral reduces to the following form:

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \int_{\mathbb{R}+}\left(\frac{c}{2\pi}\right)^{\frac{p}{2}}\left(\sigma^2\right)^{-\frac{p}{2}}\exp\left[\left(\frac{c\boldsymbol{\beta}^\top\boldsymbol{\beta}}{2}\right)\frac{-1}{\sigma^2}\right]\frac{b_0^{a_0}}{\Gamma(a_0)}\left(\sigma^2\right)^{-a_0-1}\exp\left(-\frac{b_0}{\sigma^2}\right)d\sigma^2 \tag{A.7}$$

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \left(\frac{c}{2\pi}\right)^{p/2}\frac{b_0^{a_0}}{\Gamma(a_0)}\int_0^\infty\left[(\sigma^2)^{-a_0-1-p/2}\exp\left(-\frac{1}{\sigma^2}\left(\frac{c}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}+b_0\right)\right)\right]d\sigma^2 \tag{A.8}$$

I then define the following constants and rewrite equation A.8

$$k_0 \triangleq \left(\frac{c}{2\pi}\right)^{p/2}\frac{b_0^{a_0}}{\Gamma(a_0)} \qquad k_1 \triangleq -a_0-\frac{p}{2} \qquad k_2 \triangleq \left(\frac{c}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}+b_0\right) \tag{A.9}$$

$$p(\boldsymbol{\beta}|a_0, b_0, c) = k_0\int_0^\infty\left(\sigma^2\right)^{k_1-1}e^{-k_2\sigma^2}\,d\sigma^2 = \frac{k_0}{k_2^{k_1}}\Gamma(k_1) \tag{A.10}$$

Rewriting this in terms of our original variables, we reach the final result.

$$p(\boldsymbol{\beta}|a_0, b_0, c) = \left(\frac{c}{2\pi}\right)^{p/2}\frac{b_0^{a_0}}{\Gamma(a_0)}\left(\frac{c}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta}+b_0\right)^{a_0+p/2}\Gamma(-a_0-p/2) \tag{A.11}$$

Recalling that

$$\Gamma(n-1) = \frac{\Gamma(n)}{n-1} \tag{A.12}$$

We can infer the following.

$$\Gamma(n)\prod_{i=0}^{k-1}(n+i) = \Gamma(n+k) \;\Rightarrow\; \Gamma(-a_0-p/2) = \frac{\Gamma\left(\frac{\lceil a_0+p/2\rceil}{a_0+p/2}\right)}{\prod_{i=0}^{\lceil a_0+p/2-1\rceil}(-a_0-p/2+i)} \tag{A.13}$$

Providing an alternate representation of the (potentially) negative argument to the gamma function.

A.1.2 Implications for the Hyperparameters

Given that equation A.11 defines a probability, we must impose a few constraints on the hyperparameters.

As parameters to the Inverse Gamma Distribution, $a_0, b_0 > 0$. Also, as a value used to scale the covariance matrix for the prior distribution of $\boldsymbol{\beta}$, we also know that $c > 0$. Given this, if we consider the result from A.13, we see that $\lceil a_0 + p/2 - 1\rceil$ must have even parity. Further, $a_0 + p/2$ must not be an integer. These constraints ensure that $p(\boldsymbol{\beta}|a_0, b_0, c) \geq 0$.

We must also ensure that $p(\boldsymbol{\beta}|a_0, b_0, c) \leq 1$. Using the result from equation A.13 and

assuming the parity constraints on $\lceil a_0 + p/2 \rceil$ described above, we can write this constraint as the following.

$$c^{p/2} b_0^{a_0} \left(\frac{c}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} + b_0 \right)^{a_0 + p/2} \Gamma\left(\frac{\lceil a_0 + p/2 \rceil}{a_0 + p/2} \right) \leq (2\pi)^{p/2} \, \Gamma(a_0) \prod_{i=0}^{\lceil a_0 + p/2 - 1 \rceil} (a_0 + p/2 - i) \quad \text{(A.14)}$$

While there are choices of a_0, b_0, and c that could invalidate this constraint, given the large value of p in our domain, the product on the right side of equation A.14 will likely ensure its validity.

A.2 Marginalizing Regressors: An Analytic Solution for $p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2)$

Here I provide a derivation of the analytic solution to the conditional probability of $p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2)$ by marginalizing out $\boldsymbol{\beta}$. The probability is given by the following integral. All variables refer to those described in §2.2.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2) = \int_{\mathbb{R}^p} p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) p(\boldsymbol{\beta}|\sigma^2; \Lambda_0) \, d\boldsymbol{\beta} \quad \text{(A.15)}$$

A.2.1 Conditional distribution of $\boldsymbol{y}$

As discussed in [3] §3.1, the distribution of a linear regressor target, given its regressors, their coefficients, and a normally distributed noise term is a product of n normal distributions. In our case:

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) = \prod_{i=1}^{n} \mathcal{N}\left(y_i; x_i^\top (\boldsymbol{\gamma} \odot \boldsymbol{\beta}), \sigma^2\right) \quad \text{(A.16)}$$

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - x_i^\top \boldsymbol{\beta}_{\boldsymbol{\gamma}})^2}{2\sigma^2} \right) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\sum_{i=0}^{n} (y_i - x_i^\top \boldsymbol{\beta}_{\boldsymbol{\gamma}})^2}{2\sigma^2} \right) \quad \text{(A.17)}$$

We can rewrite the sum in the argument of the exponential function as the following inner product.

$$\sum_{i=0}^{n} \left(y_i - x_i^\top \boldsymbol{\beta}_{\boldsymbol{\gamma}} \right)^2 = \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_{\boldsymbol{\gamma}} \right)^\top \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_{\boldsymbol{\gamma}} \right) \quad \text{(A.18)}$$

Since we will ultimate by integrating over $\boldsymbol{\beta}$, we want to eliminate the dependance of $\boldsymbol{\beta}$ on $\boldsymbol{\gamma}$. Fortunately, we notice the following.

$$\boldsymbol{X} \left(\boldsymbol{\beta} \odot \boldsymbol{\gamma} \right) = \left(\boldsymbol{X} \operatorname{Diag}(\boldsymbol{\gamma}) \right) \boldsymbol{\beta} \quad \text{(A.19)}$$

Using the notation $\boldsymbol{X}_{\boldsymbol{\gamma}} \triangleq \boldsymbol{X} \operatorname{Diag}(\boldsymbol{\gamma})$, we rewrite the conditional probability.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{\beta}}{2\sigma^2} \right) \quad \text{(A.20)}$$

A.2.2 Conditional distribution of $\boldsymbol{\beta}$

(Reproduced from §2.3.3) As discussed in §2.2, we take $\boldsymbol{\beta}$ to be distributed normally. Given σ^2, we found the value of the multivariate normal density function in equation A.5, reproduced here:

$$p(\boldsymbol{\beta}|\sigma^2) = \mathcal{N}\left(\boldsymbol{\beta}; \mathbf{0}, \frac{\sigma^2}{c}\boldsymbol{I}_p\right) = \left(\frac{c}{2\pi\sigma^2}\right)^{\frac{p}{2}} \exp\left[\left(\frac{c\boldsymbol{\beta}^\top\boldsymbol{\beta}}{2}\right)\frac{-1}{\sigma^2}\right] \tag{A.21}$$

A.2.3 Solution

Combining the results of §A.2.1 and §A.2.2, we find

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{X}) = \int_{\mathbb{R}^p} \frac{c^{p/2}}{(2\pi\sigma^2)^{\frac{n+p}{2}}} \exp\left(-\frac{\boldsymbol{y}^\top\boldsymbol{y} + c\boldsymbol{\beta}^\top\boldsymbol{\beta} - 2\boldsymbol{y}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\beta}}{2\sigma^2}\right) d\boldsymbol{\beta} \tag{A.22}$$

We define the following $\boldsymbol{\beta}$-independent variables and rewrite the problem.

$$k \triangleq \frac{c^{p/2}\exp\left(-\frac{\boldsymbol{y}^\top\boldsymbol{y}}{2\sigma^2}\right)}{(2\pi\sigma^2)^{\frac{n+p}{2}}} \qquad \boldsymbol{A} \triangleq \frac{\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p}{\sigma^2} \qquad \boldsymbol{J} \triangleq \frac{\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{y}}{\sigma^2} \tag{A.23}$$

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{X}) = k\int_{\mathbb{R}^p} \exp\left(-\frac{1}{2}\boldsymbol{\beta}^\top\boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{J}^\top\boldsymbol{\beta}\right) d\boldsymbol{\beta} = \frac{k\,(2\pi)^{p/2}}{\sqrt{\det(\boldsymbol{A})}} \exp\left(\frac{1}{2}\boldsymbol{J}^\top\boldsymbol{A}^{-1}\boldsymbol{J}\right) \tag{A.24}$$

This yields the full solution

$$p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2) = \frac{c^{p/2}\exp\left(\frac{-1}{2\sigma^2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}\right)}{(2\pi\sigma^2)^{n/2}\sqrt{\det\left[\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right]}} \tag{A.25}$$

A.3 Marginalizing Regressors and Variance: An Analytic Solution for $p(\boldsymbol{y}|\boldsymbol{\gamma})$

Here I provide a derivation of the analytic solution to the conditional probability of $p(\boldsymbol{y}|\boldsymbol{\gamma})$ by marginalizing out $\boldsymbol{\beta}$ and σ^2. The integral, as given in equation 2.18 is reproduced here. All variables refer to those described in §2.2.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \int_{\mathbb{R}^+}\int_{\mathbb{R}^p} p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{X})p(\boldsymbol{\beta}|\sigma^2; \Lambda_0)p(\sigma^2|a_0, b_0)\, d\boldsymbol{\beta}\, d\sigma^2 \tag{A.26}$$

A.3.1 Solution

Combining result from §A.2 with the prior on σ^2 and rewriting to separate σ^2, we arrive at the following integral.

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \frac{b_0^{a_0} c^{p/2}}{\Gamma(a_0)(2\pi)^{n/2}\left(\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)\right)^{1/2}} \cdot \int_{\mathbb{R}^+} \left(\frac{1}{\sigma^2}\right)^{n/2+a_0+1} \exp\left(\frac{-\frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y} - b_0}{\sigma^2}\right) d\sigma^2 \tag{A.27}$$

Now let me define the following constants and rewrite the full expression.

$$\alpha_0 \triangleq \frac{b_0^{a_0} c^{p/2}}{\Gamma(a_0)(2\pi)^{n/2}\sqrt{\left|\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right|}} \tag{A.28}$$

$$\alpha_1 \triangleq b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y} \tag{A.29}$$

$$k \triangleq \frac{n}{2} + a_0 + 1 \qquad u \triangleq \sigma^2 \tag{A.30}$$

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \alpha_0 \int_{\mathbb{R}^+} \frac{1}{u^k}\exp\left(\frac{-\alpha_1}{u}\right) du \tag{A.31}$$

The integral in equation A.31 converges iff $\alpha_1 > 0$. The acceptability of this constraint and its implications on the hyperparameters are discussed in §A.3.2. Taking this assumption as valid, we make the following substitutions and solve the integral in equation A.31:

$$x = \frac{\alpha_1}{u} \qquad dx = -\frac{\alpha_1}{u^2}\, du \tag{A.32}$$

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \frac{\alpha_0}{\alpha_1}\int_\infty^0 \frac{1}{u^{(k-2)}}\left(\frac{-\alpha_1}{u^2}\right)\exp\left(\frac{-\alpha_1}{u}\right) du \tag{A.33}$$

Making substitutions for x and dx and flipping the integration bounds (after applying $x = \alpha_1/u$), we find

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \frac{\alpha_0}{\alpha_1}\int_0^\infty \left(\frac{x}{\alpha_1}\right)^{k-2}\exp(-x)\, dx = \frac{\alpha_0}{\alpha_1^{k-1}}\Gamma(k-1) \tag{A.34}$$

To ensure that the above is always positive and defined, we add the constraint

$$b_0 \geq \left(\frac{\lambda_{\max} - 1}{2}\right)\boldsymbol{y}^\top\boldsymbol{y} \tag{A.35}$$

where $\lambda_{\max}$ is the first eigenvalue of $\boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X} + \Lambda_0\right)^{-1}\boldsymbol{X}^\top$. Proof is given in §A.3.2. The full result is given below in terms of all original variables

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) = \frac{b_0^{a_0} c^{p/2}\Gamma\left(\frac{n}{2} + a_0\right)}{\Gamma(a_0)(2\pi)^{n/2}\sqrt{\left|\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right|}\cdot\left(b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}\right)^{a_0+n/2}} \tag{A.36}$$

A.3.2 Implications for the Hyperparameters

In order for the integral in equation A.31 to converge, we imposed the requirement that $\alpha_1 > 0$, where we had defined α_1 as

$$\alpha_1 \triangleq b_0 + \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y} \;\Rightarrow \tag{A.37}$$

$$b_0 + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{y} > \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \tag{A.38}$$

If the constraint in equation A.38 does not hold, then the bounds in the integral in equation A.34 become $[0, -\infty]$ instead of $[0, \infty]$, which does not converge to the Gamma function.

We can enforce this constraint by the following. First, I make the following definition to simplify the expressions.

$$\boldsymbol{P} \triangleq \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \tag{A.39}$$

$$b_0 + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{y} > \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{P} \boldsymbol{y} \tag{A.40}$$

Thus, we are really interested in the relationship between the $\|\cdot\|_2$ norm of $\boldsymbol{y}$ and its projective norm, $\|\cdot\|_{\boldsymbol{P}}$. With $\boldsymbol{X}$ given as a general data matrix, I have no certainty that $\boldsymbol{P}$ will be normalized, but I will almost assuredly be standardizing it. Thus, in all likelihood, $\boldsymbol{y}^\top \boldsymbol{y} \geq \boldsymbol{y}^\top \boldsymbol{P} \boldsymbol{y}$, and we will not have to worry about setting any stricter lower bound on b_0.

In the event that $\boldsymbol{y}^\top \boldsymbol{y} < \boldsymbol{y}^\top \boldsymbol{P} \boldsymbol{y}$, we will can still enforce the constraint in A.38 by choosing b_0 to be above the necessary threshold. Here is a method for choosing b_0 such that the constraint is held for arbitrary selection of $\boldsymbol{\gamma}$. First, we note the following:

$$\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} = \boldsymbol{y}^\top Q_{\boldsymbol{\gamma}} \Lambda_{\boldsymbol{\gamma}} Q_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \leq \lambda_{\max_{\boldsymbol{\gamma}}} \|\boldsymbol{y}\|_2^2 \tag{A.41}$$

Where $Q_{\boldsymbol{\gamma}} \Lambda_{\boldsymbol{\gamma}} Q_{\boldsymbol{\gamma}}^\top$ is the eigendecomposition of $\boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top$, and $\lambda_{\boldsymbol{\gamma}\max}$ is its the largest eigenvalue. To find a bound independent of $\boldsymbol{\gamma}$, we note that since $\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right)$ is symmetric, we know its inverse is symmetric, and thus its product with $\boldsymbol{X}_{\boldsymbol{\gamma}}$ on the left and $\boldsymbol{X}_{\boldsymbol{\gamma}}^\top$ on the right is also symmetric. Being symmetric, we can apply the interlacing inequalities (given in [10] p. 219). They tell us that for a given hermitian matrix $A \in \mathbb{R}^{n \times n}$, if we write its eigenvalues as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and we consider the matrix $A_i \in \mathbb{R}^{n-1 \times n-1}$ formed by removing the ith row and and column from A, if we write its eigenvalues as $\lambda_{i1} \leq \lambda_{i2} \leq \ldots \lambda_{ik}$, we are assured that $\lambda_{k+1} \geq \lambda_{ik} \, \forall \, i \in [1, n]$. Therefore, we can write the following bound:

$$\lambda_{\max_{\boldsymbol{\gamma}}} \leq \lambda_{\max_{\mathbf{1}_p}} \tag{A.42}$$

Where $\lambda_{\max_{\mathbf{1}_p}}$ is the first eigenvalue of the parent matrix $\boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X} + \Lambda_0\right)^{-1} \boldsymbol{X}^\top$ (corresponding to $\boldsymbol{\gamma} = \mathbf{1}_M$). we can further bound by $\lambda_{\max} \|\boldsymbol{y}\|_2^2$, where $\lambda_{\max}$ is the largest eigenvalue of $Q \Lambda Q^\top = \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X} + \Lambda_0\right)^{-1} \boldsymbol{X}^\top$. Thus, we can adjust b_0 in order to guarantee that the constraint in A.38 is satisfied:

$$b_0 \geq \left(\frac{\lambda_{\max} - 1}{2}\right) \boldsymbol{y}^\top \boldsymbol{y} \tag{A.43}$$

Note however that this is not our exact setting; $\boldsymbol{X}_{\boldsymbol{\gamma}}$ is not a submatrix of $\boldsymbol{X}$, but rather $\boldsymbol{X}$ itself with certain rows and columns set to zeros according to $\boldsymbol{\gamma}$. However, as derived below in §A.3.3, when we add a constant matrix to this, the determinant of the result is equivalent to that of the submatrix up to a constant. We can guarantee that the setting presented in this section is an upper bound if we constrain $c \geq 1$.

A.3.3 Computational Note

For many of these equations, we must compute the determinant of matrices indexed on $\boldsymbol{\gamma}$. Of course, the case of $\det[\boldsymbol{M}\,\mathrm{Diag}(\boldsymbol{\gamma})]$ is trivially zero for any $\|\boldsymbol{\gamma}\|_1 < n$. However, in the case where we add some diagonal term to square $\boldsymbol{\gamma}$-indexed matrices, we can simplify the calculation from the brute-force method. The inspiring example is in the denominator of $p(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\theta})$ in equation 2.19, $\left|\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right|$.

In the given example, we have a symmetric matrix, where in the rows and columns denoted by $\boldsymbol{\gamma}$, we have only a single constant on the diagonal as a result of adding Λ_0^{-1}. Recall the equation for calculating the determinant of a matrix A, given as a sum of the products between elements $\{a_{i1}, \ldots, a_{ij}, \ldots, a_{im}\}$ in row i and the determinants of A_{ij}, the submatrices of A created by removing row i and column j.

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij}) \tag{A.44}$$

When $a_{ij} = 0 \,\forall\, i \neq j$, as is the case for the i indexed by zero entries of $\boldsymbol{\gamma}$, we can simplify the expression to

$$\det(A) = a_{ii} \det(A_{ii}) \tag{A.45}$$

We can continue to make this reduction for as many zero entries as there are in $\boldsymbol{\gamma}$. Thus, we denote the submatrix of $\boldsymbol{X}$ with the columns indicated by $\boldsymbol{\gamma}_i = 0$ actually removed instead of just zeroed as $\tilde{\boldsymbol{X}}_{\boldsymbol{\gamma}}$. The final result is

$$\left|\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + \Lambda_0\right| = c^{p - \|\boldsymbol{\gamma}\|_1} \left|\tilde{\boldsymbol{X}}_{\boldsymbol{\gamma}}{}^\top \tilde{\boldsymbol{X}}_{\boldsymbol{\gamma}} + \Lambda_0\right| \tag{A.46}$$

Appendix B

Derivations of Likelihoods and Gradients

Here I provide details of the mathematical derivations of the likelihood functions and their gradients as used in the iterative experiments used in the above notes.

B.1 Partial Pre-Marginalization

In this section I present likelihood functions and their gradients as used in the variational solution to the problem described in §3.2. To simplify computation, when solving for the likelihood functions of a variable, I assume the likelihood function $\mathcal{L}(\phi)$ to be any function that satisfies equation 3.14, reproduced below.

$$\arg\max_{\phi\in\boldsymbol{\Phi}} \mathcal{P}(\phi|\,\mathrm{MB}(\phi)) = \arg\max_{\phi\in\boldsymbol{\Phi}} \mathcal{L}(\phi) \tag{B.1}$$

B.1.1 Partial Pre-Marginalization: Likelihood of σ^2

Using the partial pre-marginalization of our model, we can imagine our sparsity problem as modeled by figure 3.1. Here, the markov blanket of σ^2 is $\{\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}\}$. Thus, we can write

$$\arg\max_{\sigma^2\in\mathbb{R}^+} \mathcal{P}\left(\sigma^2|\,\mathrm{MB}(\sigma^2)\right) = \arg\max_{\sigma^2\in\mathbb{R}^+} \frac{p(\boldsymbol{y}|\sigma^2,\boldsymbol{\gamma},\boldsymbol{X})\,p(\sigma^2)}{\int_{\mathbb{R}^+} p(\boldsymbol{y}|\sigma^2,\boldsymbol{\gamma},\boldsymbol{X})\,p(\sigma^2)\,d\sigma^2} \tag{B.2}$$

Then we remove the normalization constant and substitute in the result for $p(\boldsymbol{y}|\sigma^2,\boldsymbol{\gamma},\boldsymbol{X})$ from §A.2 and the inverse gamma prior for σ^2 as described in equation 2.7.

$$= \arg\max_{\sigma^2\in\mathbb{R}^+} \frac{c^{p/2}\exp\left(\frac{-1}{2\sigma^2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}\right)}{(2\pi\sigma^2)^{n/2}\sqrt{\det\left[\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right]}} \cdot \frac{b_0^{a_0}}{\Gamma(a_0)\left(\sigma^2\right)^{a_0+1}}\exp\left(-\frac{b_0}{\sigma^2}\right) \tag{B.3}$$

We then remove all σ^2-independent factors and take the logarithm.

$$\arg\max_{\sigma^2\in\mathbb{R}^+} \mathcal{P}\left(\sigma^2|\,\mathrm{MB}(\sigma^2)\right) = \arg\max_{\sigma^2\in\mathbb{R}^+} -(a_0+1+n/2)\log\left(\sigma^2\right) - \frac{k}{\sigma^2} \tag{B.4}$$

$$k = b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y} \tag{B.5}$$

Thus, we define the likelihood function and its gradient as the following.

$$\mathcal{L}(\sigma^2) = \frac{-k}{\sigma^2} - (a_0 + 1 + n/2)\log\left(\sigma^2\right) \tag{B.6}$$

$$\nabla\mathcal{L}(\sigma^2) = \frac{k}{\left(\sigma^2\right)^2} - \frac{a_0 + 1 + n/2}{\sigma^2} \tag{B.7}$$

Where we are taking the gradient with respect to σ^2, not σ.

B.1.2 Partial Pre-Marginalization: Likelihood of $\boldsymbol{\gamma}$

Using the partial pre-marginalization of our model, we can imagine our sparsity problem as modeled by figure 3.1. Here, the markov blanket of $\boldsymbol{\gamma}$ is $\{\sigma^2, \boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{X}\}$. Thus, we can write

$$\arg\max_{\boldsymbol{\gamma}\in\{0,1\}^p} \mathcal{P}(\boldsymbol{\gamma}|\,\mathrm{MB}(\boldsymbol{\gamma})) = \arg\max_{\boldsymbol{\gamma}\in\{0,1\}^p} \frac{p(\boldsymbol{y}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{X})\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \boldsymbol{X})}{\sum_{\boldsymbol{\gamma}'\in\{0,1\}^p} p(\boldsymbol{y}|\boldsymbol{\gamma}', \sigma^2, \boldsymbol{X})\, p(\boldsymbol{\gamma}'|\boldsymbol{\theta}, \boldsymbol{X})} \tag{B.8}$$

We can then drop the normalization and replace the first term with the result in equation A.25 and the second with equation 2.3. Note, that we are using equation 2.3 and not equation 2.4, since the normalization term does not influence the arg max operator. Taking the logarithm and dropping all $\boldsymbol{\gamma}$-independent terms, we find the following equivalence.

$$\begin{aligned}\arg\max_{\boldsymbol{\gamma}\in\{0,1\}^p} \mathcal{P}(\boldsymbol{\gamma}|\,\mathrm{MB}(\boldsymbol{\gamma})) = \arg\max_{\boldsymbol{\gamma}\in\{0,1\}^p} &\sum_{i\in\boldsymbol{\gamma}} \boldsymbol{\theta}_i + \log\left(\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}}\right)\right) \\ &- \frac{1}{2\sigma^2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y} - \frac{1}{2}\log\left(\det\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)\right)\end{aligned} \tag{B.9}$$

Again, we know $\nabla \log p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \boldsymbol{X})$, but the gradient for the last two terms is not well defined. Further, the gradient in the context of a discrete boolean vector is not well defined in general. We will still have to define a more meaningful likelihood and gradient for $\boldsymbol{\gamma}$, even in the partial pre-marginalization setting.

B.1.3 Likelihood of $\boldsymbol{\theta}$

Irrespective of the pre-marginalization of our model, i.e. if we use figure 3.2 or figure 3.1, the markov blanket of $\boldsymbol{\theta}$ is $\{\boldsymbol{\gamma}, \boldsymbol{X}\}$. Thus, we can write

$$\arg\max_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathcal{P}(\boldsymbol{\theta}|\,\mathrm{MB}(\boldsymbol{\theta})) = \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \boldsymbol{X})\, p(\boldsymbol{\theta}|\boldsymbol{X})}{\int_{\mathbb{R}^p} p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \boldsymbol{X})\, p(\boldsymbol{\theta}|\boldsymbol{X})\, d\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^p} \log p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \boldsymbol{X}) + \log p(\boldsymbol{\theta}|\boldsymbol{X}) \tag{B.10}$$

Thus, we define the likelihood function as

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \boldsymbol{X}) + \log p(\boldsymbol{\theta}|\boldsymbol{X}) \tag{B.11}$$

If we define only a uniform prior on $\boldsymbol{\theta}$, then the last term becomes irrelevant. I will begin my simulations with the uniform assumption, but this will be an area of potential development. Using the result in §6.5.4 with $\boldsymbol{f}_i = e_i$, we arrive at the following gradient.

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\gamma} - \mathrm{diag}(K) + \nabla \log p(\boldsymbol{\theta}|\boldsymbol{X}) \tag{B.12}$$

Again, where depending on our chose of $\boldsymbol{\theta}$'s prior, the final term may become irrelevant.

B.2 Maximal Pre-Marginalization

In this section I present likelihood functions and their gradients used in the variational solution to the problem described in §3.3. I use the same conception of the likelihood function $\mathcal{L}(\phi)$ as above that it can be any function that satisfies equation 3.14.

B.2.1 Maximum Pre-Marginalization: Likelihood of a_0

Using the maximal pre-marginalization of our model, we can imagine our sparsity problem as modeled by figure 3.2. Here, the markov blanket of a_0 is $\{b_0, c, \boldsymbol{\gamma}, \boldsymbol{y}, \boldsymbol{X}\}$. As a hyperparameter, we use the uninformative Jeffreys prior $\pi(a_0)$ (equation 2.41) over its domain, $\mathbb{R}^+$ [15]. Thus, we can make the following reductions.

$$\mathcal{P}(a_0|\,\mathrm{MB}(a_0)) = p(a_0|b_0, c, \boldsymbol{\gamma}, \boldsymbol{y}, \boldsymbol{X}) \tag{B.13}$$

$$\begin{aligned}
\arg\max_{a_0\in\mathbb{R}^+} \mathcal{P}(a_0|\,\mathrm{MB}(a_0)) &= \arg\max_{a_0\in\mathbb{R}^+} \frac{p(\boldsymbol{y}|a_0, b_0, c, \boldsymbol{\gamma}, \boldsymbol{X})\, p(a_0|b_0, c, \boldsymbol{\gamma}, \boldsymbol{X})}{\int_{\mathbb{R}^+} p(\boldsymbol{y}|a_0, b_0, \boldsymbol{\gamma}, \boldsymbol{X})\, p(a_0|b_0, \boldsymbol{\gamma}, \boldsymbol{X})\, da_0} && \text{(B.14)}\\
&= \arg\max_{a_0\in\mathbb{R}^+} p(\boldsymbol{y}|a_0, b_0, c, \boldsymbol{\gamma}, \boldsymbol{X})\, \pi(a_0) && \text{(B.15)}\\
&= \arg\max_{a_0\in\mathbb{R}^+} \log p(\boldsymbol{y}|a_0, b_0, c, \boldsymbol{\gamma}, \boldsymbol{X}) + \log \pi(a_0) && \text{(B.16)}
\end{aligned}$$

Using the result in equation A.36, we define the likelihood:

$$\log p(\boldsymbol{y}|a_0, b_0, \boldsymbol{\gamma}, \boldsymbol{X}) = a_0 k_1 + k_2 + \log\left[\Gamma\left(n/2 + a_0\right)\right] - \log\Gamma(a_0) \tag{B.17}$$

$$\mathcal{L}(a_0) = a_0 k_1 + \log\left[\Gamma(n/2 + a_0)\right] - \log\Gamma(a_0) + \frac{1}{2}\log\left(a_0\psi^{(1)}(a_0) - 1\right) \tag{B.18}$$

Where $\psi^{(i)}$ is the ith polygamma function, k_2 includes all a_0-independent terms, and k_1 is defined as follows.

$$k_1 \triangleq \log\left(\frac{b_0}{b_0 + \frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}}\left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}\right)^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right)\boldsymbol{y}}\right) \tag{B.19}$$

Thus, the gradient is given by

$$\nabla\mathcal{L}(a_0) = k_1 + \psi^{(0)}\left(\frac{n}{2} + a_0\right) - \psi^{(0)}(a_0) + \frac{1}{2}\left(\frac{\psi^{(1)}(a_0) + a_0\psi^{(2)}(a_0)}{a_0\psi^{(1)}(a_0) - 1}\right) \tag{B.20}$$

Again, where $\psi^{(i)}$ is the ith polygamma function.

B.2.2 Maximum Pre-Marginalization: Likelihood of b_0

Using the maximal pre-marginalization of our model, we can imagine our sparsity problem as modeled by figure 3.2. Here, the markov blanket of b_0 is $\{a_0, c, \boldsymbol{\gamma}, \boldsymbol{y}, \boldsymbol{X}\}$. As a hyperparameter, we use the uninformative Jeffreys prior, $\pi(b_0)$ (equation 2.40) over its

domain, $\mathcal{B} = \{x \in \mathbb{R} | x \geq \max(0, \frac{1}{2}(\lambda_{\max} - 1)\boldsymbol{y}^\top \boldsymbol{y})\}$. Utilizing its similarity to a_0, we rewrite equation B.16 in a form that apply to b_0 here.

$$\arg\max_{b_0 \in \mathcal{B}} \mathcal{P}(b_0 | \operatorname{MB}(b_0)) = \arg\max_{b_0 \in \mathcal{B}} \log p(\boldsymbol{y} | a_0, b_0, c, \boldsymbol{\gamma}, \boldsymbol{X}) + \log \pi(b_0) \tag{B.21}$$

Thus we can write the following likelihood function for b_0.

$$\log p(\boldsymbol{y} | a_0, b_0, c, \boldsymbol{\gamma}, \boldsymbol{X}) = k + a_0 \log b_0 - \left(a_0 + \frac{n}{2}\right) \log \left(b_0 + \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}\right) \tag{B.22}$$

$$\mathcal{L}(b_0) = (a_0 + 1) \log b_0 - \left(a_0 + \frac{n}{2}\right) \log \left(b_0 + \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}\right) \tag{B.23}$$

Where

$$k = \log \Gamma(a_0 + n/2) - \log \Gamma(a_0) - \frac{n}{2} \log(2\pi) - \frac{1}{2} \log \left(\det \left[\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}\right]\right) \tag{B.24}$$

Note, the addition of 1 in the coefficient of $\log b_0$ in the first term of $\mathcal{L}(b_0)$ is accounting for the addition of the Jeffreys prior on b_0. The gradient is given by

$$\nabla \mathcal{L}(b_0) = \frac{a_0 + 1}{b_0} - \left(\frac{a_0 + \frac{n}{2}}{b_0 + \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}}\right) \tag{B.25}$$

B.2.3 Maximum Pre-Marginalization: Likelihood of c

Using the maximal pre-marginalization of our model, we can imagine our sparsity problem as modeled by figure 3.2. Here, the markov blanket of c is $\{a_0, b_0, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\gamma}\}$. As a hyperparameter, we use the uninformative Jeffreys prior, $\pi(c)$ (equation 2.49) over its domain, $\mathbb{R}^+$. Utilizing its similarity to a_0, we rewrite equation B.16 in a form that applies to c.

$$\arg\max_{c \in \mathcal{R}^+} \mathcal{P}(c | \operatorname{MB}(c)) = \arg\max_{c \in \mathcal{R}^+} \log p(\boldsymbol{y} | a_0, b_0, c, \boldsymbol{\gamma}, \boldsymbol{X}) + \log \pi(c) \tag{B.26}$$

Dropping all of the c-independent terms, the above reduces to the following likelihood.

$$\begin{aligned}\mathcal{L}(c) = \left(\frac{p^2}{2} + p\right) \log(c) - \frac{1}{2} \log \det \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right) \\ - \left(a_0 + \frac{n}{2}\right) \log \left(b_0 + \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top\right) \boldsymbol{y}\right)\end{aligned} \tag{B.27}$$

The gradient for the first term is trivial. For use in the second two terms, let us use the eigendecomposition notation $\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$. Thus for the second term, we find the following partial derivative.

$$\frac{\partial}{\partial c} \log \det \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p\right) = \frac{\partial}{\partial c} \log \det \left(\boldsymbol{Q} \left[\boldsymbol{\Lambda} + c\boldsymbol{I}_p\right] \boldsymbol{Q}^{-1}\right) = \frac{\partial}{\partial c} \log \left(\det (\boldsymbol{Q}) \det (\boldsymbol{\Lambda} + c\boldsymbol{I}_p) \left(\frac{1}{\det (\boldsymbol{Q})}\right)\right) \tag{B.28}$$

$$= \frac{\partial}{\partial c} \log \left(\prod_{i=1}^{p} (\lambda_i + c) \det \boldsymbol{I}_p \right) = \frac{\partial}{\partial c} \sum_{i=1}^{p} \log (\lambda_i + c) = \sum_{i=1}^{p} \frac{1}{\lambda_i + c} \tag{B.29}$$

When taking the gradient of the third term, we will need to find the following partial.

$$\frac{\partial}{\partial c} \left(\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \right) \tag{B.30}$$

Using the same eigendecomposition, we reduce this to the following.

$$\frac{\partial}{\partial c} \left(\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{Q} \left[\boldsymbol{\Lambda} + c\boldsymbol{I}_p \right] \boldsymbol{Q}^{-1} \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \right) = \frac{\partial}{\partial c} \left(\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{Q} \left[\boldsymbol{\Lambda} + c\boldsymbol{I}_p \right]^{-1} \boldsymbol{Q}^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \right) \tag{B.31}$$

Let us define the matrix $\boldsymbol{A} \triangleq \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{Q}$ where the ith column of $\boldsymbol{A}$ is denoted by $\boldsymbol{a}_i$. We can rewrite the result in equation B.31 as the following.

$$\frac{\partial}{\partial c} \left(\sum_{i=1}^{p} \boldsymbol{a}_i^\top \boldsymbol{y} \left(\frac{1}{\lambda_i + c} \right) \boldsymbol{a}_i^\top \boldsymbol{y} \right) = - \sum_{i=1}^{p} \left(\frac{\boldsymbol{a}_i^\top \boldsymbol{y}}{\lambda_i + c} \right)^2 \tag{B.32}$$

Which we note is equivalent to our original expression, up to a negative and with eigenvalues $\lambda_i + c \mapsto (\lambda_i + c)^2$. Thus, we can write the result in matrix form.

$$\frac{\partial}{\partial c} \left(\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \right) = -\boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{Q} \left(\boldsymbol{\Lambda} + c\boldsymbol{I}_p \right)^{-2} \boldsymbol{Q}^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y} \tag{B.33}$$

Using the results in equations B.29 and B.33, we arrive at the full gradient.

$$\nabla \mathcal{L}(c) = \frac{p\left(p + \frac{1}{2}\right)}{c} - \frac{1}{2} \sum_{i=1}^{p} \frac{1}{\lambda_i + c} - \left(a_0 + \frac{n}{2} \right) \frac{\frac{1}{2} \boldsymbol{y}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{Q} \left(\boldsymbol{\Lambda} + c\boldsymbol{I}_p \right)^{-2} \boldsymbol{Q}^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{y}}{b_0 + \frac{1}{2} \boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I}_p \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y}} \tag{B.34}$$

B.2.4 Maximum Pre-Marginalization: Likelihood of $\boldsymbol{\gamma}$

Using the maximal pre-marginalization of our model, we can imagine our sparsity problem as modeled by figure 3.2. Here, the markov blanket of $\boldsymbol{\gamma}$ is $\{a_0, b_0, \boldsymbol{\theta}, \boldsymbol{y}, \boldsymbol{X}\}$. Thus, we can write

$$\arg \max_{\boldsymbol{\gamma} \in \{0,1\}^p} \mathcal{P}(\boldsymbol{\gamma} | \operatorname{MB}(\boldsymbol{\gamma})) = \arg \max_{\boldsymbol{\gamma} \in \{0,1\}^p} \frac{p(\boldsymbol{y} | \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{X}, a_0, b_0)\, p(\boldsymbol{\gamma} | \boldsymbol{\theta}, \boldsymbol{X})}{\sum_{\boldsymbol{\gamma}' \in \{0,1\}^p} p(\boldsymbol{y} | \boldsymbol{\gamma}', \boldsymbol{\theta}, a_0, b_0, \boldsymbol{X})\, p(\boldsymbol{\gamma}' | \boldsymbol{\theta}, \boldsymbol{X})} \tag{B.35}$$

We first substitute in the result from equation A.36. Then, dropping the normalization sum, taking the logarithm, and dropping all $\boldsymbol{\gamma}$-independent terms, we can write the following likelihood.

$$\begin{aligned} \mathcal{L}(\boldsymbol{\gamma}) = \sum_{i \in \boldsymbol{\gamma}} \boldsymbol{\theta}_i + \log \left(\det \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} \right) \right) - \frac{1}{2} \log \left(\det \left[\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I} \right] \right) \\ - \left(a_0 + \frac{n}{2} \right) \log \left(b_0 + \frac{1}{2} \boldsymbol{y}^\top \left(\boldsymbol{I}_n - \boldsymbol{X}_{\boldsymbol{\gamma}} \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^\top \boldsymbol{X}_{\boldsymbol{\gamma}} + c\boldsymbol{I} \right)^{-1} \boldsymbol{X}_{\boldsymbol{\gamma}}^\top \right) \boldsymbol{y} \right) \end{aligned} \tag{B.36}$$

The gradients here are not well defined. In fact, the very concept of a gradient is not defined for $\boldsymbol{\gamma}$, as it is not a continuous variable. Thus, I will have to investigate different ways of optimizing this likelihood.

Bibliography

[1] Raja Hafiz Affandi, Emily B. Fox, Ryan P. Adams, and Ben Taskar. Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning (ICML)*, volume 32, 2014.

[2] N. K. Batmanghelich, G. Quon, A. Kulesza, M. Kellis, P. Golland, and L. Bornn. Diversifying sparsity using variational determinantal point processes. *arXiv*, 2014.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] Barbara E. Engelhardt and Ryan P. Adams. Bayesian structured sparsity from gaussian fields. *arXiv*, 2014.

[5] Jennifer A Gillenwater, Alex Kulesza, Emily Fox, and Ben Taskar. Expectation-maximization for learning determinantal point processes. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3149–3157. Curran Associates, Inc., 2014.

[6] Harold Jeffreys. *Theory of Probability*. Oxford University Press, 3 edition, 1967.

[7] Mutsuki Kojima and Fumiyasu Komaki. Determinantal point process priors for bayesian variable selection in linear regression. *arXiv*, 2014.

[8] Alex Kulesza and Ben Taskar. Learning determinantal point processes. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, Barcelona, Spain, July 2011.

[9] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *arXiv*, 2013.

[10] Marvin Marcus and Henryk Minc. *A survey of matrix theory and matrix inequalities*. Allyn and Bacon, 1964.

[11] Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

[12] T. Salimans and D. A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *arXiv*, 2014.

[13] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.

[14] Gertraud Malsiner Walli. Bayesian variable selection in normal regression models. Master's thesis, Institut für Angewandte Statistik Johannes Kepler Universität Linz, 2010.

[15] Ruoyong Yang and James O. Berger. A catalog of noninformative priors. Draft, Parexel International, Duke University, 1998.