Bayesian Classification and Regression with High Dimensional Features

Longhai Li

longhai@utstat.utoronto.ca

Department of Statistics

University of Toronto

Supervisor: Radford M. Neal

Ph.D. Thesis Defense, 27 August 2007

Outline

- High Dimensional Measurements, such as Gene Expression Data

  Commonly select a small subset of features by looking at how “useful” they are in predicting y. However, this procedure will make y appear more predictable than it actually is. We propose a Bayesian method to avoid this bias.

- Considering High-order Interactions of Discrete Features

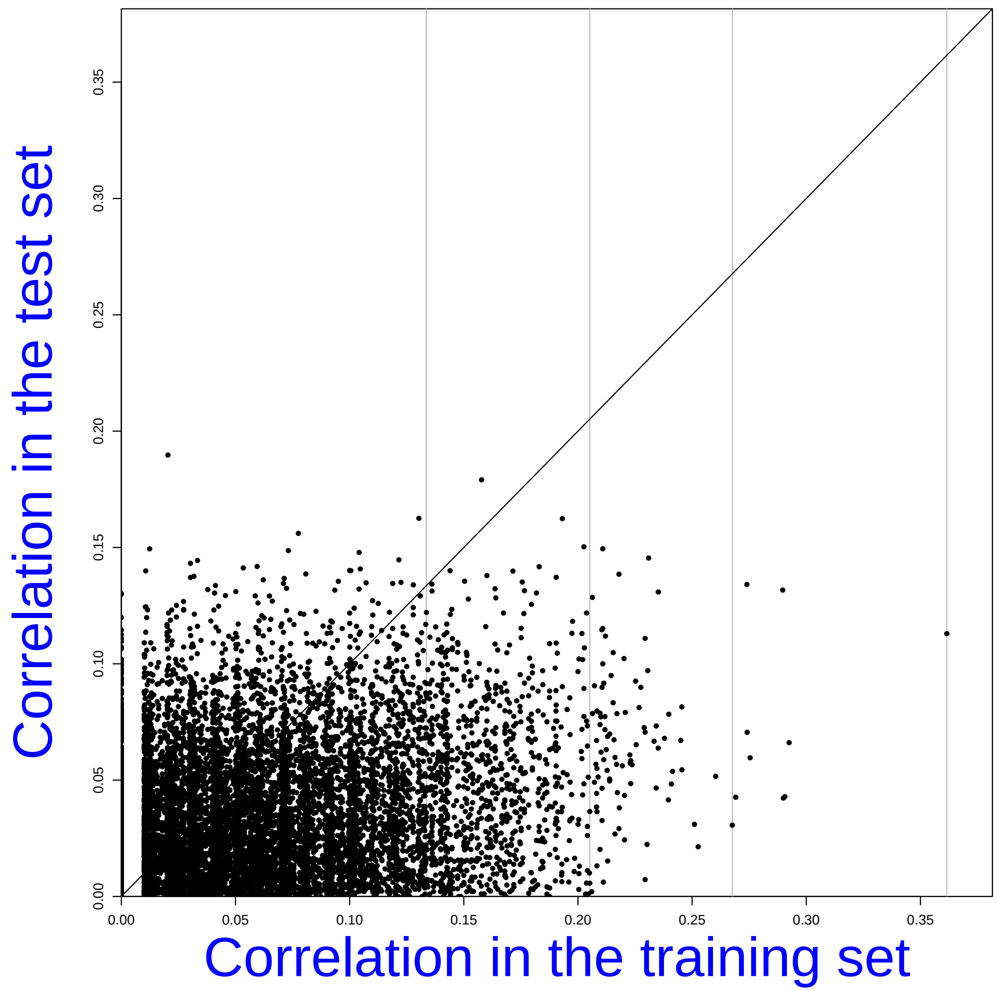
  The number of interactions increases exponentially with the order considered. We propose a Bayesian method to compress the parameters.

Part 1

Avoiding Bias from Feature Selection

Bias from Feature Selection: Stronger Relationship

Selecting a subset of features by looking at the correlations with y, will make the relationship between y and x stronger than it actually is:



Bias from Feature Selection: Effect on Predictions

- Predictive probabilities are lack of calibration:

$$P(\boldsymbol{Y} = 1 \mid \hat{Y}(\boldsymbol{X}) \in (c_1, c_2)) \quad \neq \quad E(\hat{Y}(\boldsymbol{X}) \mid \hat{Y}(\boldsymbol{X}) \in (c_1, c_2))$$

- Predictive probabilities are overconfident:

  The predictive probabilities of $y^{(i)} = 1$ are close to 1, say 0.9, for a set of test cases, but actually the frequency of $y^{(i)} = 1$, is smaller, say 0.7

- Error rates are underestimated:

  The **expected error rate** is smaller than the **actual error rate**

Our Method for Avoiding Bias from Feature Selection

- Idea: Our predictions should condition not only on the retained features $x_{1:k}^{\text{train}}$, but also on the fact that the other $p-k$ features have sample correlations with the response less than γ in absolute value:

$$y^{\text{train}}, \quad x_{1:k}^{\text{train}}, \text{ and } |\text{COR}(y^{\text{train}}, x_t^{\text{train}})| \leq \gamma \qquad \text{for} \quad t = k+1, \ldots, p$$

- Models: Given the response y, a model parameter α, and perhaps some latent values z^{train}, the features $x_1, \ldots, x_p$, are modeled to be independent and has identical distribution:

$$P(x_1, \cdots, x_p \mid y, \alpha, z^{\text{train}}) = \prod_{t=1}^{p} \left[P(x_t \mid y, \alpha, z^{\text{train}}) \right]$$

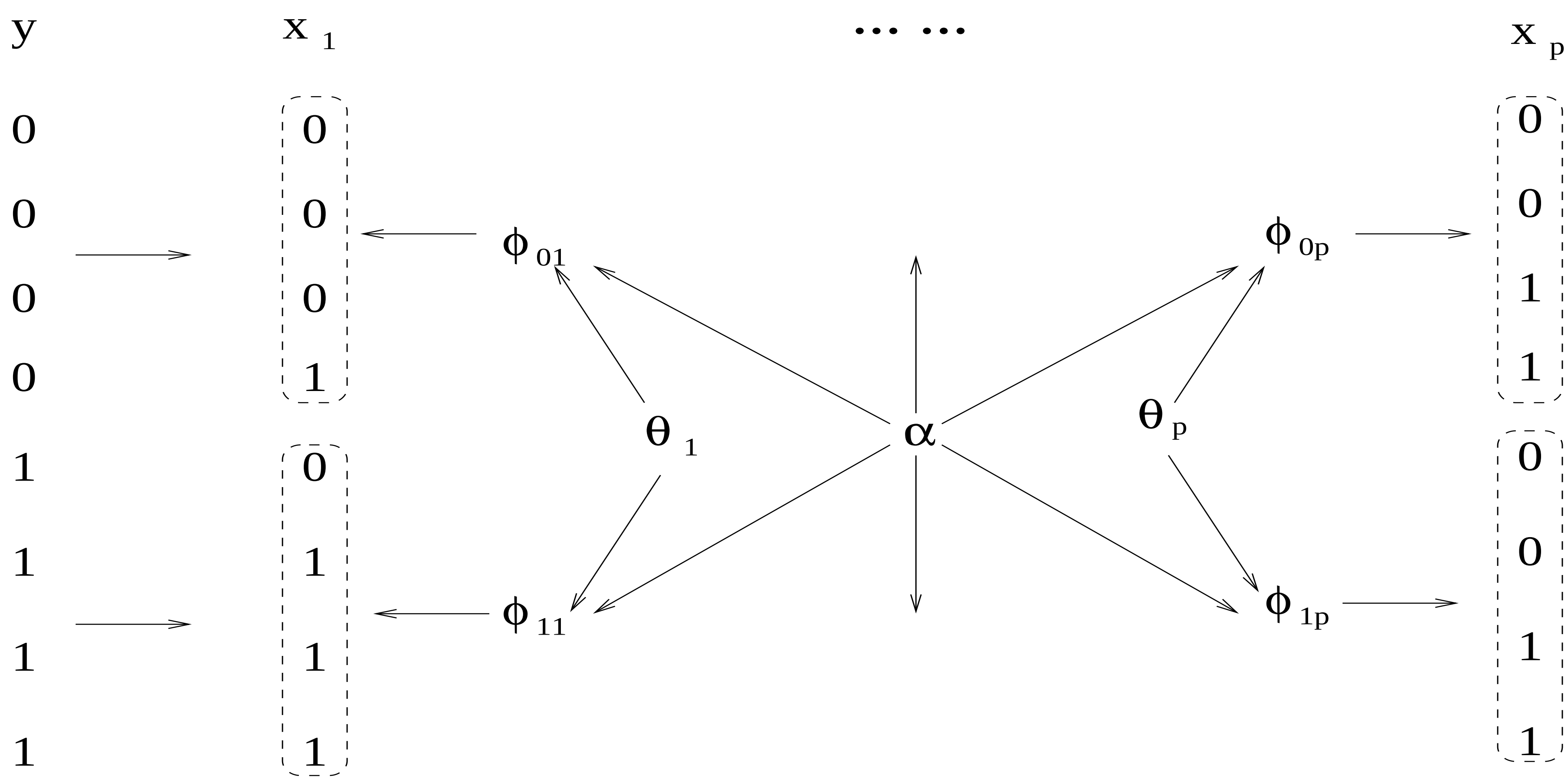
- Adjustment factor: The likelihood function of α and latent value z^{train} based only on $y^{\text{train}}, x_{1:k}^{\text{train}}$ is multiplied by:

$$P(\ |\text{COR}(y^{\text{train}}, x_t^{\text{train}})| \leq \gamma \text{ for } t = k+1, \ldots, p \mid \alpha, z^{\text{train}}, y^{\text{train}})$$
$$= \quad \left[P(\ |\text{COR}(y^{\text{train}}, x_t^{\text{train}})| \leq \gamma \mid \alpha, z^{\text{train}}, y^{\text{train}}) \right]^{p-k}$$

Part 1.1

Application to Naive Bayes Models

A Bayesian Naive Bayes Model for Binary Data



$$x_j^{(i)} \mid y^{(i)}, \phi \quad \sim \quad \text{Bernoulli}\,(\phi_{y^{(i)},j}), \quad \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, p$$

$$\phi_{0,j},\, \phi_{1,j} \mid \alpha,\, \theta_j \quad \overset{\text{IID}}{\sim} \quad \text{Beta}\,(\alpha\theta_j,\, \alpha(1-\theta_j)), \quad \text{for } j = 1, \ldots, p$$

Sample Correlation of Binary Data

COR$(x_t^{\text{train}}, y^{\text{train}})$ can be written as:

$$\text{COR}(x_t^{\text{train}},\, y^{\text{train}}) \;=\; \frac{(0-\overline{y})\, I_0 \;+\; (1-\overline{y})\, I_1}{\sqrt{n\overline{y}(1-\overline{y})}\,\sqrt{I_0 + I_1 - (I_0 + I_1)^2/n}}$$

where I_0,I_1 are:

y^{train}	:	0	0	0	0	1	1	1	1
x_t^{train}	:	0	1	1	1	0	0	0	1

$I_0 = 3$

$I_1 = 1$

Computation of the Adjustment Factor

I_1 \ I_0	0	1	2	3	4	5	6	7	8
14	+1.00	+0.90	+0.81	+0.72	+0.62	+0.53	+0.42	+0.29	0.00
13	+0.91	+0.80	+0.70	+0.60	+0.49	+0.38	+0.25	+0.09	−0.16
$H_+ \longrightarrow$ 12	+0.83	+0.72	+0.61	+0.50	+0.39	+0.27	+0.13	−0.03	−0.24
11	+0.76	+0.64	+0.52	+0.41	+0.30	+0.17	+0.04	−0.11	−0.30
10	+0.69	+0.57	+0.45	+0.33	+0.21	+0.09	−0.04	−0.18	−0.36
9	+0.63	+0.50	+0.38	+0.26	+0.14	+0.02	−0.11	−0.25	−0.41
8	+0.57	+0.44	+0.31	+0.19	+0.07	−0.05	−0.18	−0.31	−0.46
7	+0.52	+0.38	+0.24	+0.12	0.00	−0.12	−0.24	−0.37	−0.52
6	+0.46	+0.31	+0.18	+0.05	−0.07	−0.19	−0.31	−0.44	−0.57
5	+0.41	+0.25	+0.11	−0.02	−0.14	−0.26	−0.38	−0.50	−0.63
4	+0.36	+0.18	+0.04	−0.09	−0.21	−0.33	−0.45	−0.57	−0.69
3	+0.30	+0.11	−0.04	−0.17	−0.30	−0.41	−0.52	−0.64	−0.76
2	+0.24	+0.03	−0.13	−0.27	−0.39	−0.50	−0.61	−0.72	−0.83
1	+0.16	−0.09	−0.25	−0.38	−0.49	−0.60	−0.70	−0.80	−0.91
0	0.00	−0.29	−0.42	−0.53	−0.62	−0.72	−0.81	−0.90	−1.00

$$P(\,|\text{COR}(x_t^{\text{train}}, y^{\text{train}})| \le \gamma \mid \alpha,\, y^{\text{train}}) \;=\; 1 - 2 \sum_{(I_0,I_1)\in H_+} P(I_0,\, I_1 \mid \alpha,\, y^{\text{train}})$$

A Simulation Experiment on the Naive Bayes Model

- Generating data

  $\alpha = 300$, $p = 10000$, 200 training cases, 2000 test cases

- Selecting features

  4 subsets with only $1, 10, 100$ and 1000 features with largest correlations (in absolute value) were selected

- Priors

$$\alpha \sim \text{Inverse-Gamma}(0.5, 5)$$

- Computations

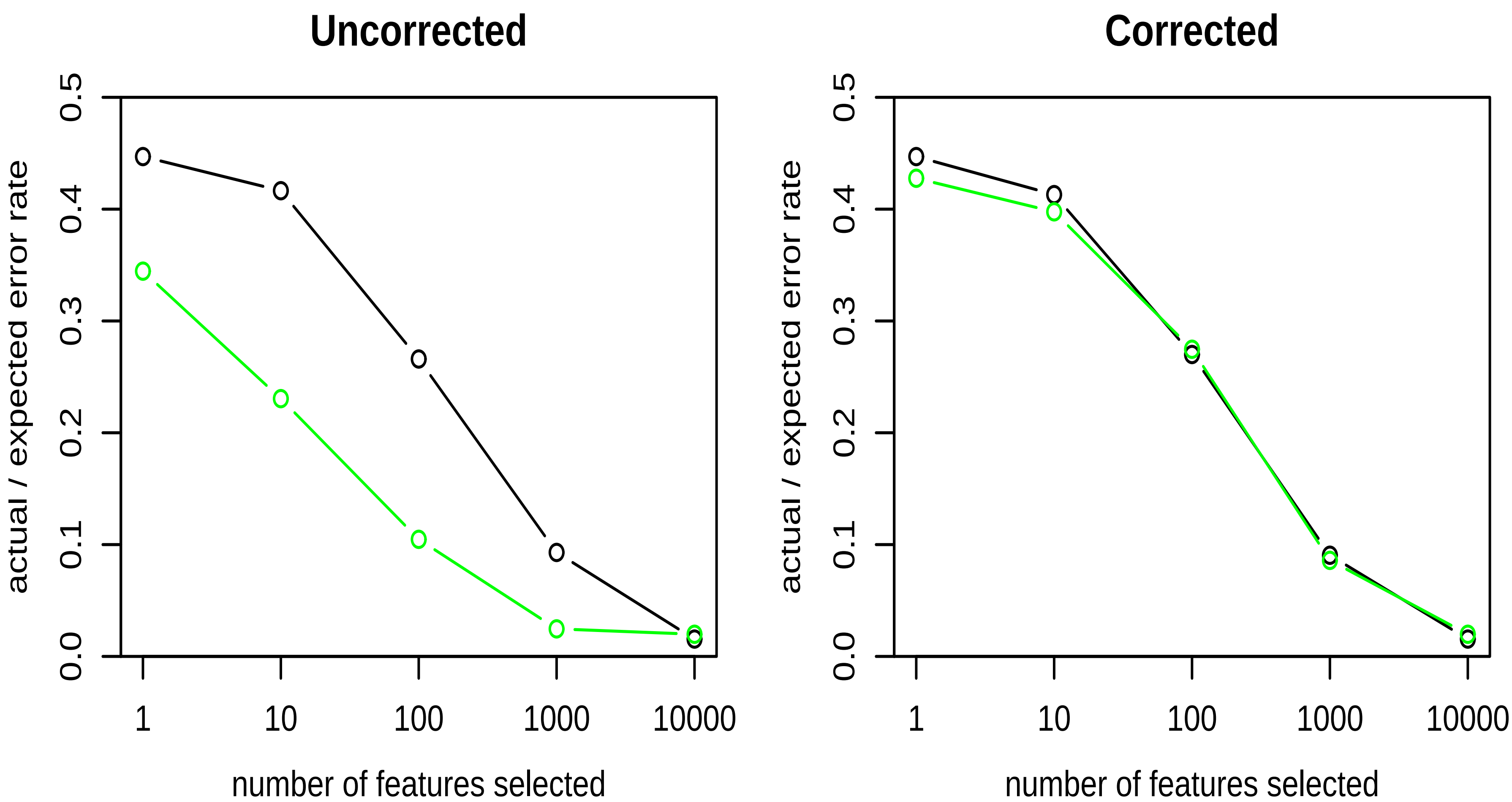
  We applied Simpson Rule to the integral over θ_j; apply midpoint Rule to the integral over α

  Computation times for uncorrected methods and corrected methods are almost identical.

Calibration of Predictions

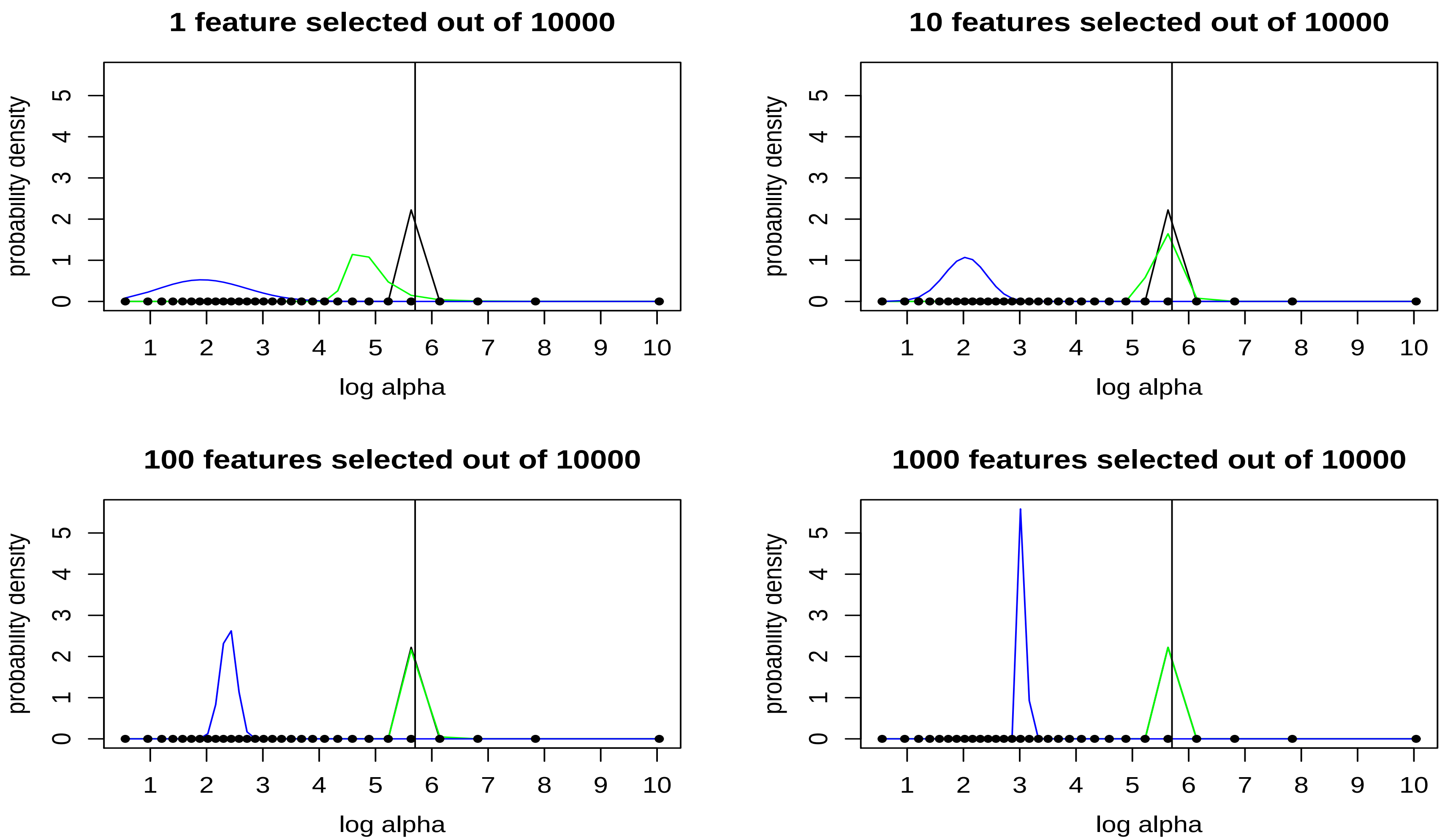
	100 features selected out of 10000					
	Corrected			Uncorrected		
Category	#	Pred	Actual	#	Pred	Actual
0.0 – 0.1	155	0.067	0.077	717	0.017	0.199
0.1 – 0.2	247	0.151	0.162	133	0.150	0.391
0.2 – 0.3	220	0.247	0.286	70	0.251	0.429
0.3 – 0.4	225	0.352	0.356	68	0.351	0.515
0.4 – 0.5	237	0.450	0.494	58	0.451	0.500
0.5 – 0.6	227	0.545	0.586	78	0.552	0.603
0.6 – 0.7	202	0.650	0.728	77	0.654	0.532
0.7 – 0.8	214	0.749	0.785	80	0.746	0.662
0.8 – 0.9	182	0.847	0.857	98	0.852	0.633
0.9 – 1.0	91	0.935	0.923	621	0.979	0.818

Actual and Expected Error Rate



Green = Expected Error Rate   Black = Actual Error Rate

Approximate Posterior Distribution of $\log(\alpha)$

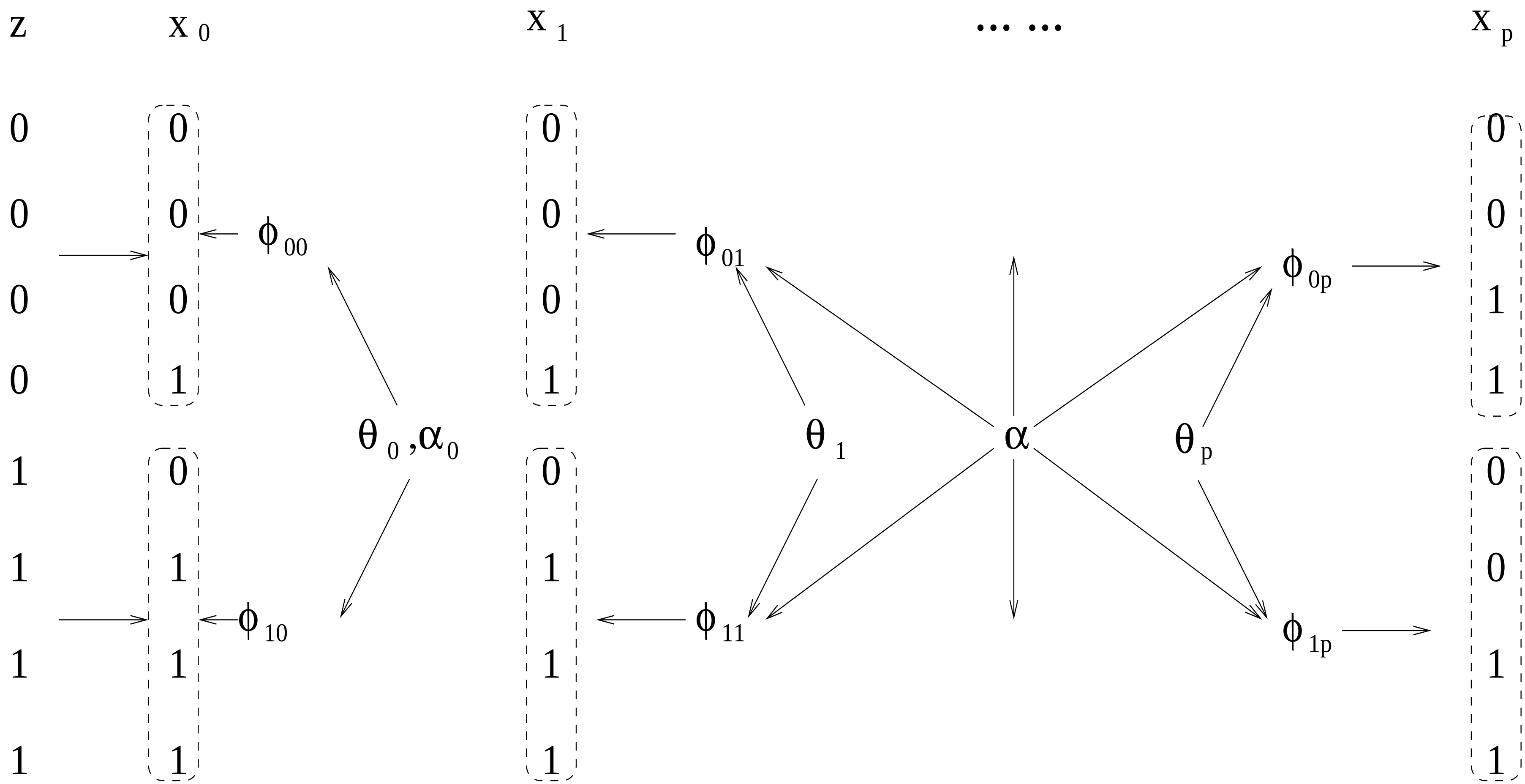


Blue=Uncorrected, Green=Corrected, Black=Complete Data, Vertical=True Value

Part 1.2

Application to Mixture Models

A Bayesian Mixture Model for Binary Data



$$x_j^{(i)} \mid z^{(i)},\, \phi \quad \sim \quad \text{Bernoulli}\,(\phi_{z^{(i)},j}), \quad \text{for } i = 1, \ldots, n \text{ and } j = 0, 1, \ldots, p$$

$$\phi_{0,j},\, \phi_{1,j} \mid \alpha,\, \theta_j \quad \overset{\text{IID}}{\sim} \quad \text{Beta}\,(\alpha\theta_j,\, \alpha(1-\theta_j)), \quad \text{for } j = 1, \ldots, p$$

Computation of the Adjustment Factor

Computation of the adjustment factor for this mixture model is similar to the preceding naive Bayes model. But it is more difficult because:

- It depends on the **unknown** latent values z^{train}. We have to use MCMC to sample for z^{train}, and therefore have to recompute the adjustment factor whenever we change z^{train}.
- I_0 and I_1 are not independent given $z^{\text{train}}, x_0^{\text{train}}$,$\theta_t$, and α. We need to split I_0 into $I_0^{[z]}$ for $z = 0, 1$, as well as for I_1.

$$P(I_0, I_1 \mid x_0^{\text{train}}, \boldsymbol{z}^{\text{train}}, \theta_t, \alpha) = \sum_{\substack{I_0^{[0]} + I_0^{[1]} = I_0 \\ I_1^{[0]} + I_1^{[1]} = I_1}} \prod_{z=0}^{1} P(I_0^{[z]}, I_1^{[z]} \mid x_0^{\text{train}}, \boldsymbol{z}^{\text{train}}, \theta_t, \alpha)$$

Part 2

Compressing Parameters in Bayesian Models with High-order Interactions

Predictor Variables Derived from Interactions

Discrete Measurements

i	x_1	x_2
1	1	2
2	2	1
3	1	1

$\Longrightarrow$

Indicators on Interaction Patterns

i	$I_{[00]}$	$I_{[10]}$	$I_{[20]}$	$I_{[01]}$	$I_{[02]}$	$I_{[11]}$	$I_{[21]}$	$I_{[12]}$	$I_{[22]}$
1	1	1	0	0	1	0	0	1	0
2	1	0	1	1	0	0	1	0	0
3	1	1	0	1	0	1	0	0	0

Facts:

- The number of predictor variables increases exponentially with the order considered.
- Many predictor variables derived this way have the same value for all training cases.

Compressing Parameters

When groups of predictor variables have the same value for all training cases, the likelihood function of a linear regression model depends only on the sums over groups:

$$\begin{aligned} L^{\beta}(\beta_{11}, \ldots, \beta_{1,n_1}, \ \ldots \ , \beta_{G1}, \ldots, \beta_{G,n_G}) &= L\left(\sum_{k=1}^{n_1} \beta_{1k}, \ \ldots, \ \sum_{k=1}^{n_G} \beta_{Gk}\right) \\ &= L(s_1, \ \ldots \ , s_G) \end{aligned}$$

We use priors as $\beta_{gk} \sim N(0, \sigma^2_{gk})$ or $\beta_{gk} \sim$ Cauchy$(0, \sigma_{gk})$, because the priors of the s_g's can be found easily:

$$s_g \sim N\left(0, \ \sum_{k=1}^{n_g} \sigma^2_{gk}\right) \quad \text{or} \quad s_g \sim \text{Cauchy}\left(0, \ \sum_{k=1}^{n_g} \sigma_{gk}\right)$$

The posterior of the s_g's given the training data $\mathcal{D}$:

$$P(\boldsymbol{s} \mid \mathcal{D}) = \frac{1}{c(\mathcal{D})} L(s_1, \ \ldots \ , s_G) \ P^{(s)}_1(s_1) \ \cdots \ P^{(s)}_g(s_G)$$

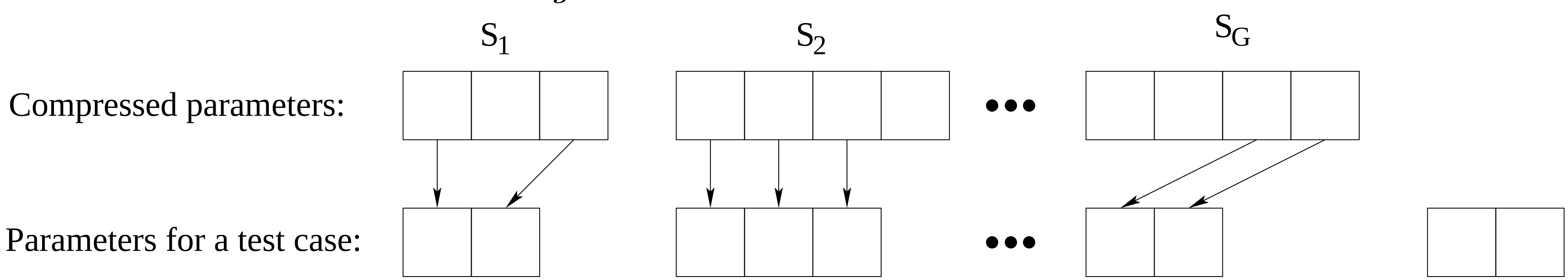
Splitting Compressed Parameters

After obtaining the samples of s_g's using MCMC, we can recover the original parameters, using the splitting distribution:

$$P(\beta_{g1}, \ldots, \beta_{g,n_g-1} \mid s_g) = \prod_{k=1}^{n_g-1} P_{gk}(\beta_{gk})\ P_{g,n_g}\left(s_g - \sum_{k=1}^{n_g-1} \beta_{gk}\right) / P_g^s(s_g)$$

The splitting distribution is unrelated to $\mathcal{D}$. We can directly sample from it.

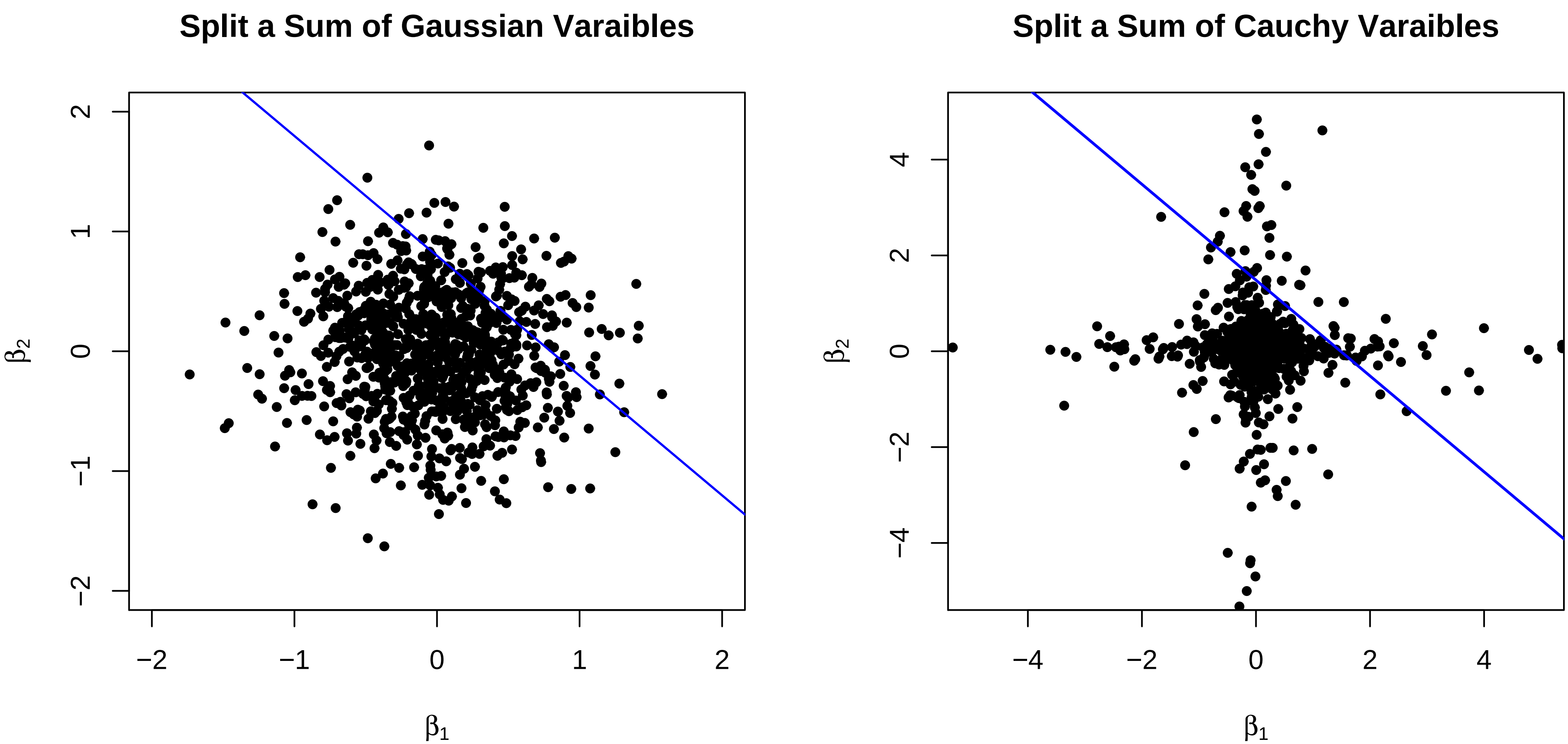
To save space, we can split s_g temporarily for each test case.



Need only to split s_g into two parts for a particular test case:

$$P(s_g^t \mid s_g) = P_g^{(1)}(s_g^t)\ P_g^{(2)}(s_g - s_g^t) / P_g^s(s_g)$$

Splitting s_g into Two Parts: Graphical Illustration



Splitting s_g into Two Parts: Formulae

- Split a sum of Gaussian variables:

$$s_g^t \mid s_g \sim N\left(s_g \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\,,\ \sigma_1^2\left(1 - \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)\right)$$

- Split a sum of Cauchy variables:

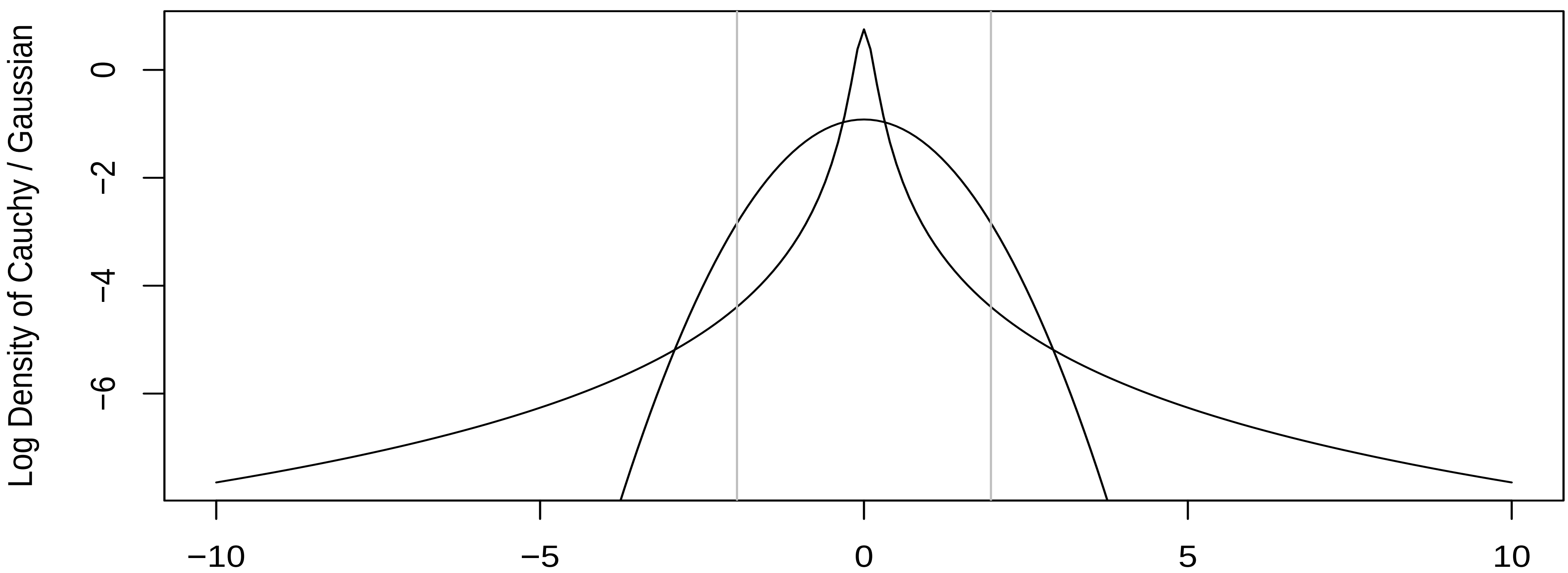
$$\begin{aligned} F(s_g^t\,;\,s_g,\sigma_1,\sigma_2) \quad = \quad & \frac{1}{C}\left[r\log\left(\frac{(s_g^t)^2+\sigma_1^2}{(s_g^t-s_g)^2+\sigma_2^2}\right)+\right. \\ & p_0\left(\arctan\left(\frac{s_g^t}{\sigma_1}\right)+\frac{\pi}{2}\right)+ \\ & \left. p_s\left(\arctan\left(\frac{s_g^t-s_g}{\sigma_2}\right)+\frac{\pi}{2}\right)\right] \end{aligned}$$

Being able to compute the CDF, we can use inversion method to sample from the above distribution, with the inverse CDF found numerically.

Cauchy priors VS Gaussian priors

A Cauchy distribution centered at 0 describes more accurately the scenario that most parameters are close to 0 but a few may be very large.

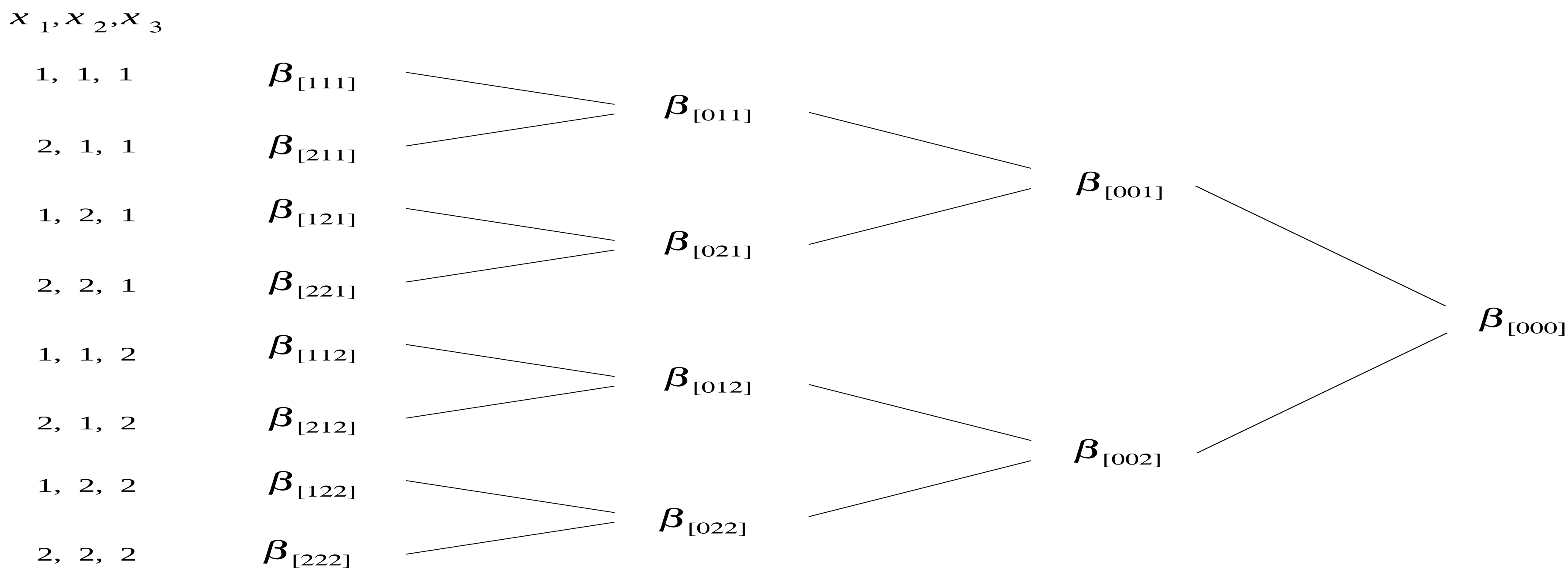
For example, if we expect 95% parameters are in interval $(-1.96, 1.96)$, we should use N$(0,1)$ or Cauchy$(0,0.15)$. Their log density function are plotted as:



Part 2.1

Application to Logistic Sequence Prediction Models

A Picture of Logistic Sequence Prediction Models



Remarks:

- By including low-order interactions, the predictive distributions given similar preceding sequences are similar.
- We are not forced to use a short sequence. Some useful high-order interactions can be considered.

Experiments on English Text

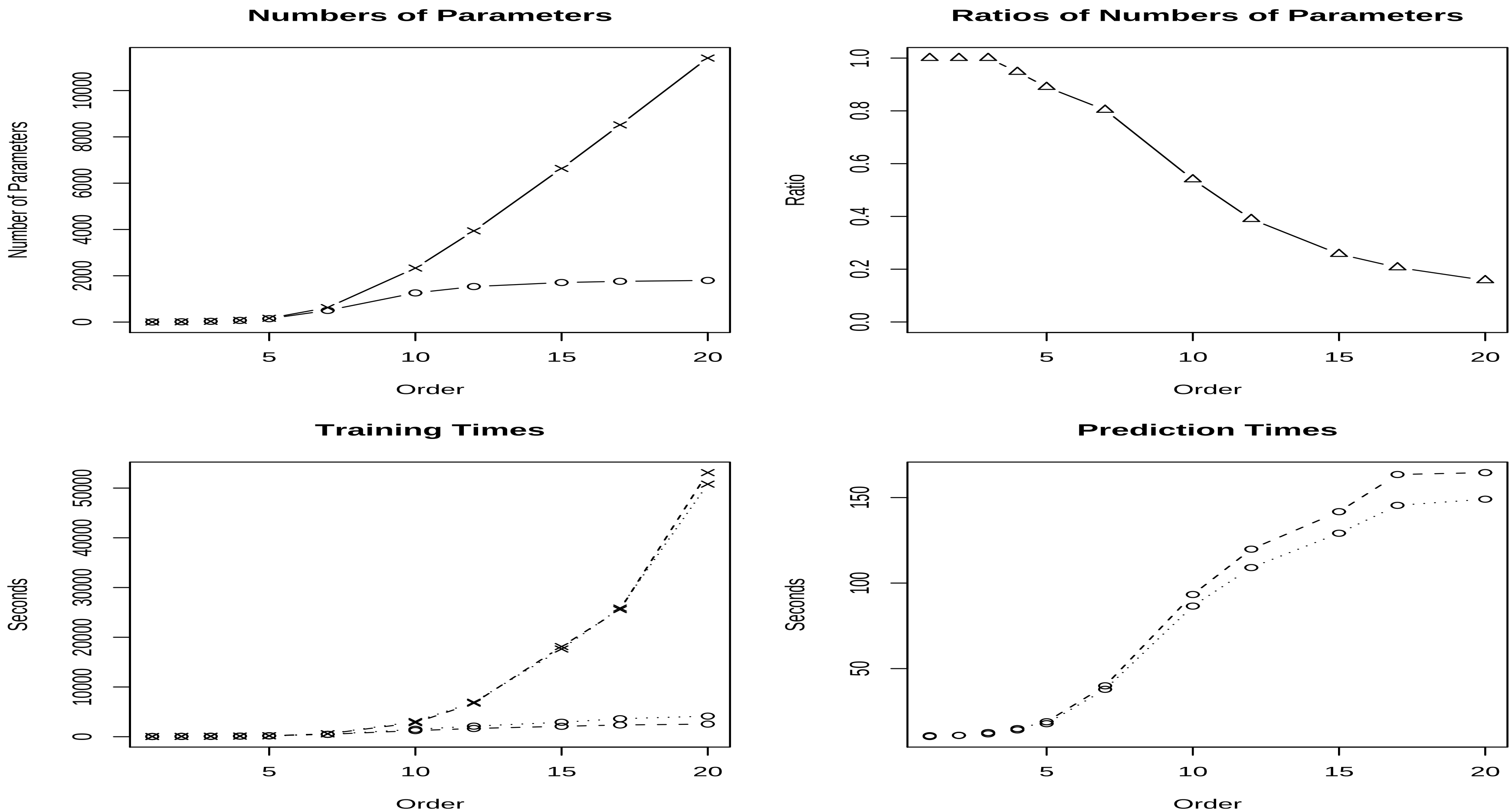
An online article, which introduces the Department of Statistics at University of Toronto, is encoded:

- 1 = vowel letters
- 2 = consonant letters
- 3 = all other characters, such as space, number, special symbols (remove consecutive instances)

There were a total of 3930 characters, giving 3910 overlapped sequences of length 21. Tested our method by predicting the 21st character based on varying numbers of preceding characters.

The first 1000 sequences were used as training cases. The remaining 2910 were used as test cases.

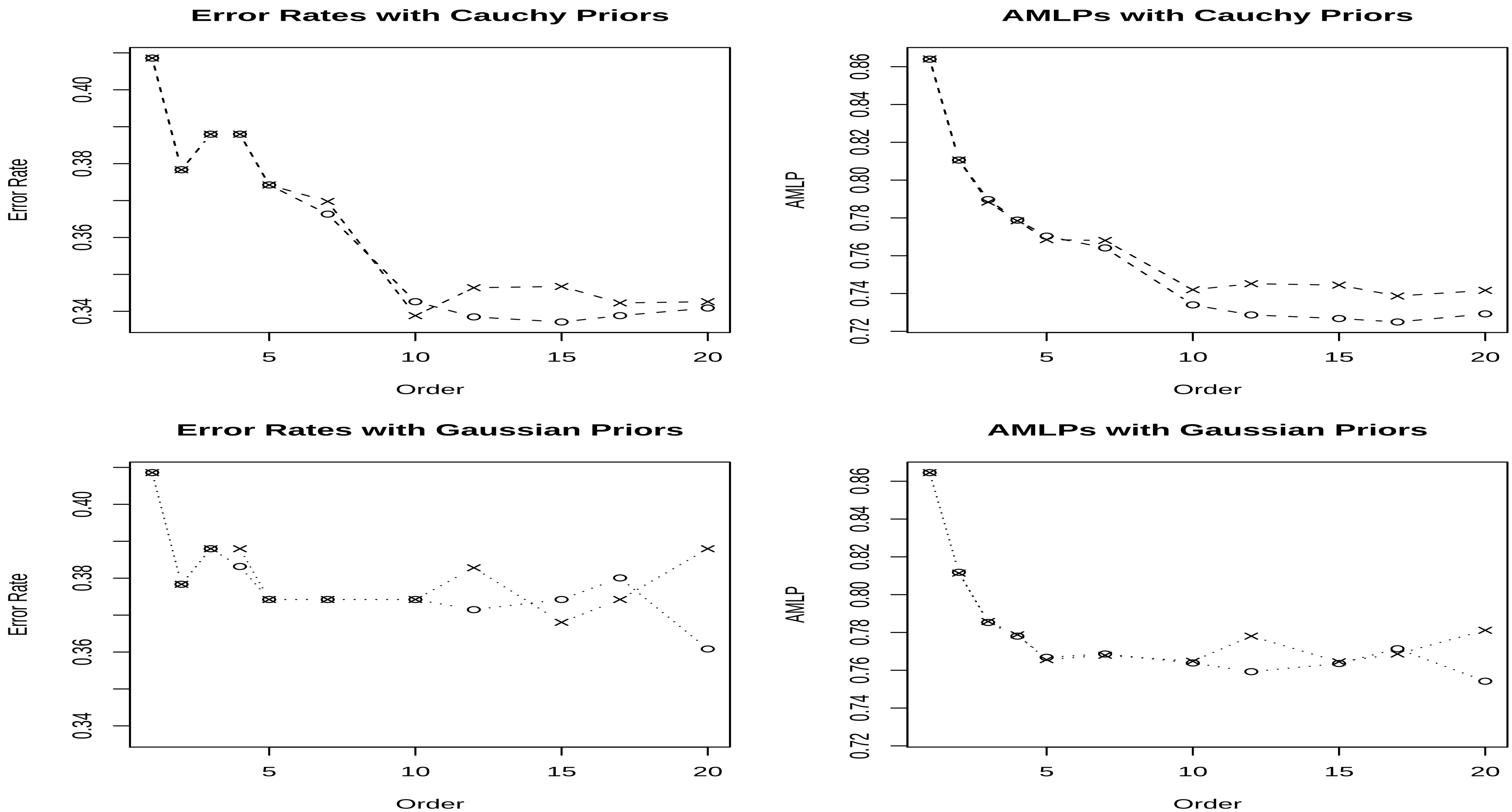
Parameter Reduction



○ = Parameters Compressed, × = Original Parameters

— — — = Using Cauchy Priors, · · · = Using Gaussian Priors

Error Rate and Average Minus Log Probability (AMLP)



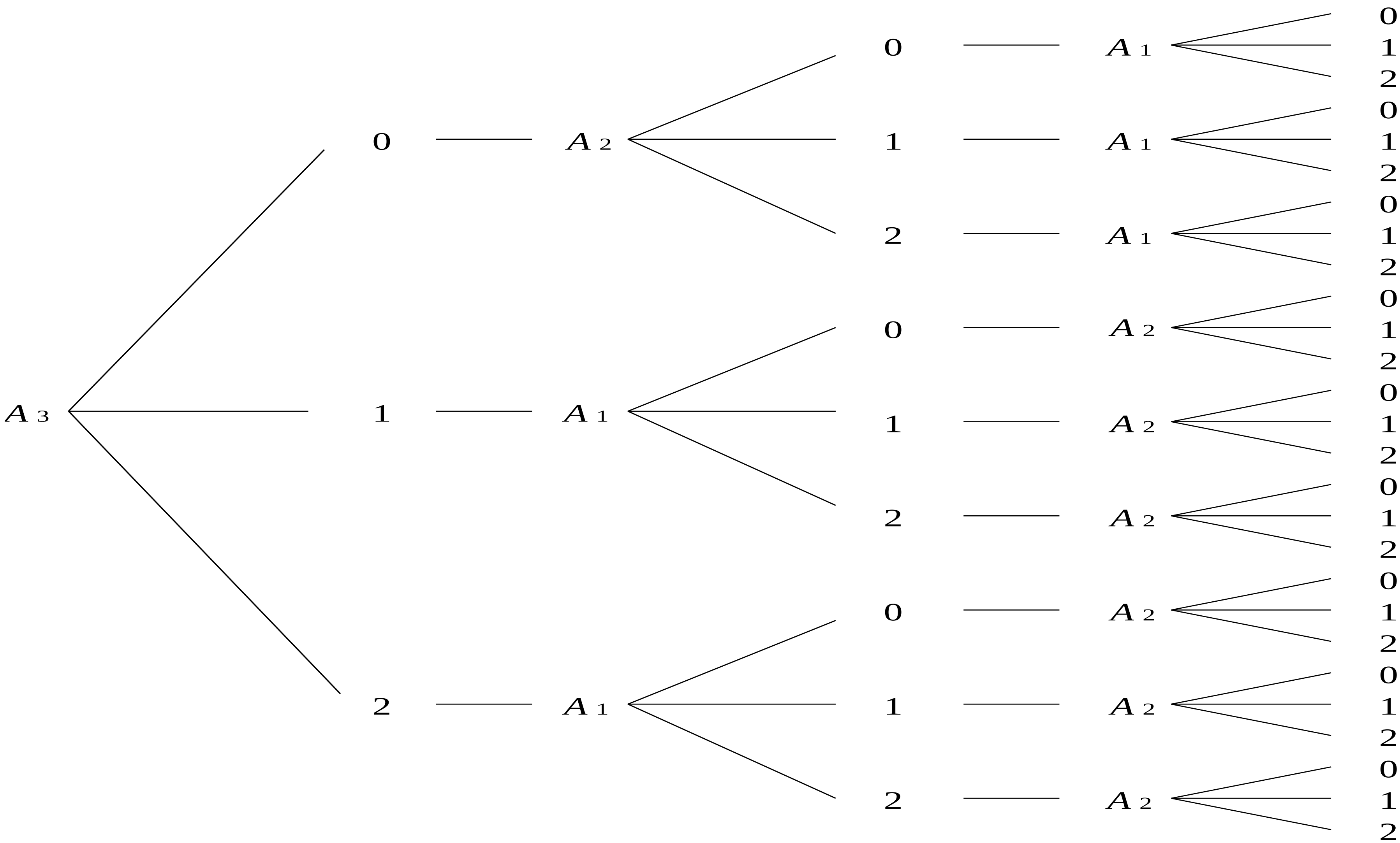
○ = Parameters Compressed, × = Original Parameters

— — — = Using Cauchy Priors, · · · = Using Gaussian Priors

Part 2.2

Application to Logistic Classification Models

A Picture of Logistic Classification Model



Conclusions

- We propose a Bayesian method to make well-calibrated predictions using a small subset of features selected from a large number.
- We propose a method to compress the parameters in Bayesian models with high-order interactions. The number of parameters is reduced greatly.
- We demonstrate empirically that Cauchy distributions could be better than Gaussian distributions as the priors for the regression coefficients of high-order models for some problems.