Avoiding Bias from Feature Selection

Longhai Li

Joint work with Jianguo Zhang and Radford M. Neal

longhai@utstat.utoronto.ca

Department of Statistics

University of Toronto

Toronto Ontario CANADA

Outline of the talk

- Introduction to Classification Problems
- Challenge from High Dimensional Features
- Bias from Feature Selection
- A Method for Avoiding Bias from Feature Selection
- Application to Naive Bayes Classification Model
  - Definition of Naive Bayes Classification Model
  - Predictions for Test Cases
  - Computation of Adjustment Factor
  - Demonstration with Simulated and Real Datasets
- Conclusion and Discussion

Introduction to Classification Problems

- Goal: Given a feature vector x, we want to predict the associated response y, i.e. find a predictive function C from x to y:

$$y = C(x)$$

- Probabilistic Classification: A better way is to report the predictive distribution of y given x:

$$P(y \mid x)$$

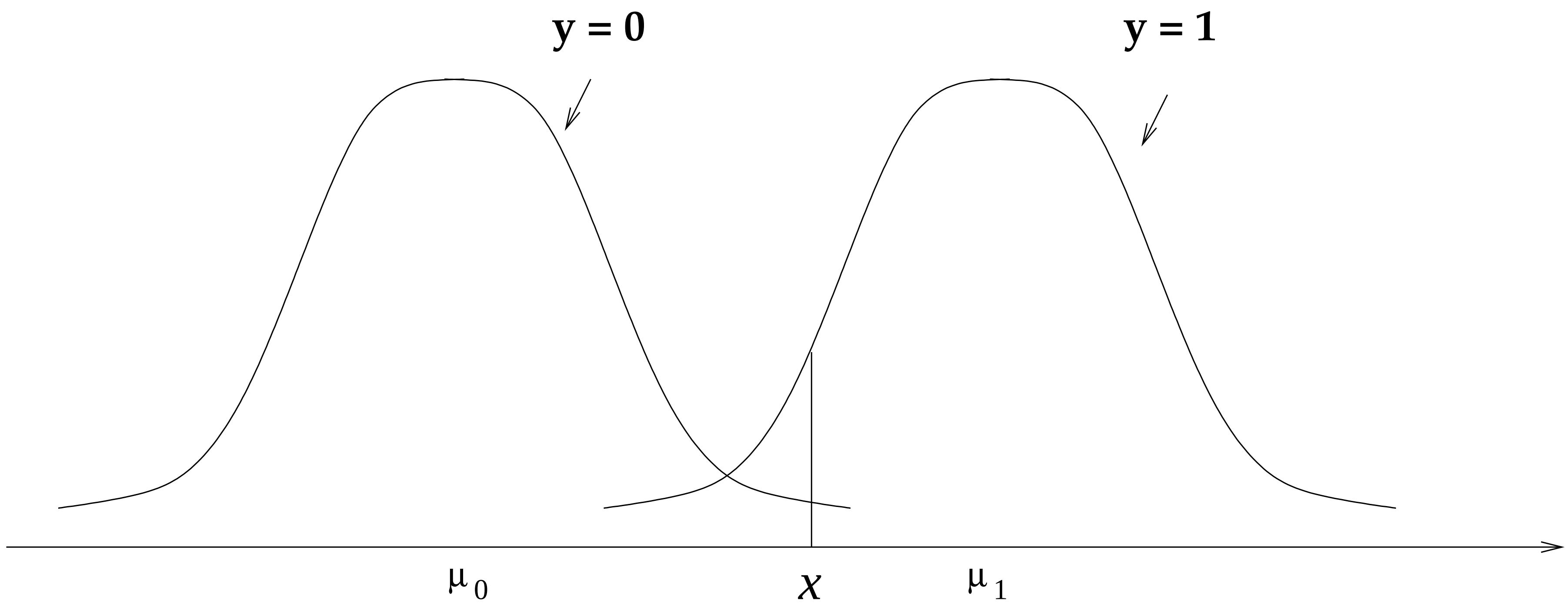
- Statistical Method: Estimate $P(y \mid x)$ by learning from the available data $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, collectively $\{x^{\text{train}}, y^{\text{train}}\}$, called training data
- Examples:
  - Given the image of handwritten digit, recognize the digit on it
  - Given the content of a message, determine whether it is a spam
  - Given gene expression data of a patient, classify the type of tumor

Model-based Classification

Model $P(y)$ and $P(x \mid y)$, or model $P(y, x)$ directly, then $P(y \mid x)$ is found by Bayes rule:

$$P(y \mid x) = P(y, x)/P(x) = P(y)\, P(x \mid y)/P(x)$$

Illustration with a simple example:



Bayesian Statistical Classification

As the joint distribution of $P(x, y)$ is defined with some unknown parameter θ,it is rewritten as $P(x, y \mid \theta)$.

Bayesian Statistical Classification

As the joint distribution of $P(x, y)$ is defined with some unknown parameter θ,it is rewritten as $P(x, y \mid \theta)$.

- Our preference as to which θ may be good for our data is expressed with a probability distribution:

  $$\theta \sim P(\theta)$$

  This is called prior distribution of θ.

Bayesian Statistical Classification

As the joint distribution of $P(x, y)$ is defined with some unknown parameter θ,it is rewritten as $P(x, y \mid \theta)$.

- Our preference as to which θ may be good for our data is expressed with a probability distribution:

  $$\theta \ \sim \ P(\theta)$$

  This is called prior distribution of θ.

- Updating our knowledge for θ in light of training data $x^{\text{train}}, y^{\text{train}}$:

  $$P(\theta \mid x^{\text{train}}, y^{\text{train}}) = \frac{P(x^{\text{train}}, y^{\text{train}} \mid \theta) \ P(\theta)}{P(x^{\text{train}}, y^{\text{train}})}$$

  This is called posterior distribution of θ given $x^{\text{train}}, y^{\text{train}}$.

Bayesian Statistical Classification

As the joint distribution of $P(x, y)$ is defined with some unknown parameter θ,it is rewritten as $P(x, y \mid \theta)$.

- Our preference as to which θ may be good for our data is expressed with a probability distribution:

$$\theta \ \sim \ P(\theta)$$

  This is called prior distribution of θ.

- Updating our knowledge for θ in light of training data $x^{\text{train}}, y^{\text{train}}$:

$$P(\theta \mid x^{\text{train}}, y^{\text{train}}) = \frac{P(x^{\text{train}}, y^{\text{train}} \mid \theta) \ P(\theta)}{P(x^{\text{train}}, y^{\text{train}})}$$

  This is called posterior distribution of θ given $x^{\text{train}}, y^{\text{train}}$.

- The joint distribution of x^*, y^* for a test case is found with updated information of θ:

$$P(x^*, y^* \mid x^{\text{train}}, y^{\text{train}}) = \int P(x^*, y^* \mid x^{\text{train}}, y^{\text{train}}, \theta) \ P(\theta \mid x^{\text{train}}, y^{\text{train}}) \ d\theta$$

Challenge from High Dimensional Features

- Examples of High Dimensional Features

Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes

Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.

Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features

Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features
  - Time: Computationally intensive

Challenge from High Dimensional Features

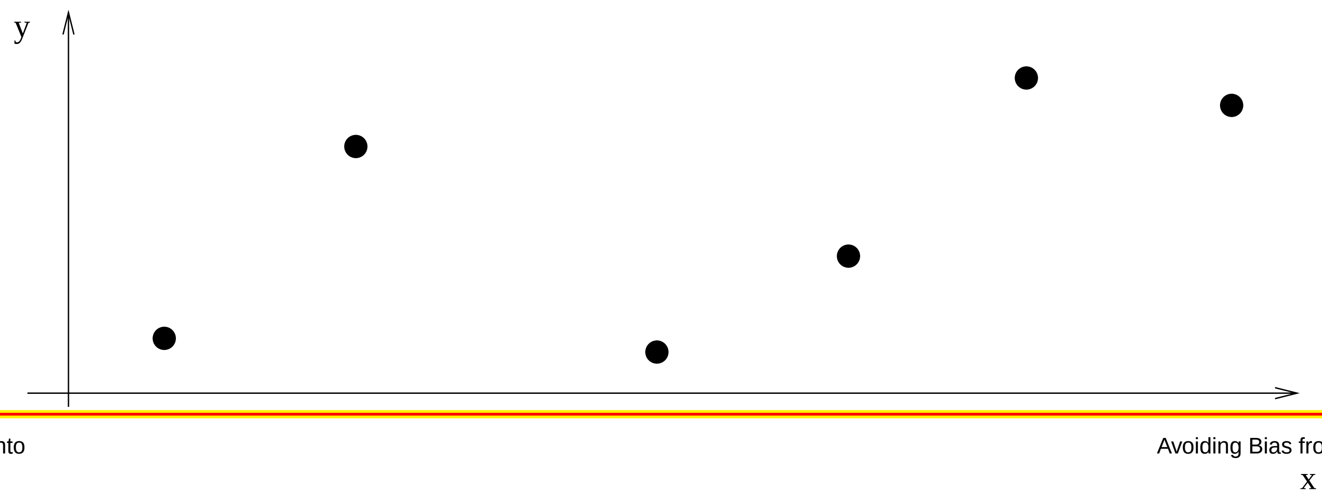
- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features
  - Time: Computationally intensive
  - Money: Costly in measuring the features for future cases

Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features
  - Time: Computationally intensive
  - Money: Costly in measuring the features for future cases
  - Overfitting: When the number of observations is smaller than the number of parameters, the likelihood function $P(x^{\text{train}}, y^{\text{train}} \mid \theta)$ will favor “too good” θ.

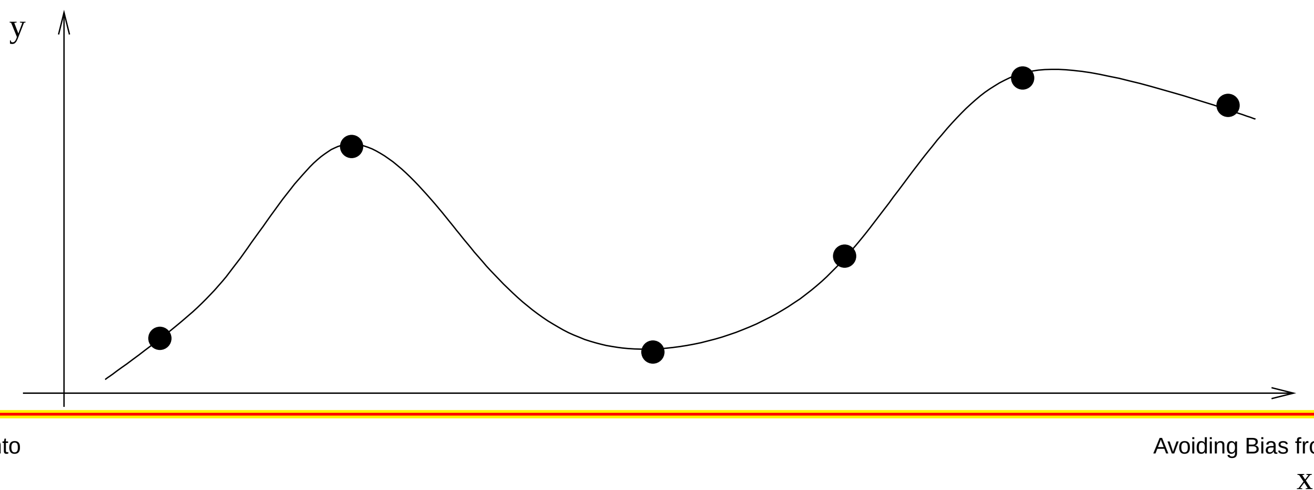
Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features
  - Time: Computationally intensive
  - Money: Costly in measuring the features for future cases
  - Overfitting: When the number of observations is smaller than the number of parameters, the likelihood function $P(x^{\text{train}}, y^{\text{train}} \mid \theta)$ will favor "too good" θ.



Challenge from High Dimensional Features

- Examples of High Dimensional Features
  - Gene Expression Data: Measure the expression levels of tens of thousands of genes
  - Document Classification: Count the frequency (times of occurance) of all words in a large predefined dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features
  - Time: Computationally intensive
  - Money: Costly in measuring the features for future cases
  - Overfitting: When the number of observations is smaller than the number of parameters, the likelihood function $P(x^{\text{train}}, y^{\text{train}} \mid \theta)$ will favor "too good" θ.



Bias from Feature Selection

For previous reasons, one may like to select a subset of features to use based on some measure of the dependency, such as absolute correlation, with y. However, this procedure will introduce an optimistic bias, i.e., the relationship between y and x appears stronger than it actually is, which can be phrased in following ways:

Bias from Feature Selection

For previous reasons, one may like to select a subset of features to use based on some measure of the dependency, such as absolute correlation, with y. However, this procedure will introduce an optimistic bias, i.e., the relationship between y and x appears stronger than it actually is, which can be phrased in following ways:

- Lack of calibration predictions: e.g.
  If y^* is binary, for a bunch of test cases, the predictive probabilities of $y^* = 1$ are 0.95, but actually the frequency of $y^* = 1$ among these test cases is only 0.70

Bias from Feature Selection

For previous reasons, one may like to select a subset of features to use based on some measure of the dependency, such as absolute correlation, with y. However, this procedure will introduce an optimistic bias, i.e., the relationship between y and x appears stronger than it actually is, which can be phrased in following ways:

- Lack of calibration predictions: e.g.
  If y^* is binary, for a bunch of test cases, the predictive probabilities of $y^* = 1$ are 0.95, but actually the frequency of $y^* = 1$ among these test cases is only 0.70

- Underestimation of prediction errors: e.g.
  If y^* is binary, and guess y^* by thresholding the predictive probability $\hat{p}^*$ at $1/2$, the *expected (estimated) error rate*, defined as

  $$\hat{p}^* \, I(\hat{p}^* < 1/2) + (1 - \hat{p}^*) \, I(\hat{p}^* \geq 1/2),$$

  is smaller than the actual error rate.

A Method for Avoiding Bias from Features 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 correlation with the response that is less than γ in absolute value:

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

A Method for Avoiding Bias from Features Selection

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

$$y^{\text{train}}, \quad x^{\text{train}}_{1:k}, \quad |\text{COR}(y^{\text{train}}, x^{\text{train}}_t)| \leq \gamma \text{ for } t = k+1, \ldots, p$$

- Models: The response and the predictors are modeled jointly. Given the response y and a model parameter α, the features $x_1, \ldots, x_p$, are modeled to be independent and has identical distribution:

$$P(x_1, \cdots, x_p \mid y, \alpha) = \prod_{t=1}^{p} \Big[P(x_t \mid y, \alpha) \Big]$$

A Method for Avoiding Bias from Features 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 correlation with the response that is less than γ in absolute value:

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

- Models: The response and the predictors are modeled jointly. Given the response y and a model parameter α, the features $x_1, \ldots, x_p$, are modeled to be independent and has identical distribution:

$$P(x_1, \cdots, x_p \mid y, \alpha) = \prod_{t=1}^{p} \Big[P(x_t \mid y, \alpha) \Big]$$

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

$$\begin{aligned} & P(\, |\text{COR}(y^{\text{train}}, x_t^{\text{train}})| \leq \gamma \text{ for } t = k+1, \ldots, p \mid \alpha, y^{\text{train}}) \\ = \quad & \Big[P(\, |\text{COR}(y^{\text{train}}, x_t^{\text{train}})| \leq \gamma \mid \alpha, y^{\text{train}}) \Big]^{p-k} \end{aligned}$$

Remark: we need to compute the adjustment factor only once regardless how many features are discarded.

Bayesian Naive Bayes Model for Binary Data

Data distribution:

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

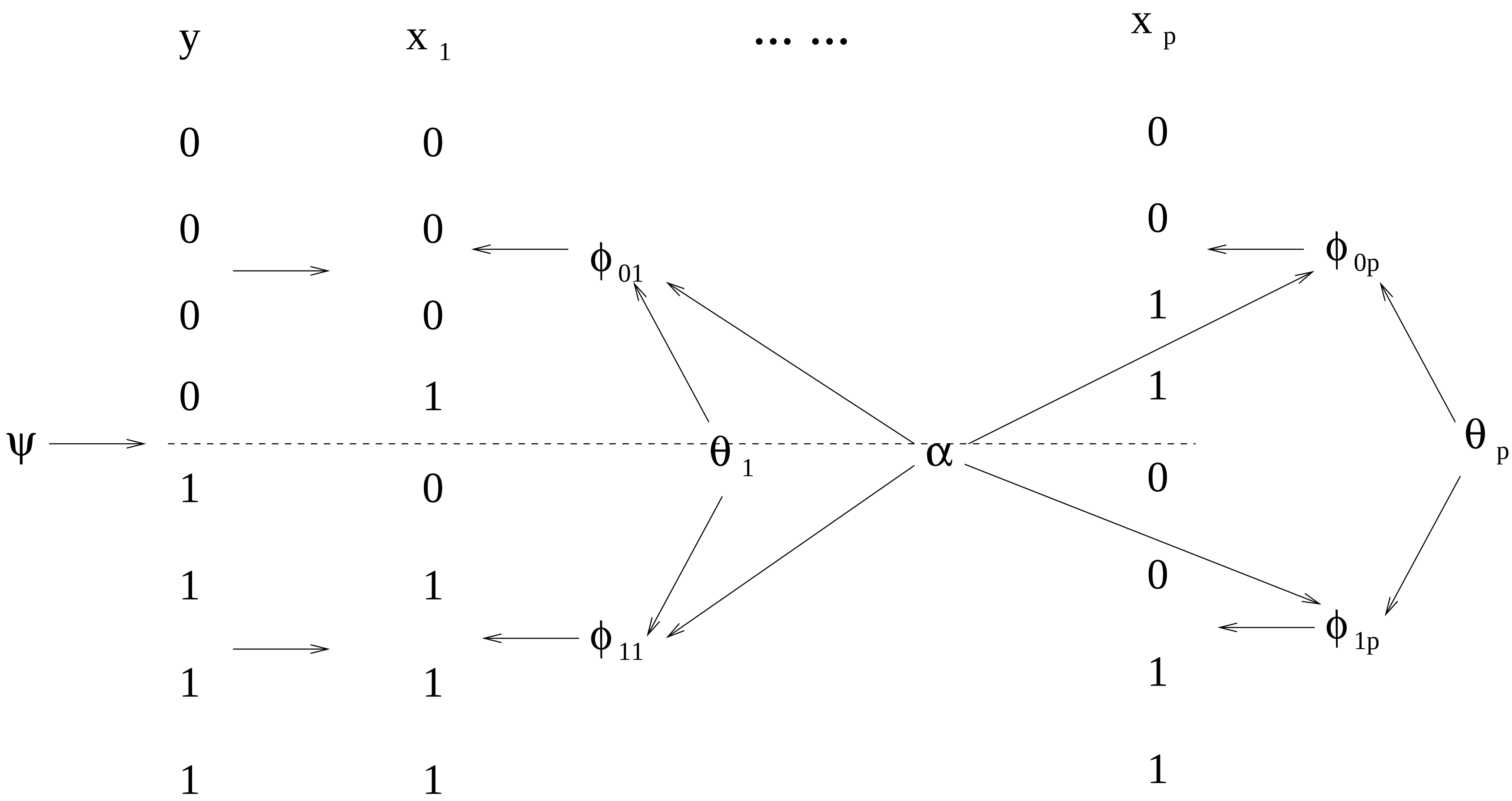
Prior distribution:

$$
\begin{aligned}
\psi &\sim \text{Beta}\,(f_1, f_0) \\
\phi_{0,j},\, \phi_{1,j} \mid \alpha,\, \theta_j &\overset{\text{IID}}{\sim} \text{Beta}\,(\alpha\theta_j,\, \alpha(1-\theta_j)), \quad \text{for } j = 1, \ldots, p \\
\alpha &\sim \text{Inverse-Gamma}(a, b) \\
\theta_1, \ldots, \theta_p &\overset{\text{IID}}{\sim} \text{Uniform}(0, 1)
\end{aligned}
$$

Note:

$$
\begin{aligned}
\text{E}(\phi_{yj}) &= \theta_j \\
\text{Var}(\phi_{yj}) &= \theta_j(1-\theta_j)/(\alpha+1)
\end{aligned}
$$

A Picture of Bayesian Naive Bayes Model



Obtaining $P(y^* \mid y^{\text{train}})$

For the model defined previously, $P(y^* \mid y^{\text{train}})$, which is needed to find the conditional $P(y^* \mid x^*, x^{\text{train}}, y^{\text{train}})$, is found to be:

$$P(y^* \mid y^{\text{train}}) \quad = \quad \text{Bernoulli}\,(y^*; \hat{\psi})$$

where $\hat{\psi} = (f_1 + N_1) \,/\, (f_0 + f_1 + n)$.

$$\begin{array}{lll} & f_1 \searrow & \\ y^{\text{train}}\ : & & \psi \longrightarrow \underbrace{0 \quad 0 \quad 0 \quad 0}_{n - N_1} \quad \underbrace{1 \quad 1 \quad 1 \quad 1}_{N_1} \\ & f_0 \nearrow & \end{array}$$

In more details, $P(y^{\text{train}})$ is computed as follows:

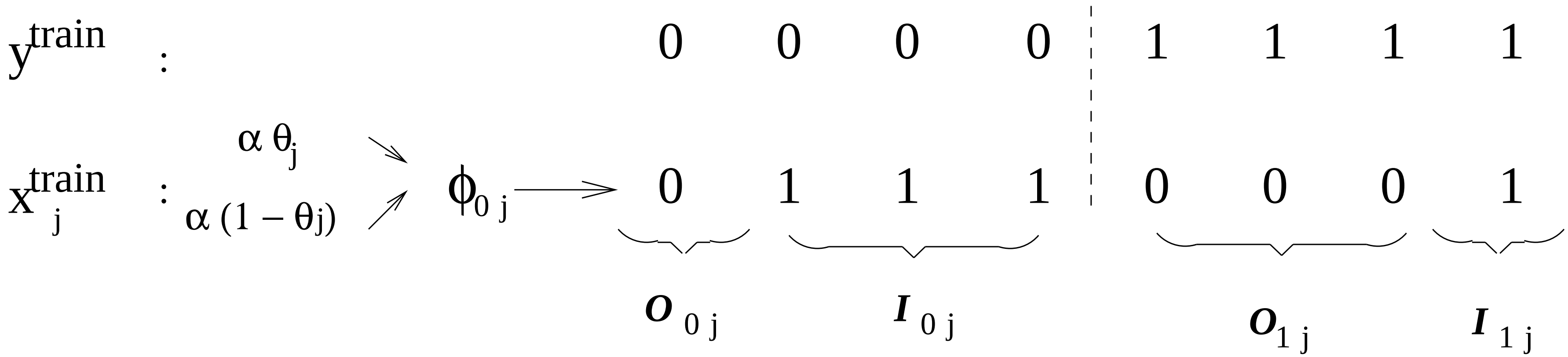
$$\begin{aligned} P(y^{\text{train}}) &= \int_0^1 \frac{\Gamma(f_0+f_1)}{\Gamma(f_0)\Gamma(f_1)} \psi^{f_1} (1 - \psi)^{f_0}\ \psi^{\sum\limits_{i=1}^{n} I(y^{(i)}=1)}\ (1-\psi)^{\sum\limits_{i=1}^{n} I(y^{(i)}=0)} d\psi \\ &= U\Big(f_0,\ f_1,\ \sum_{i=1}^{n}\ I\big(y^{(i)}=0\big),\ \sum_{i=1}^{n}\ I\big(y^{(i)}=1\big)\Big) \end{aligned}$$

Obtaining $P(x_j^* \mid \theta_j,\, \alpha,\, x_j^{\text{train}},\, y^{\text{train}},\, y^*)$

Similarly,

$$P(x_j^* \mid \theta_j,\, \alpha,\, x_j^{\text{train}},\, y^{\text{train}},\, y^*) = \text{Bernoulli}\,(x_j^*; \hat{\phi}_{y^*,j})$$

where $\hat{\phi}_{y^*,j} = (\alpha\theta_j + I_{y^*,j}) \,/\, (\alpha + N_{y^*})$ and $I_{y,j} \;=\; \sum_{i=1}^{n} I(y^{(i)} = y,\, x_j^{(i)} = 1)$.



Predictions for test cases (I)

- Using only retained features:

$$P(y^* \mid x^*_{1:k},\, x^{\text{train}}_{1:k},\, y^{\text{train}}) \quad = \quad \frac{P(y^* \mid y^{\text{train}})\, P(x^*_{1:k} \mid y^*,\, x^{\text{train}}_{1:k},\, y^{\text{train}})}{\displaystyle\sum_{y=0}^{1} P(y^* = y \mid y^{\text{train}})\, P(x^*_{1:k} \mid y^* = y,\, x^{\text{train}}_{1:k},\, y^{\text{train}})}$$

Predictions for test cases (I)

- Using only retained features:

$$P(y^* \mid x^*_{1:k},\, x^{\text{train}}_{1:k},\, y^{\text{train}}) \;=\; \frac{P(y^* \mid y^{\text{train}})\, P(x^*_{1:k} \mid y^*,\, x^{\text{train}}_{1:k},\, y^{\text{train}})}{\sum_{y=0}^{1} P(y^* = y \mid y^{\text{train}})\, P(x^*_{1:k} \mid y^* = y,\, x^{\text{train}}_{1:k},\, y^{\text{train}})}$$

$$P(x^*_{1:k} \mid y^*,\, x^{\text{train}}_{1:k},\, y^{\text{train}}) \;=\; \frac{P(x^*_{1:k},\, x^{\text{train}}_{1:k} \mid y^*,\, y^{\text{train}})}{P(x^{\text{train}}_{1:k} \mid y^{\text{train}})} \;\propto\; P(x^*_{1:k},\, x^{\text{train}}_{1:k} \mid y^*,\, y^{\text{train}})$$

Predictions for test cases (I)

- Using only retained features:

$$P(y^* \mid x^*_{1:k},\, x^{\text{train}}_{1:k},\, y^{\text{train}}) \;=\; \frac{P(y^* \mid y^{\text{train}})\, P(x^*_{1:k} \mid y^*,\, x^{\text{train}}_{1:k},\, y^{\text{train}})}{\sum_{y=0}^{1} P(y^* = y \mid y^{\text{train}})\, P(x^*_{1:k} \mid y^* = y,\, x^{\text{train}}_{1:k},\, y^{\text{train}})}$$

$$P(x^*_{1:k} \mid y^*,\, x^{\text{train}}_{1:k},\, y^{\text{train}}) \;=\; \frac{P(x^*_{1:k},\, x^{\text{train}}_{1:k} \mid y^*,\, y^{\text{train}})}{P(x^{\text{train}}_{1:k} \mid y^{\text{train}})} \;\propto\; P(x^*_{1:k},\, x^{\text{train}}_{1:k} \mid y^*,\, y^{\text{train}})$$

$$\begin{aligned} P(x^*_{1:k},\, x^{\text{train}}_{1:k} \mid y^*,\, y^{\text{train}}) &= \int P(\alpha)\, P(x^*_{1:k},\, x^{\text{train}}_{1:k} \mid \alpha,\, y^*,\, y^{\text{train}})\, d\alpha \\ &= \int P(\alpha) \prod_{j=1}^{k} P(x^*_j,\, x^{\text{train}}_j \mid \alpha,\, y^*,\, y^{\text{train}})\, d\alpha \end{aligned}$$

The above integral about α is approximated with Midpoint Rule.

Predictions for test cases (II)

$$
\begin{aligned}
& P(x_j^*,\, x_j^{\text{train}} \mid \alpha,\, y^*,\, y^{\text{train}}) \\
&= \int_0^1 P(x_j^* \mid \theta_j,\, \alpha,\, x_j^{\text{train}},\, y^{\text{train}},\, y^*)\, P(x_j^{\text{train}} \mid \theta_j,\, \alpha,\, y^{\text{train}})\, d\theta_j \\
&= \int_0^1 \text{Bernoulli}\,(x_j^*; \hat{\phi}_{y^*,j}) \prod_{y=0}^{1} U(\alpha\theta_j,\, \alpha(1-\theta_j),\, I_{y,j},\, O_{y,j})\, d\theta_j
\end{aligned}
$$

The above integral about θ_j is approximated with Simpson's Rule.

Predictions for test cases (II)

$$
\begin{aligned}
&P(x_j^*,\, x_j^{\text{train}} \mid \alpha,\, y^*,\, y^{\text{train}}) \\
&\quad = \int_0^1 P(x_j^* \mid \theta_j,\, \alpha,\, x_j^{\text{train}},\, y^{\text{train}},\, y^*)\, P(x_j^{\text{train}} \mid \theta_j,\, \alpha,\, y^{\text{train}})\, d\theta_j \\
&\quad = \int_0^1 \text{Bernoulli}\,(x_j^*; \hat{\phi}_{y^*,j}) \prod_{y=0}^{1} U(\alpha\theta_j,\, \alpha(1-\theta_j),\, I_{y,j},\, O_{y,j})\, d\theta_j
\end{aligned}
$$

The above integral about θ_j is approximated with Simpson's Rule.

- With correction for feature selection

$P(x_{1:k}^*,\, x_{1:k}^{\text{train}},\, \mid y^*,\, y^{\text{train}})$ is replaced with $P(x_{1:k}^*,\, x_{1:k}^{\text{train}}, \mathcal{S} \mid y^*,\, y^{\text{train}})$:

$$
P(x_{1:k}^*,\, x_{1:k}^{\text{train}},\, \mathcal{S} \mid y^*,\, y^{\text{train}}) = \int P(\alpha) P(\mathcal{S} \mid \alpha, y^{\text{train}}) \prod_{j=1}^{k} P(x_j^*, x_j^{\text{train}} \mid \alpha, y^*, y^{\text{train}}) d\alpha
$$

where $P(\mathcal{S} \mid \alpha,\, y^{\text{train}}) = \Big[P(|\text{COR}(y^{\text{train}}, x_t^{\text{train}})| \le \gamma \mid \alpha,\, y^{\text{train}}) \Big]^{p-k}$

Computation of the adjustment factor (I)

$\text{COR}(x_t^{\text{train}}, y^{\text{train}})$ can be written in terms of $I_0 = \sum_{i=1}^n I(y^{(i)} = 0,\, x_t^{(i)} = 1)$ and $I_1 = \sum_{i=1}^n I(y^{(i)} = 1,\, x_t^{(i)} = 1)$:

$$
\begin{aligned}
\text{COR}(x_t^{\text{train}},\, y^{\text{train}}) &= \frac{\displaystyle\sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right) x_t^{(i)}}{\sqrt{\displaystyle\sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right)^2} \sqrt{\displaystyle\sum_{i=1}^{n} \left(x_t^{(i)} - \bar{x}_t\right)^2}} \\
&= \frac{(0 - \overline{y})\, I_0 \;+\; (1 - \overline{y})\, I_1}{\sqrt{n\overline{y}(1-\overline{y})}\, \sqrt{I_0 + I_1 - (I_0 + I_1)^2/n}}
\end{aligned}
$$

I_0,I_1 are visualized:

$$
\begin{array}{ll}
y^{\text{train}} : & 0 \quad 0 \quad 0 \quad 0 \;\vdots\; 1 \quad 1 \quad 1 \quad 1 \\[1em]
x_t^{\text{train}} : & \underbrace{0}_{\boldsymbol{O}_0} \quad \underbrace{1 \quad 1 \quad 1}_{\boldsymbol{I}_0} \;\vdots\; \underbrace{0 \quad 0 \quad 0}_{\boldsymbol{O}_1} \quad \underbrace{1}_{\boldsymbol{I}_1}
\end{array}
$$

Computation of the adjustment factor (II)

An example showing values of $|\text{COR}(x_t^{\text{train}}, y^{\text{train}})|$ in terms of I_0 and I_1:

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
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

Computation of the adjustment factor (III)

- $P(I_0,\, I_1 \mid \alpha,\, y^{\text{train}})$ is symmetric for H_+ and H_-, so

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

- Conditioning on θ_t:

$$\sum_{(I_0,I_1)\,\in\, H_+} P(I_0,\, I_1 \mid \alpha,\, y^{\text{train}}) \;=\; \int_0^1 \sum_{(I_0,I_1)\,\in\, H_+} P(I_0,\, I_1 \mid \alpha,\, \theta_t,\, y^{\text{train}})\, d\theta_t$$

The above integration about θ_t is approximated with Simpson's Rule.

- $|\text{COR}(x_t^{\text{train}}, y^{\text{train}})|$ is monotone with respect to I_0 or I_1, so

$$\sum_{(I_0,I_1)\,\in\, H_+} P(I_0,\, I_1 \mid \alpha,\, \theta_t,\, y^{\text{train}}) \;=\; \sum_{I_1=b_0}^{n\overline{y}} \sum_{I_0=0}^{r_{I_1}} P(I_0,\, I_1 \mid \alpha,\, \theta_t,\, y^{\text{train}})$$

- I_0 and I_1 are independent given $\alpha,\, \theta_t,\, y^{\text{train}}$, so

$$P(I_0,\, I_1 \mid \alpha,\, \theta_t,\, y^{\text{train}}) \;=\; P(I_1 \mid \alpha,\, \theta_t,\, y^{\text{train}}) P(I_0 \mid \alpha,\, \theta_t,\, y^{\text{train}})$$

Computation of the adjustment factor (IV)

Finally, we can easily compute probability of I_0 and I_1:

$$P(I_1 \mid \alpha,\, \theta_t,\, y^{\text{train}}) \;=\; \binom{N_1}{I_1} U(\alpha\theta_t,\, \alpha(1-\theta_t),\, I_1,\, N_1 - I_1)$$

and

$$P(I_0 \mid \alpha,\, \theta_t,\, y^{\text{train}}) \;=\; \binom{n - N_1}{I_0} U(\alpha\theta_t,\, \alpha(1-\theta_t),\, I_0,\, n - N_1 - I_0)$$

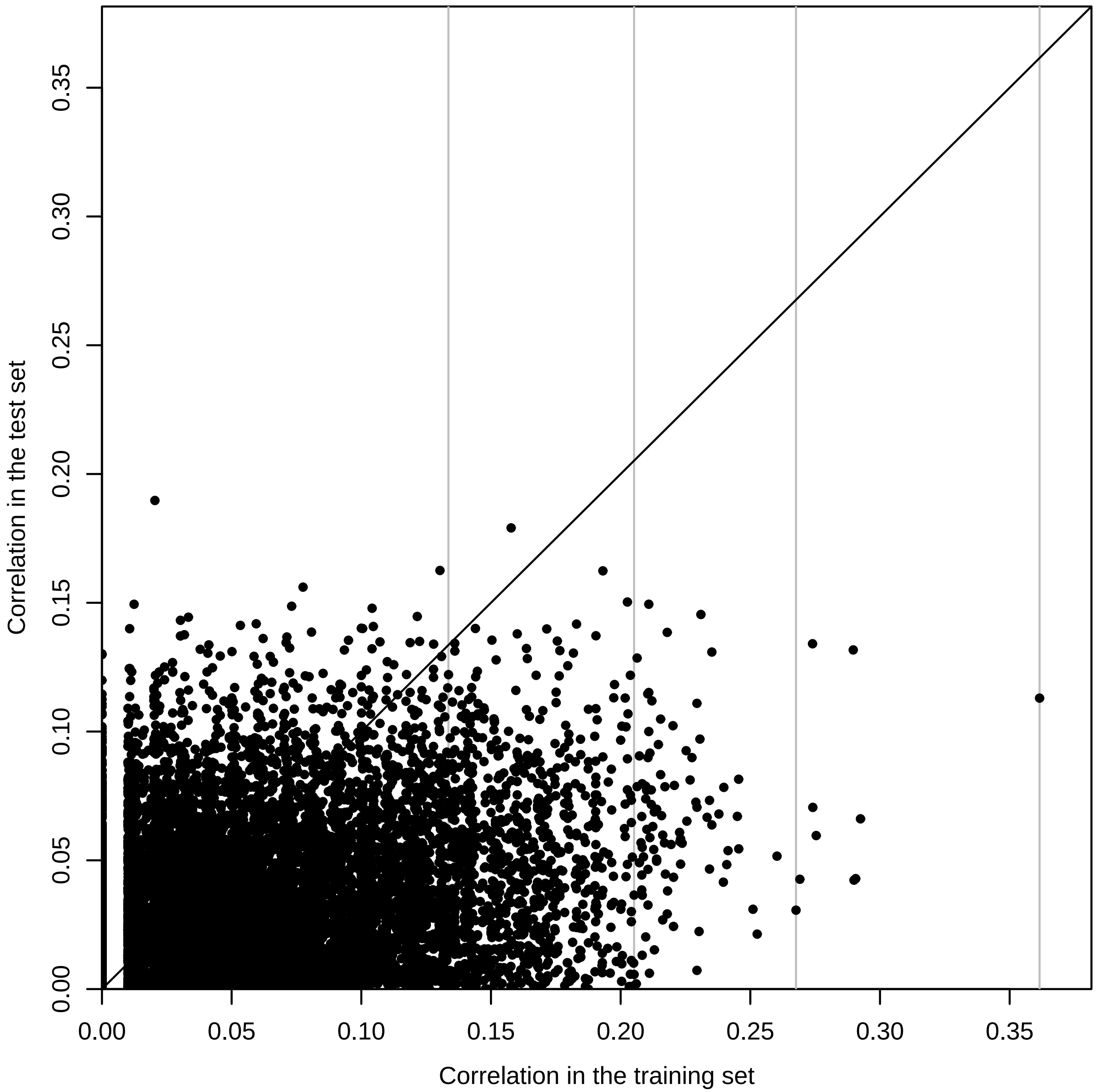
A Simulation Experiment

- Generating data: $\alpha = 300$, $p = 10000$, 100 training cases, 2000 test cases

- Selecting features: 4 subsets with only 1, 10, 100 and 1000 features were selected, with smallest absolute value of correlation being 0.36,0.27,0.21, and 0.13.

- Prior: $f_0 = f_1 = 1$, $a = 0.5$, $b = 5$

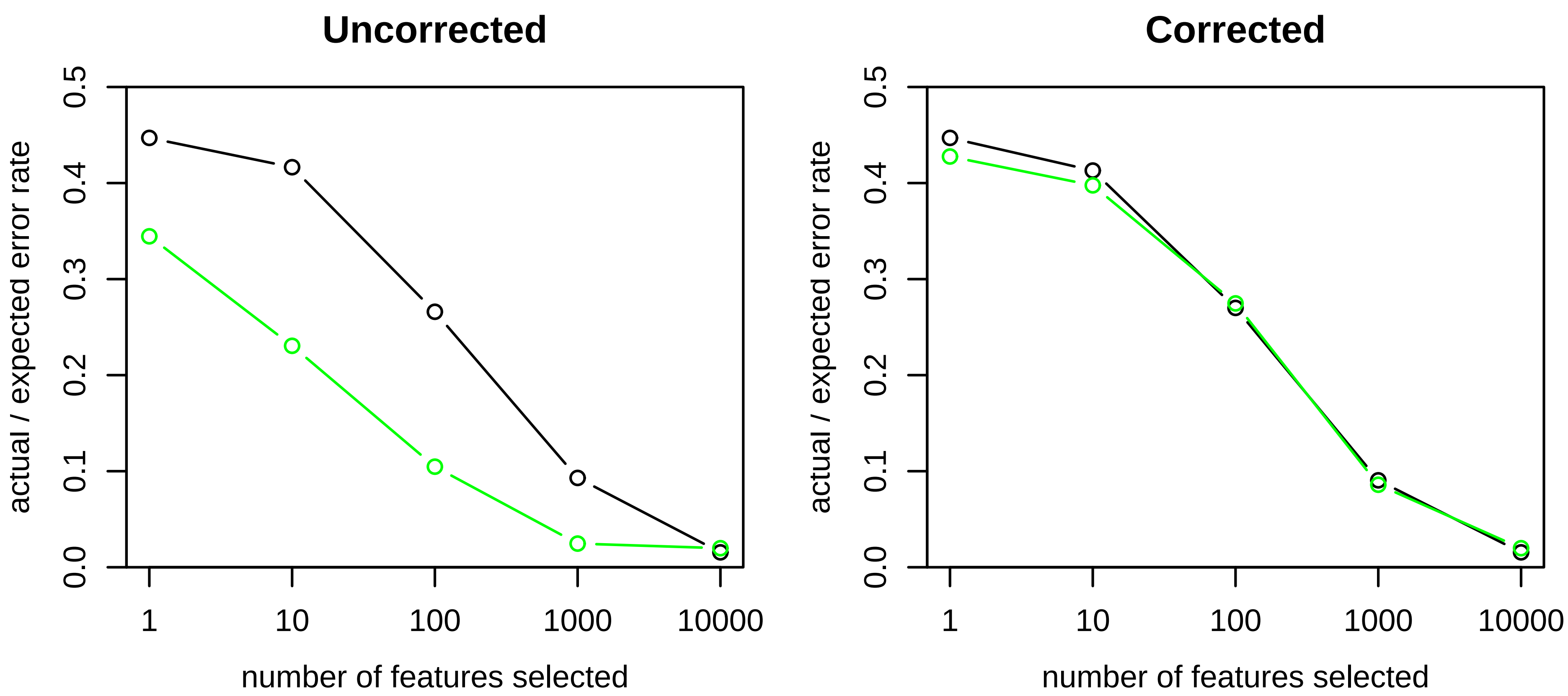
Looking into Feature Selection



Comparison of Actual and Expected Error Rates

Average Expected Error Rate:

$$(1/N)\sum_{i=1}^{N}\hat{p}^{(i)}\, I(\hat{p}^{(i)} < 1/2) + (1-\hat{p}^{(i)})\, I(\hat{p}^{(i)} \geq 1/2)$$



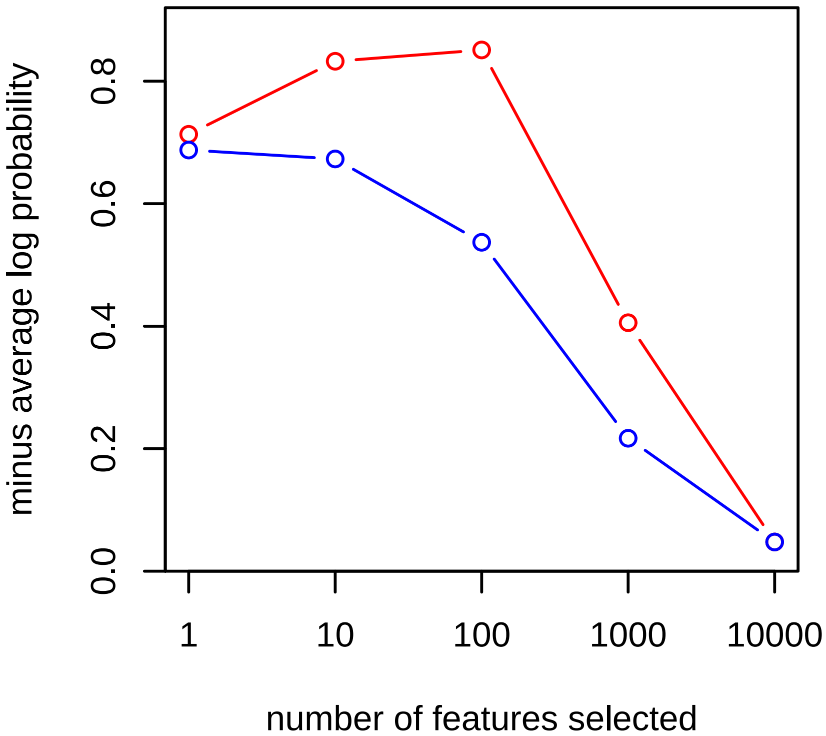
Comparison of Two Measures

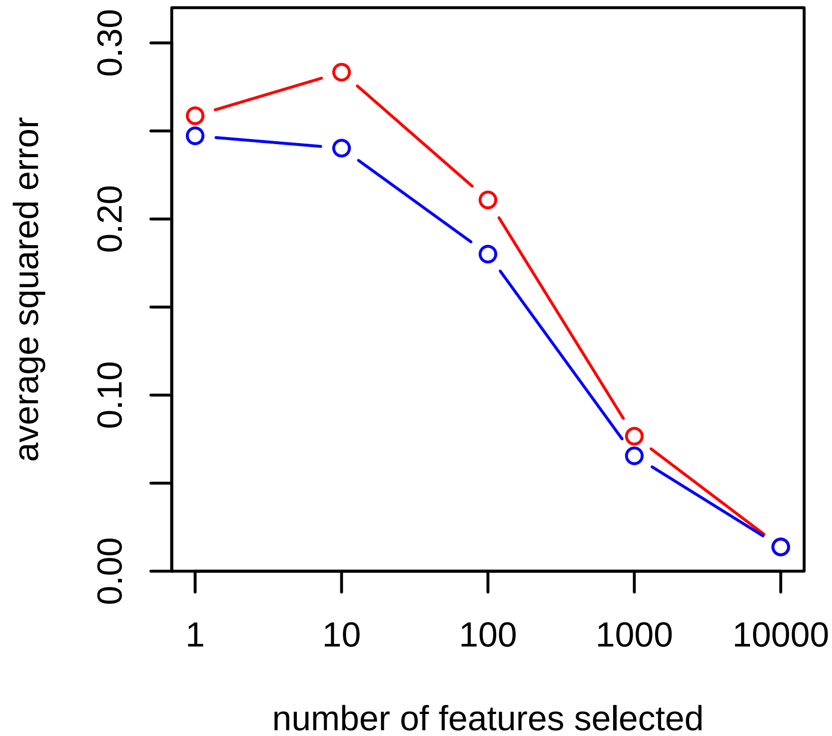
Average Minus Log Probability:

$$-(1/N)\sum_{i=1}^{N}[y^{(i)}\log(\hat{p}^{(i)}) + (1-y^{(i)})\log(1-\hat{p}^{(i)})]$$

Average Squared Error:

$$(1/N)\sum_{i=1}^{N}(y^{(i)} - \hat{p}^{(i)})^2$$

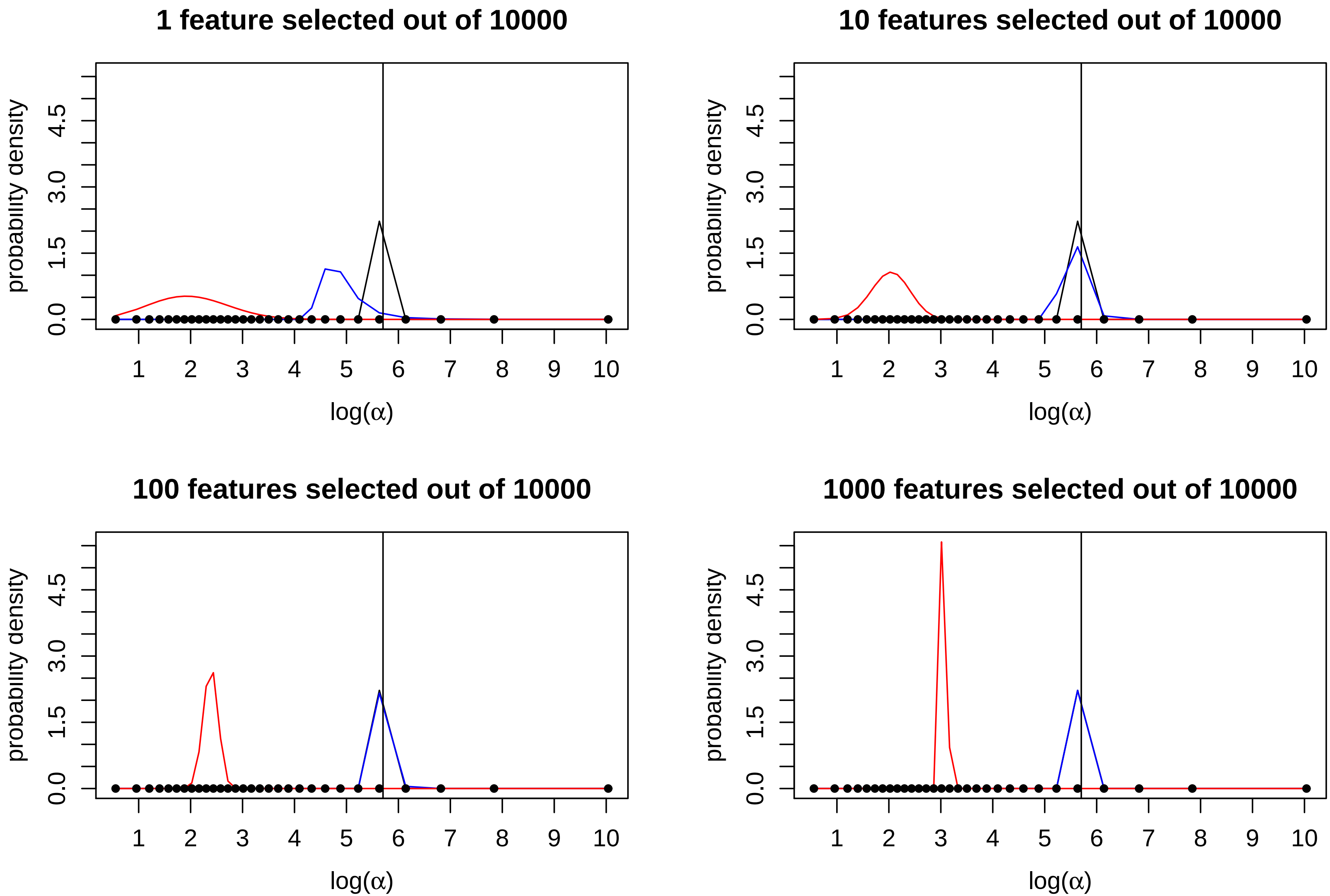




Comparison of Calibration for Predictions

	1000 features selected out of 10000					
	Corrected			Uncorrected		
Category	#	Pred	Actual	#	Pred	Actual
0.0 - 0.1	774	0.018	0.027	954	0.004	0.066
0.1 - 0.2	97	0.143	0.165	28	0.149	0.500
0.2 - 0.3	63	0.243	0.302	13	0.248	0.846
0.3 - 0.4	48	0.346	0.438	17	0.349	0.412
0.4 - 0.5	45	0.446	0.600	14	0.449	0.786
0.5 - 0.6	44	0.547	0.614	16	0.546	0.375
0.6 - 0.7	53	0.647	0.698	16	0.667	0.812
0.7 - 0.8	81	0.755	0.815	22	0.751	0.636
0.8 - 0.9	124	0.854	0.863	25	0.865	0.560
0.9 - 1.0	671	0.977	0.982	895	0.995	0.946

Posterior Distribution of $\log(\alpha)$



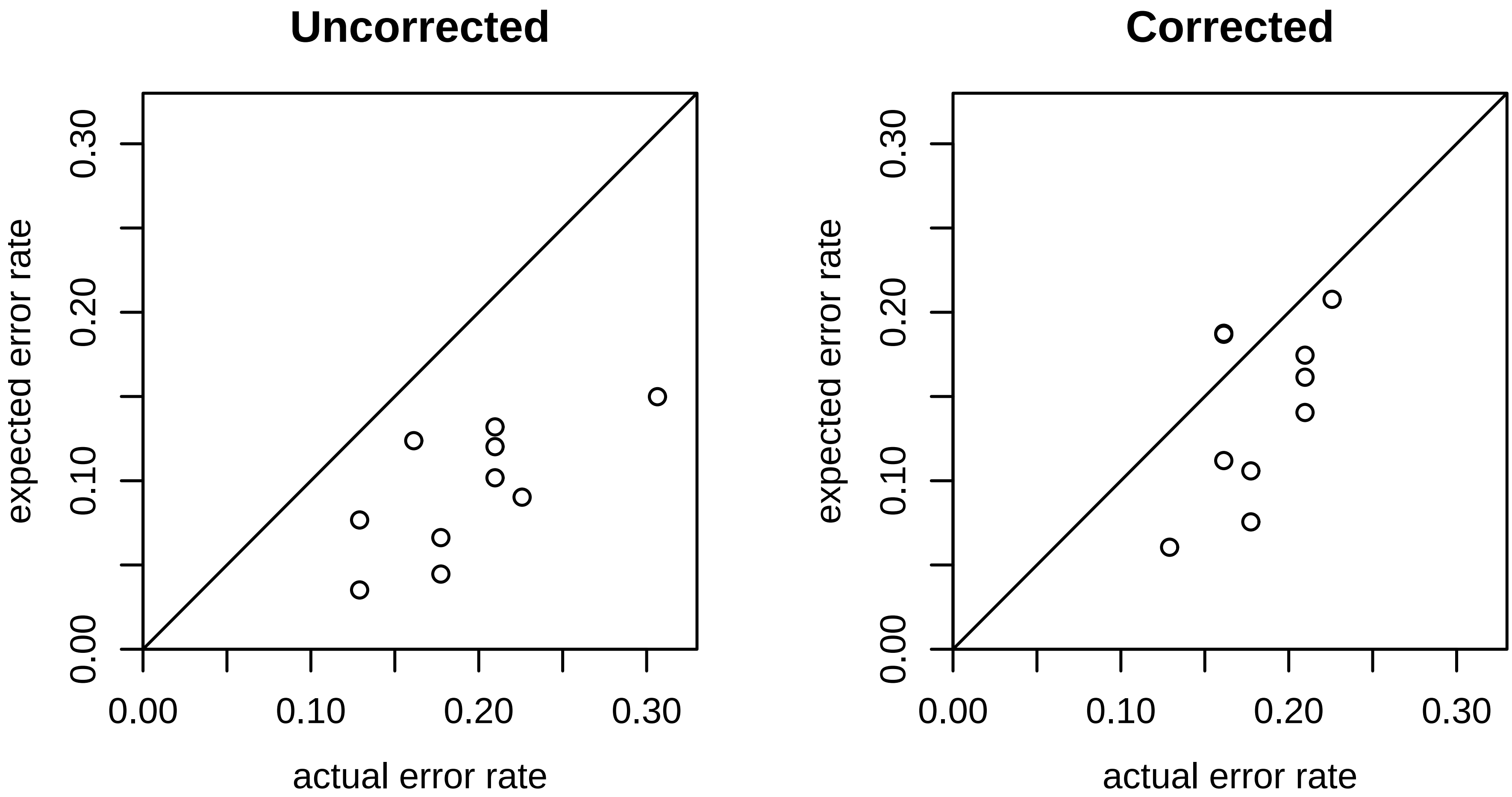
Computational Time

# of Features Selected	1	10	100	1000	10000
Uncorrected	12.66	28.80	203.60	2065.73	20627.71
Corrected	12.72	29.55	204.54	2076.21	

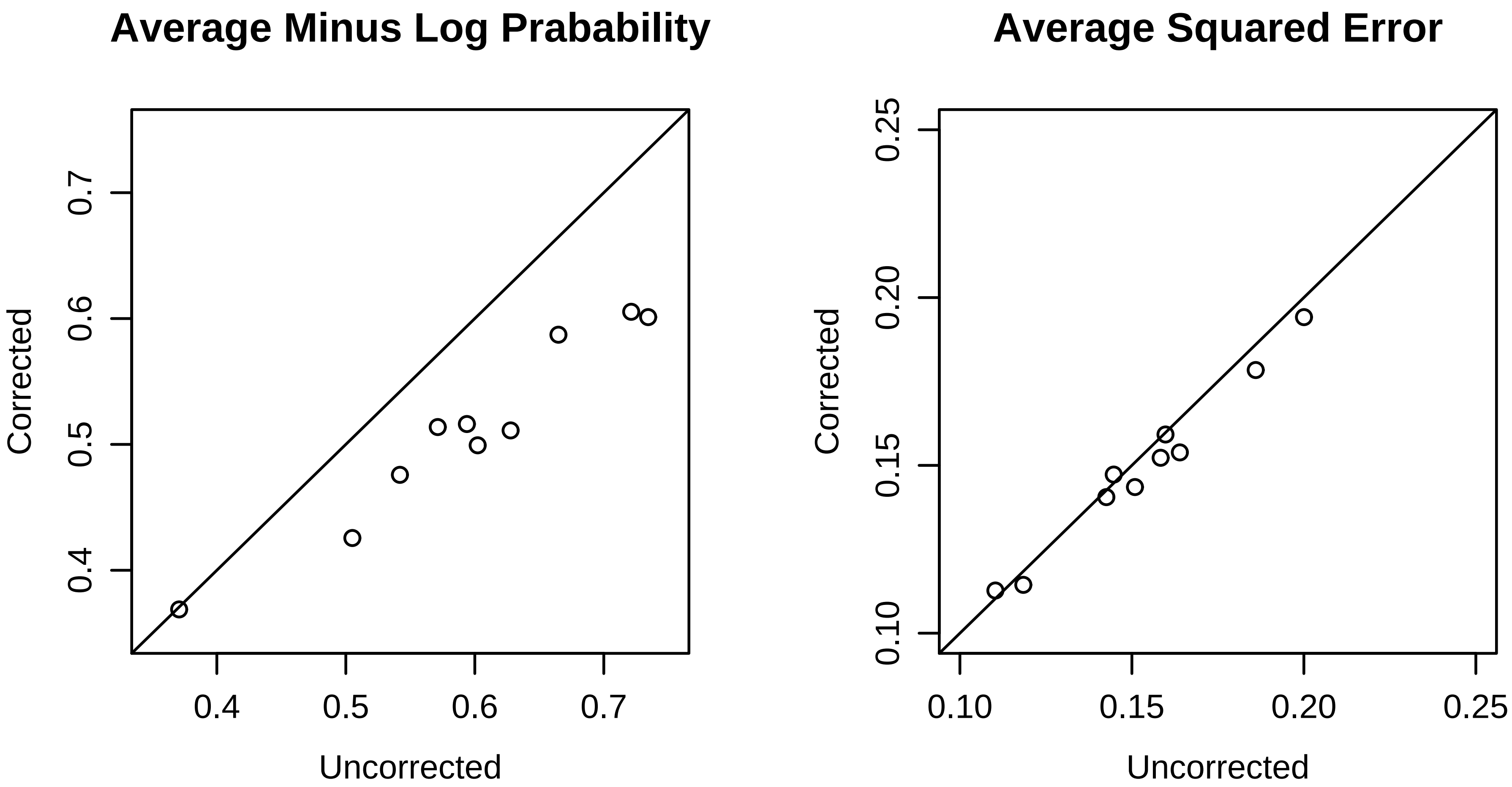
A Test with Gene Expression Data

- Data set: 2000 genes, 62 cases (40 Cancereous vs 22 Normal tissues)
- Converted into binary data by thresholding at medians of features
- Splited 2000 genes randomly into 10 subsets each with 200 genes.
- 5 Genes were selected out of 200 genes
- Used leave-one-out cross-validation to obtain predictive probabilities
- Prior the same as previous simulation experiment

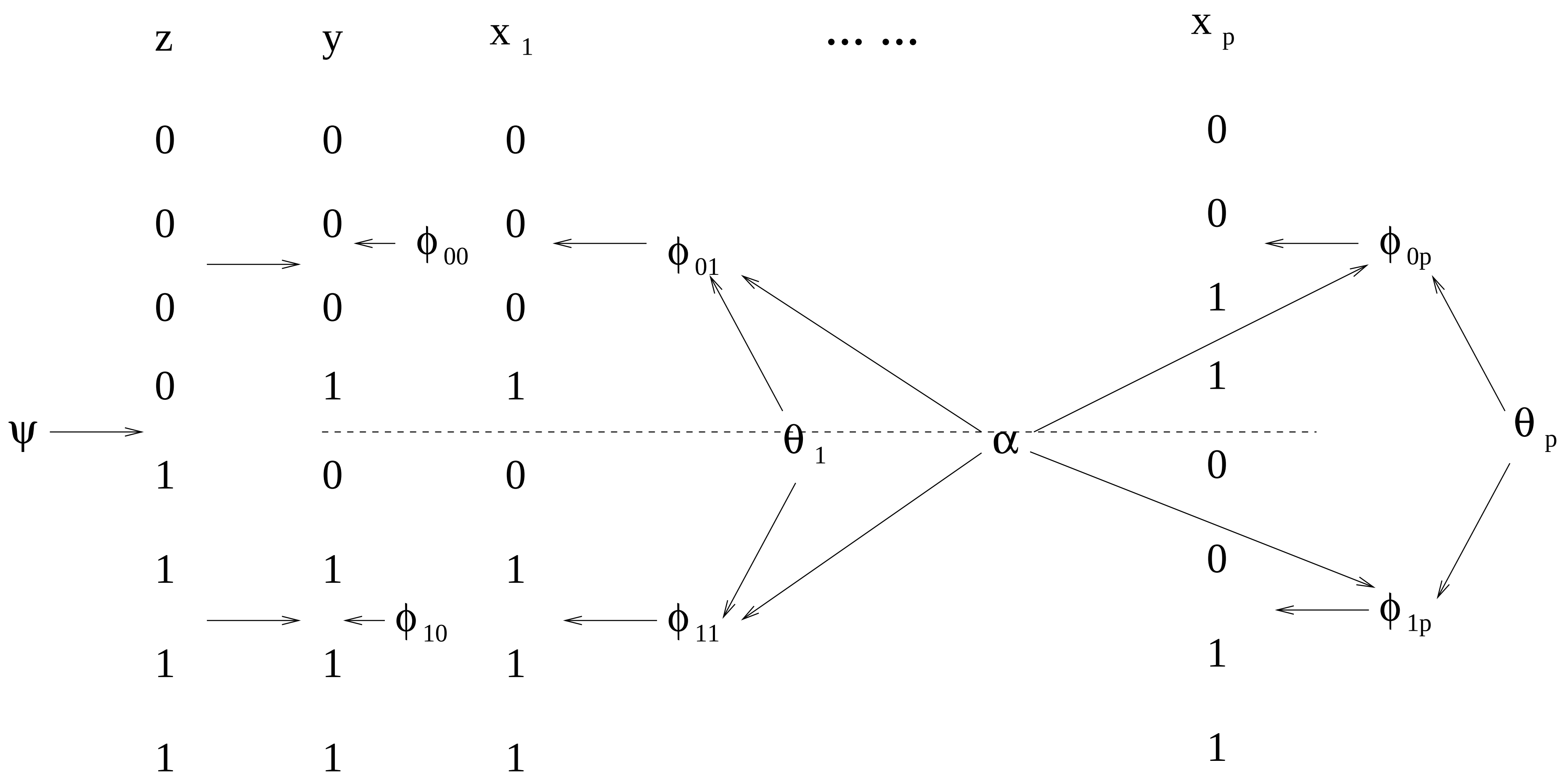
Comparison of Actual and Expected Error Rates



Comparison of Two Measures



Extended to Binary Mixture Models



Conclusion and Discussion

- We've proposed a correction method to avoid the bias from feature selection.

Conclusion and Discussion

- We've proposed a correction method to avoid the bias from feature selection.
- We've applied the method to binary naive Bayes models. The simulation results show that it does avoid the bias from feature selection and it is faster than using all features. The results from microarray datasets show that the corrected method improves the predictive performance.

Conclusion and Discussion

- We've proposed a correction method to avoid the bias from feature selection.
- We've applied the method to binary naive Bayes models. The simulation results show that it does avoid the bias from feature selection and it is faster than using all features. The results from microarray datasets show that the corrected method improves the predictive performance.
- The computation of the adjustment factor is difficult generally. For binary naive Bayes models, the method we used is very fast. This method can be generalized to all discrete naive Bayes models.

Conclusion and Discussion

- We've proposed a correction method to avoid the bias from feature selection.
- We've applied the method to binary naive Bayes models. The simulation results show that it does avoid the bias from feature selection and it is faster than using all features. The results from microarray datasets show that the corrected method improves the predictive performance.
- The computation of the adjustment factor is difficult generally. For binary naive Bayes models, the method we used is very fast. This method can be generalized to all discrete naive Bayes models.
- We've applied the correction method to binary mixture models and factor analysis models. Computation of the adjustment factor for these two models is more difficult but still feasible. Future work could be done to improve efficiency of computing adjustment factor.

Conclusion and Discussion

- We've proposed a correction method to avoid the bias from feature selection.
- We've applied the method to binary naive Bayes models. The simulation results show that it does avoid the bias from feature selection and it is faster than using all features. The results from microarray datasets show that the corrected method improves the predictive performance.
- The computation of the adjustment factor is difficult generally. For binary naive Bayes models, the method we used is very fast. This method can be generalized to all discrete naive Bayes models.
- We've applied the correction method to binary mixture models and factor analysis models. Computation of the adjustment factor for these two models is more difficult but still feasible. Future work could be done to improve efficiency of computing adjustment factor.
- One could extend the method to other models and other selection criteria.