Avoiding Bias from Feature Selection

Longhai Li

Based on a paper coauthored with Jianguo Zhang and Radford Neal

longhai@math.usask.ca
http://math.usask.ca/~longhai

Department of Mathematics and Statistics

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

CRiSM Workshop “Bayesian Analysis of High-dimensional Data”
University of Warwick, United Kingdom

16 April 2008

Outline of the talk

- Why do We Need to Select Features?
- Problems with Feature Selection — the optimistic bias
- Our Solution for Avoiding Bias from Feature Selection
- Application to Naive Bayes Classification Model
  - Definition of a Naive Bayes Classification Model
  - Computation of Adjustment Factor for Avoiding Bias
  - Demonstration with Simulated and Real Datasets
- Conclusions and Discussions
- Future Work

Review of Statistical Classification

- 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:

$$\hat{y} = C(x).$$

Review of Statistical Classification

- 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:

$$\hat{y} = C(x).$$

- How do we find $\hat{y}$?

Review of Statistical Classification

- 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:

$$\hat{y} = C(x).$$

- How do we find $\hat{y}$?
  - First find the predictive distribution of y given x:

$$P(y|x)$$

Review of Statistical Classification

- 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:

$$\hat{y} = C(x).$$

- How do we find $\hat{y}$?
  - First find the predictive distribution of y given x:

$$P(y|x)$$

  - Given x, predict y with $\hat{y}$ that minimizes expected loss $E(L(y, \hat{y})|x)$. e.g., y is binary, and $L(y = 0, \hat{y} = 1)/L(y = 1, \hat{y} = 0) = r$,

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|x) \geq 1 - \frac{1}{1+r}, \\ 0 & \text{if } P(y = 1|x) < 1 - \frac{1}{1+r} \end{cases}$$

Review of Statistical Classification

- 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:

$$\hat{y} = C(x).$$

- How do we find $\hat{y}$?
  - First find the predictive distribution of y given x:

$$P(y|x)$$

  - Given x, predict y with $\hat{y}$ that minimizes expected loss $E(L(y, \hat{y})|x)$. e.g., y is binary, and $L(y = 0, \hat{y} = 1)/L(y = 1, \hat{y} = 0) = r$,

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|x) \geq 1 - \frac{1}{1+r}, \\ 0 & \text{if } P(y = 1|x) < 1 - \frac{1}{1+r} \end{cases}$$

- Statistical Method: Estimate $P(y|x)$ 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.

Review of Statistical Classification

- 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:

$$\hat{y} = C(x).$$

- How do we find $\hat{y}$?
  - First find the predictive distribution of y given x:

$$P(y|x)$$

  - Given x, predict y with $\hat{y}$ that minimizes expected loss $E(L(y, \hat{y})|x)$. e.g., y is binary, and $L(y = 0, \hat{y} = 1)/L(y = 1, \hat{y} = 0) = r$,

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|x) \geq 1 - \frac{1}{1+r}, \\ 0 & \text{if } P(y = 1|x) < 1 - \frac{1}{1+r} \end{cases}$$

- Statistical Method: Estimate $P(y|x)$ 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.
- Example: Given gene expression data of a patient, classify the type of tumour.

Why do We Need to Select Features?

- Examples of High Dimensional Features

Why do We Need to Select Features?

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

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features:

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features:
  - In Time: Computation time grows with the number of features.

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features:
  - In Time: Computation time grows with the number of features.
  - In Money: Costly in measuring the features for future cases.

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features:
  - In Time: Computation time grows with the number of features.
  - In Money: Costly in measuring the features for future cases.
  - In Statistics:

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features:
  - In Time: Computation time grows with the number of features.
  - In Money: Costly in measuring the features for future cases.
  - In Statistics:
    - When the number of observations is smaller than the number of parameters, the likelihood function $P(x^{\text{train}}, y^{\text{train}}|\theta)$ based on a simple model, such as a linear model, will favour "too good" θ.

Why do We Need to Select 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 occurrence) of all words in a large dictionary, which may have, for example, 30,000 words.
- Difficulties with High Dimensional Features:
  - In Time: Computation time grows with the number of features.
  - In Money: Costly in measuring the features for future cases.
  - In Statistics:
    - When the number of observations is smaller than the number of parameters, the likelihood function $P(x^{\text{train}}, y^{\text{train}}|\theta)$ based on a simple model, such as a linear model, will favour "too good" θ.
    - Our models may fail when applied to high-dimensional features, e.g., we may not be able to model the dependency between features very well; our priors for high-dimensional parameters may be inappropriate.

Optimistic Bias from Feature Selection

Feature Selection:

For previous reasons or others, one may like to select a subset of features to use based on some measure of the dependency (e.g. absolute correlation) between the features and y. There are also more sophisticated Bayesian/non-Bayesian methods by looking at how well the subset can predict y (e.g.,LASSO, Bayesian methods using a prior with mass at 0 for coefficients).

Optimistic Bias from Feature Selection

Feature Selection:

For previous reasons or others, one may like to select a subset of features to use based on some measure of the dependency (e.g. absolute correlation) between the features and y. There are also more sophisticated Bayesian/non-Bayesian methods by looking at how well the subset can predict y (e.g.,LASSO, Bayesian methods using a prior with mass at 0 for coefficients).

Optimistic Bias:

However, such procedures will make the relationship between y and x *appears* stronger than it actually is, i.e., the response *appears* more predictable from the features. We will be overconfident with our prediction.

Optimistic Bias from Feature Selection

Feature Selection:

For previous reasons or others, one may like to select a subset of features to use based on some measure of the dependency (e.g. absolute correlation) between the features and y. There are also more sophisticated Bayesian/non-Bayesian methods by looking at how well the subset can predict y (e.g.,LASSO, Bayesian methods using a prior with mass at 0 for coefficients).

Optimistic Bias:

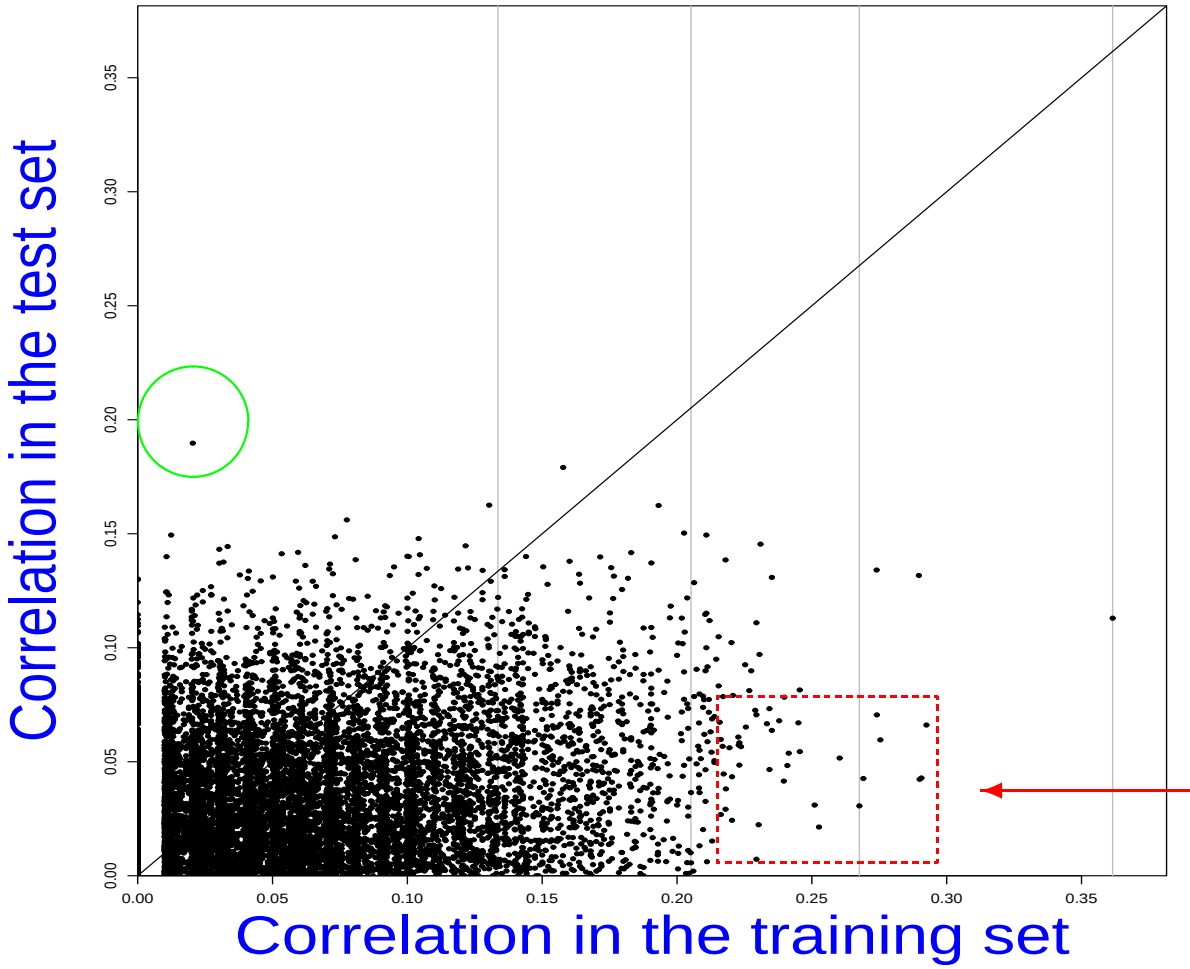
However, such procedures will make the relationship between y and x *appears* stronger than it actually is, i.e., the response *appears* more predictable from the features. We will be overconfident with our prediction.

An Extreme Example:

All features $x_1, \ldots, x_p$ and the response y are actually independent. We select only k features $x_1^*, \ldots, x_k^*$ using the criterion of absolute correlation. These k features will exhibit strong relationship with y in training data, when $p \gg k$.

What Causes Bias from Feature Selection?

Let's look at the absolute correlations of $10,000$ binary features with a binary response from a data set simulated using a method to be discussed later.

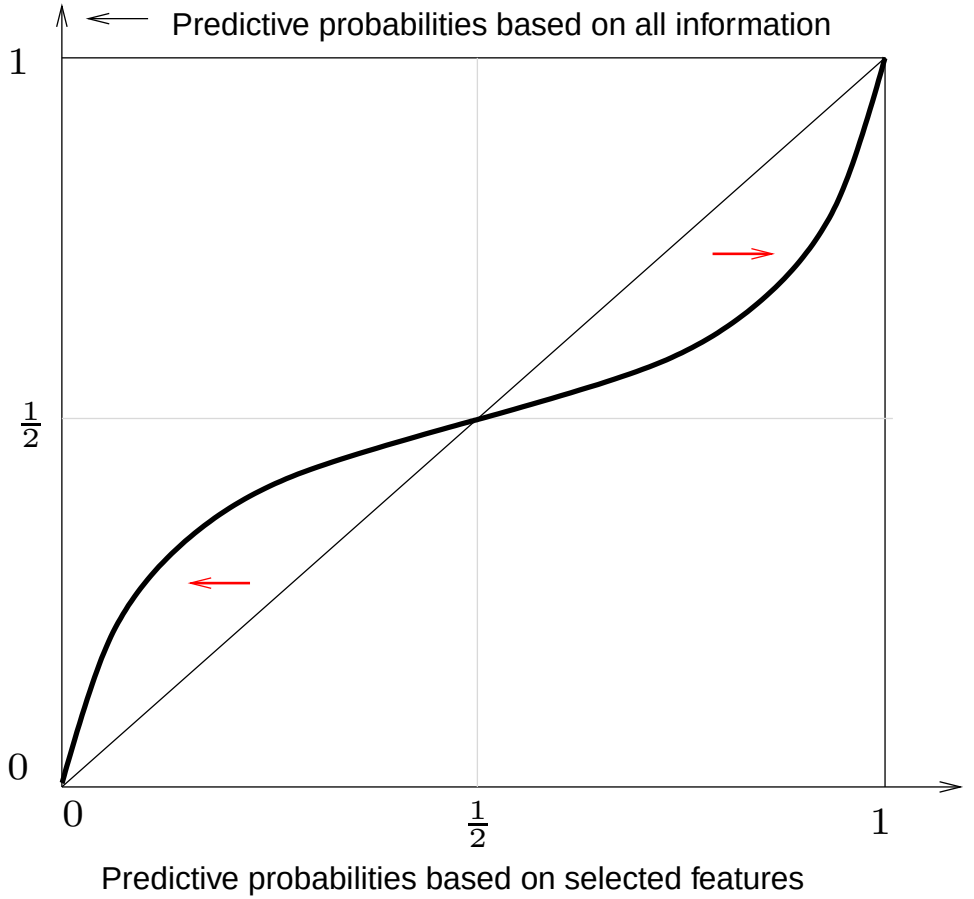


When we measure more features, this bias may be more severe, as more "fake" features compete with our "signal" features.

The absolute correlations of these features with y are high in training set, but very low in test set.

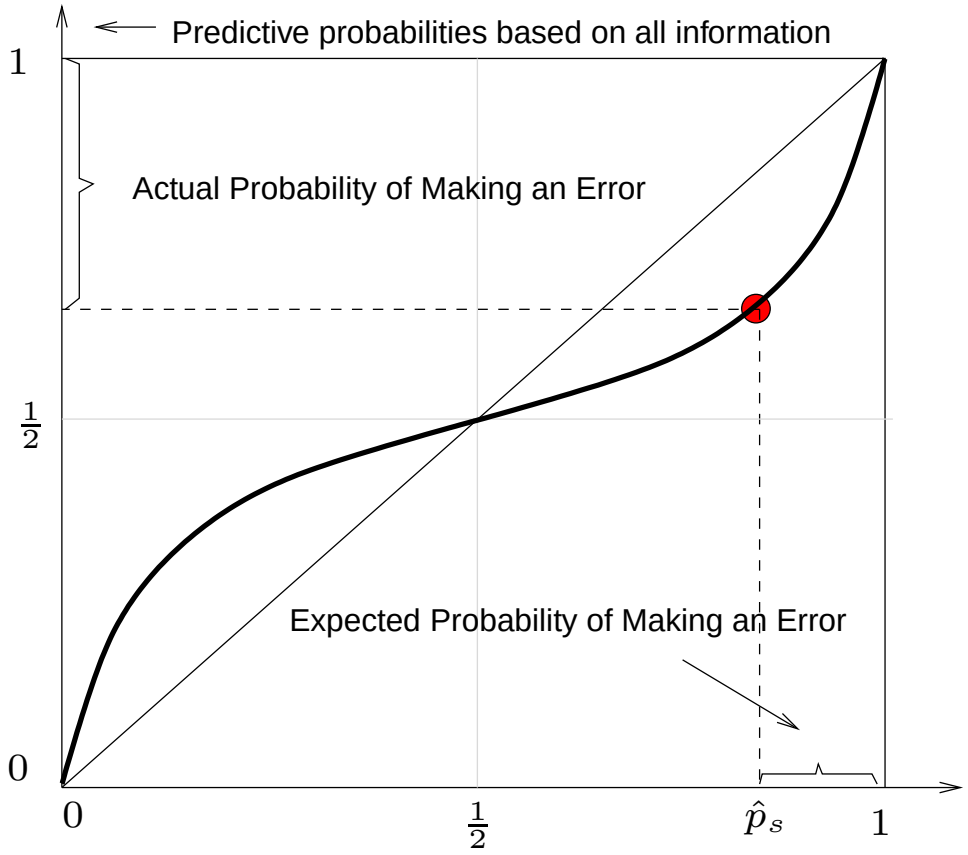
Problems Resulting from Feature Selection Bias (I)

The following graph displays a typical plot of predictive probabilities based on selected features versus the actual predictive probabilities based on all information available:



Problems Resulting from Feature Selection Bias (I)

The following graph displays a typical plot of predictive probabilities based on selected features versus the actual predictive probabilities based on all information available:



The predictive probabilities are over-confident. For example, if we use $\frac{1}{2}$ as cutoff for predicting the class labels, expected error rate,

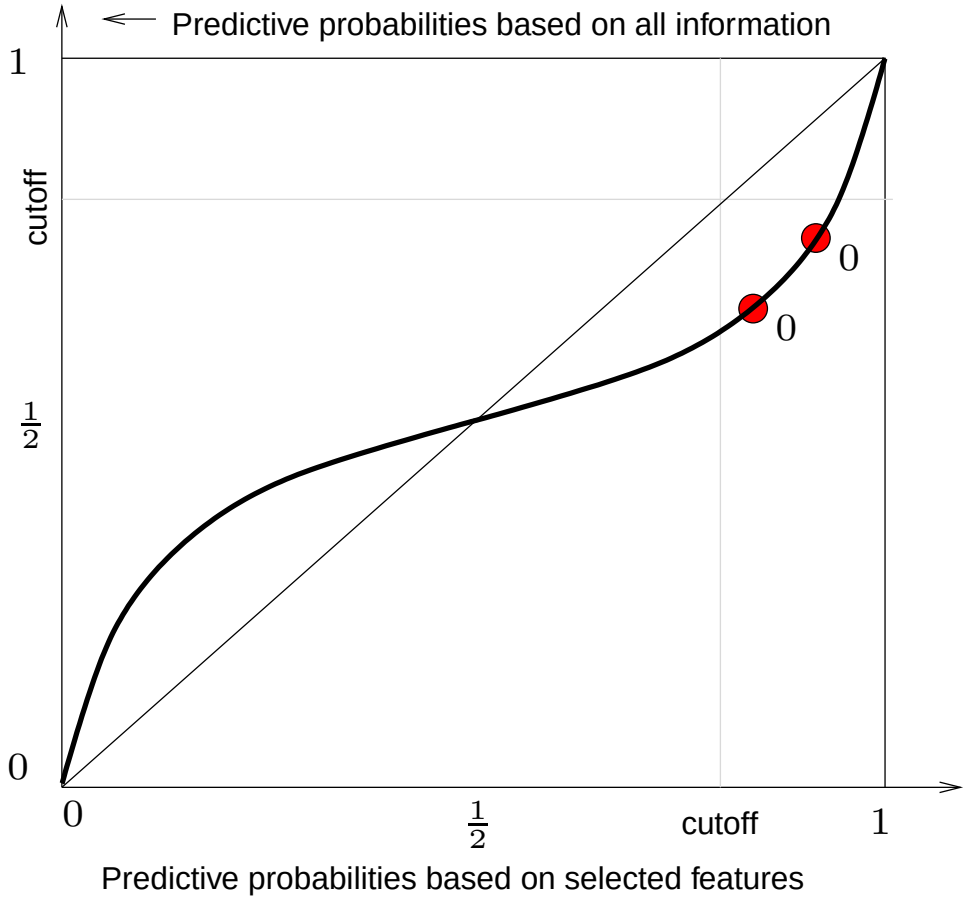
$$\frac{1}{n}\sum_{i=1}^{n}\left(\hat{p}_s^{(i)} I(\hat{p}_s^{(i)} < \frac{1}{2}) + (1-\hat{p}_s^{(i)}) I(\hat{p}_s^{(i)} \geq \frac{1}{2})\right),$$

is smaller than actual error rate

$$\frac{1}{n}\sum_{i=1}^{n} I(y^{(i)} \neq \hat{y}_s^{(i)})$$

Problems Resulting from Feature Selection Bias (II)

More seriously, when the loss by misclassifying 0 to 1 is different from the loss by misclassifying 1 to 0, the prediction based on selected features results in more loss in practice than that based on all information available, i.e.,



$$E(L(\hat{y}_s^{(*)}, y^{(*)})) \geq E(L(\hat{y}_B^{(*)}, y^{(*)}))$$

where,

$$\hat{y}_s^{(*)} = \begin{cases} 1 & \text{if } \hat{p}_s^{(*)} \geq \text{cutoff} \\ 0 & \text{if } \hat{p}_s^{(*)} < \text{cutoff} \end{cases},$$

and cutoff = $1 - \frac{1}{1+r}$, and $r = \frac{L_{0\to 1}}{L_{1\to 0}}$. $\hat{y}_B^{(*)}$ is similar, but using the predictive probability $\hat{p}_B^{(*)}$ based on all information available.

Our Method for Avoiding Features Selection Bias

- All available information: 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:

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

Our Method for Avoiding Features Selection Bias

- All available information: 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:

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

Our Method for Avoiding Features Selection Bias

- All available information: 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:

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

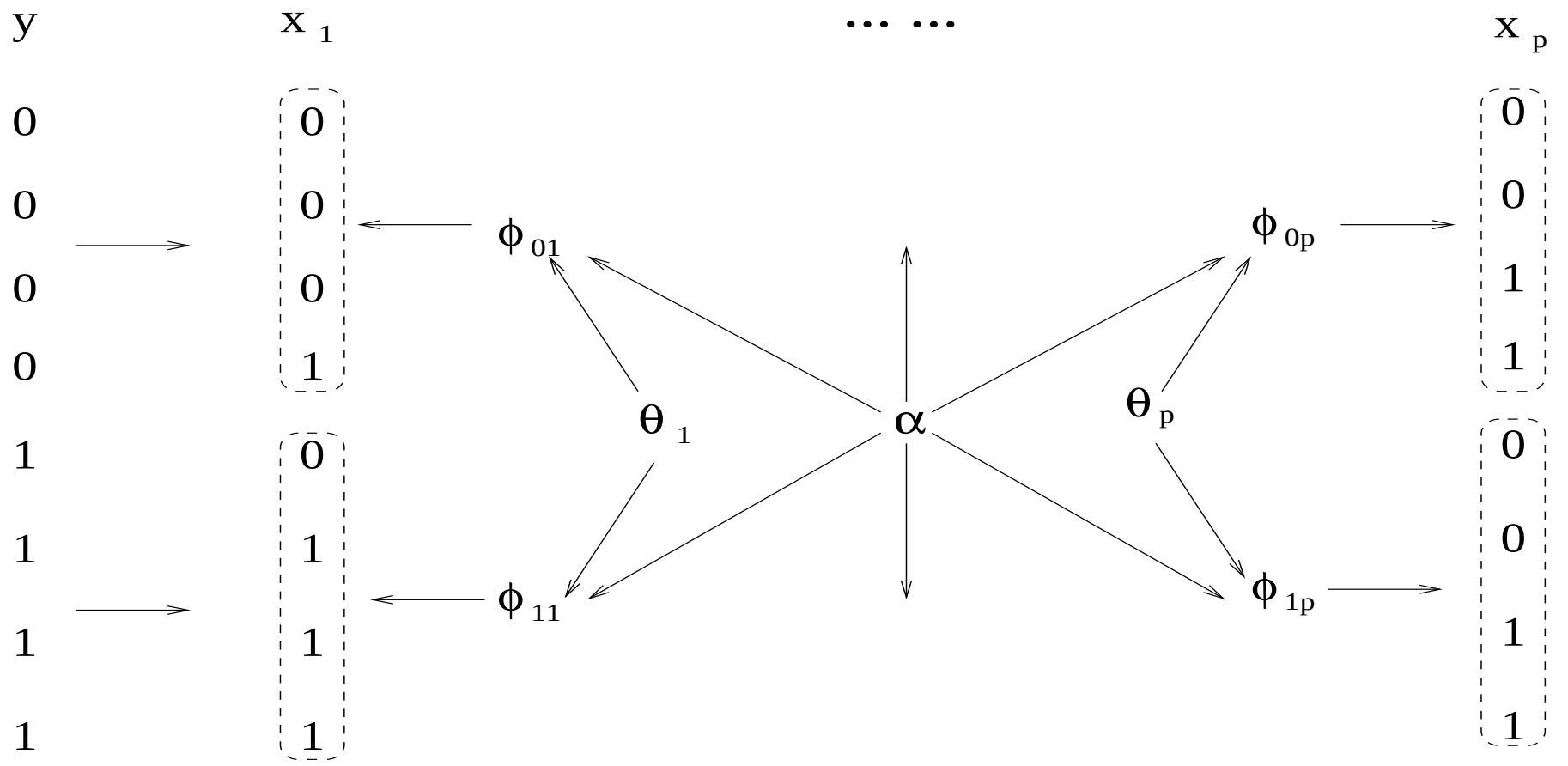
- Models: The response and the predictors are modeled jointly. Given the response y and a model parameter α, the features $x_1, \ldots, x_p$, are modeled to be independent and has identical distribution:

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

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

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

A Bayesian Naive Bayes Model for Binary Data



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

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

Computation of the adjustment factor (I)

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 $I_1 = \sum_{i=1}^{n} I(y^{(i)} = 1,\, x_t^{(i)} = 1)$:

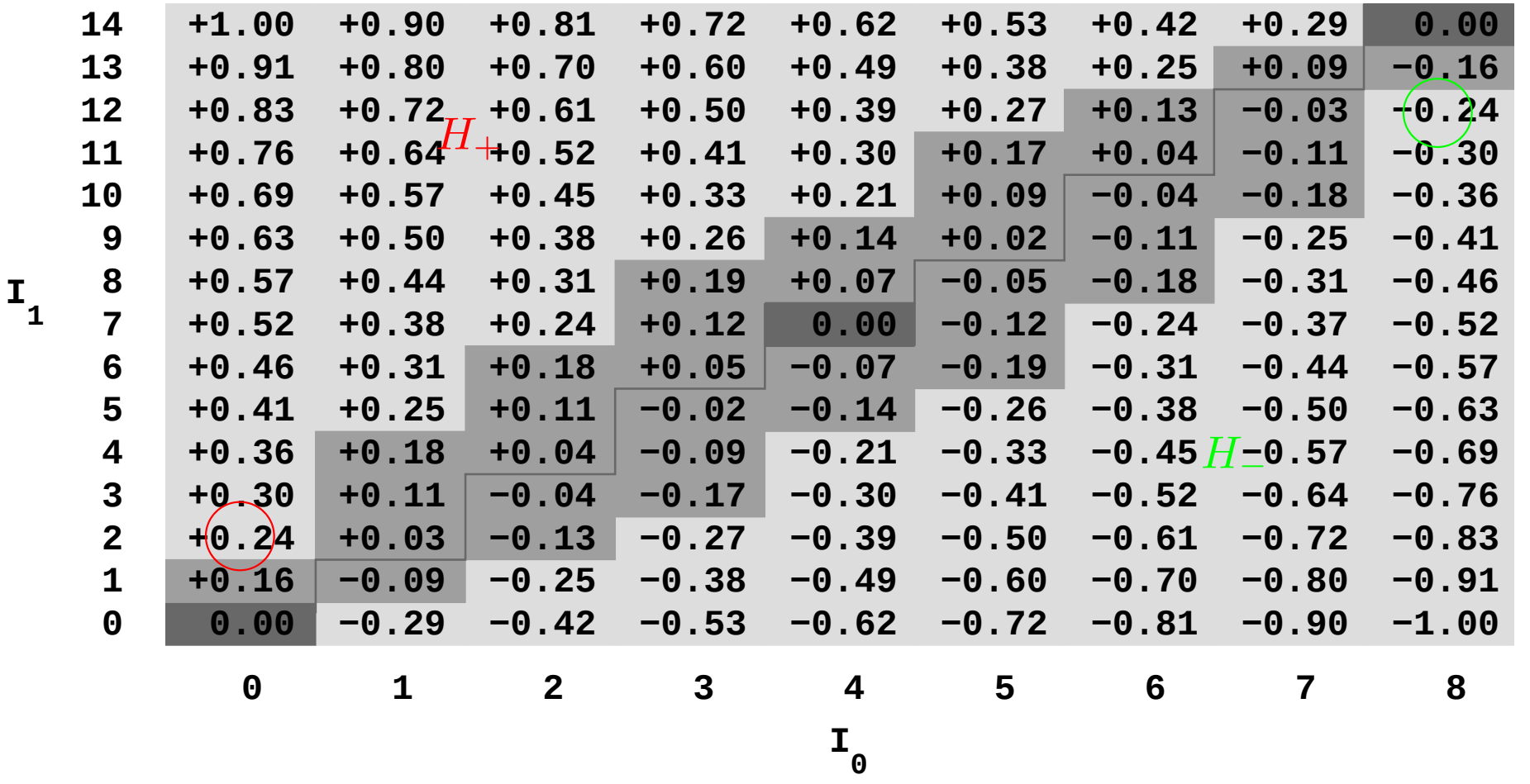
$$\text{COR}(x_t^{\text{train}},\, y^{\text{train}}) \;=\; \frac{(0-\overline{y})\, I_0 \;+\; (1-\overline{y})\, I_1}{\sqrt{n\overline{y}(1-\overline{y})}\,\sqrt{I_0 + I_1 - (I_0+I_1)^2/n}}$$

I_0,I_1 are visualized:

y^{train}	:	0	0	0	0	1	1	1	1
x_t^{train}	:	0	1	1	1	0	0	0	1
		O_0		I_0		O_1		I_1	

Computation of the adjustment factor (II)

The following graph displays the values of $|\text{COR}(x_t^{\text{train}}, y^{\text{train}})|$ in terms of I_0 and I_1 for a particular example, with the threshold of magnitude of correlation in selecting features is $\gamma = 0.2$:



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:

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

- $|\text{COR}(x_t^{\text{train}}, y^{\text{train}})|$ 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:

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

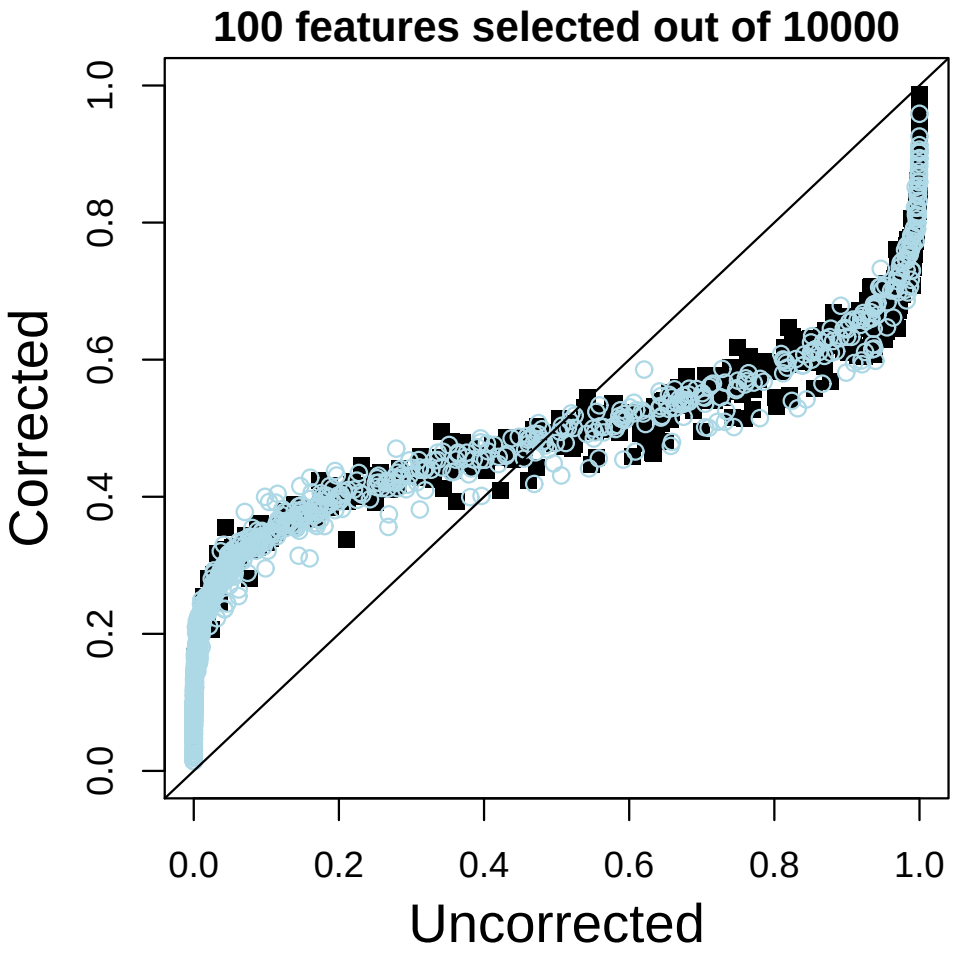
and

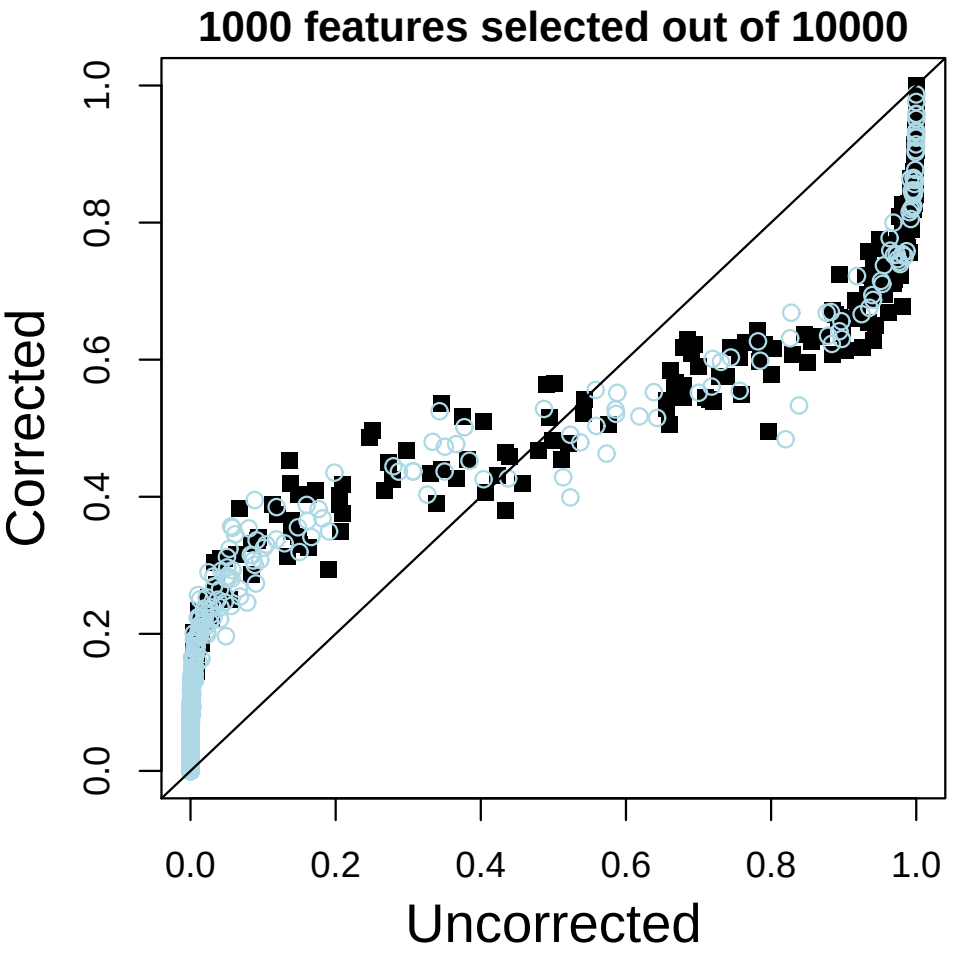
$$P(I_0 \mid \alpha, \theta_t, y^{\text{train}}) = \binom{n-N_1}{I_0} U(\alpha\theta_t, \alpha(1-\theta_t), I_0, n - N_1 - I_0)$$

A Simulation Experiment

- Generating data: $\alpha = 300$, $p = 10,000$, 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: $f_0 = f_1 = 1$, $a = 0.5$, $b = 5$

Plot of Predictive Probabilities



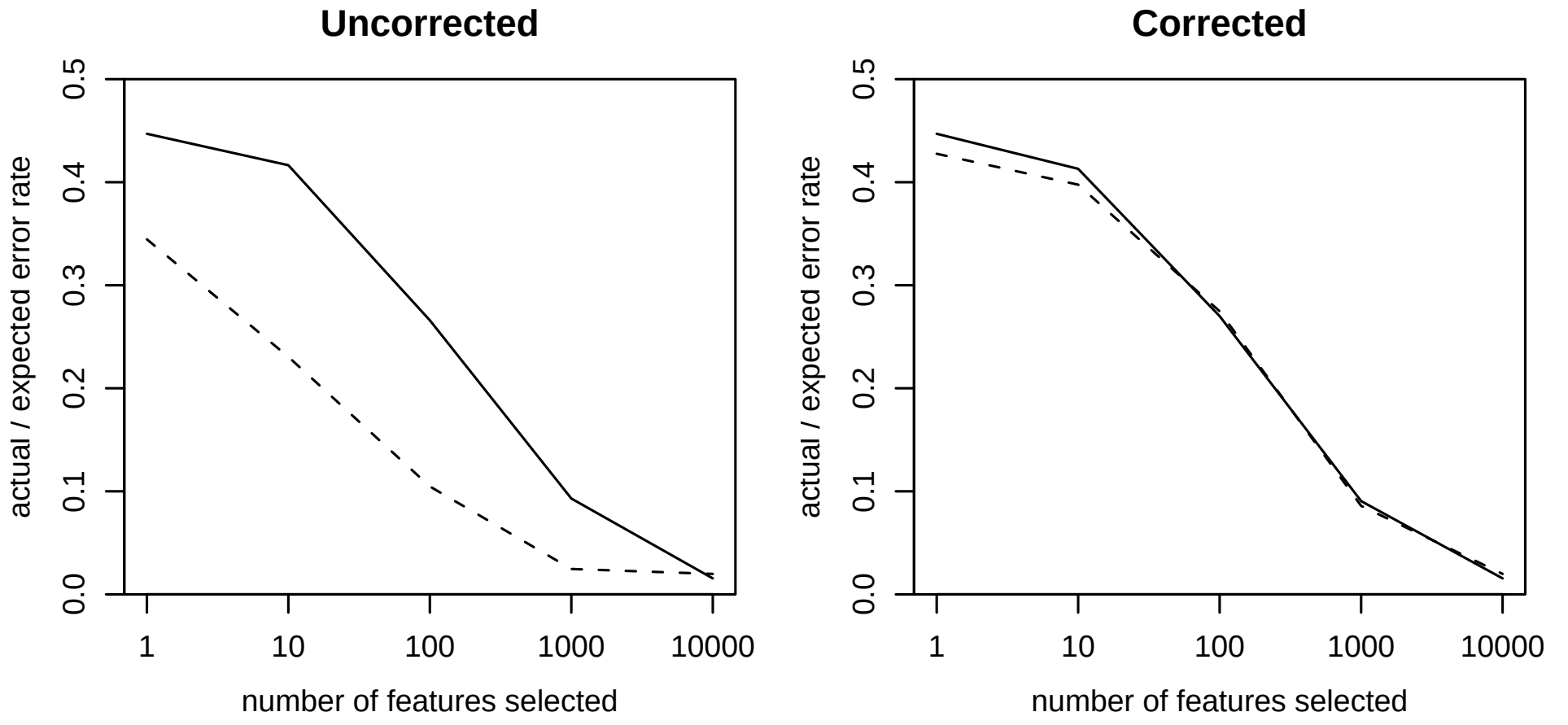


○: Class 0 ■: Class 1

Calibration of Predictive Probabilities

	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

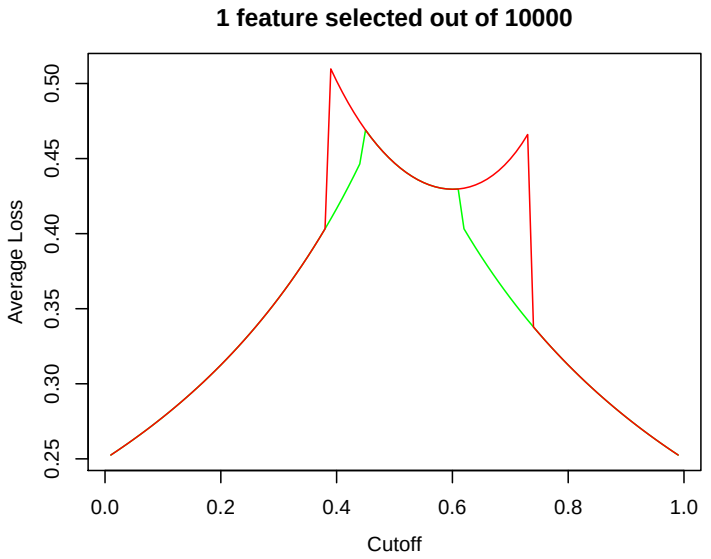
Expected versus Actual Error Rates

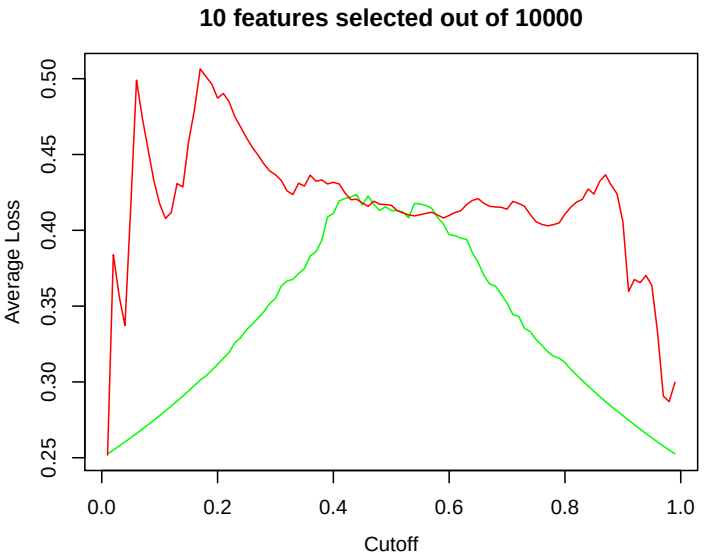


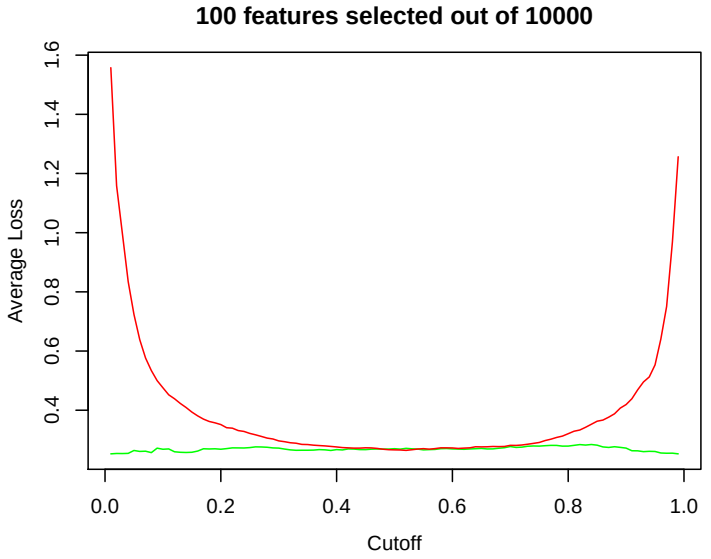
- - - - - : Expected Error Rate = $\frac{1}{n}\sum_{i=1}^{n}\left(\hat{p}^{(i)}I(\hat{p}^{(i)}<\frac{1}{2})+(1-\hat{p}^{(i)})I(\hat{p}^{(i)}\geq\frac{1}{2})\right)$

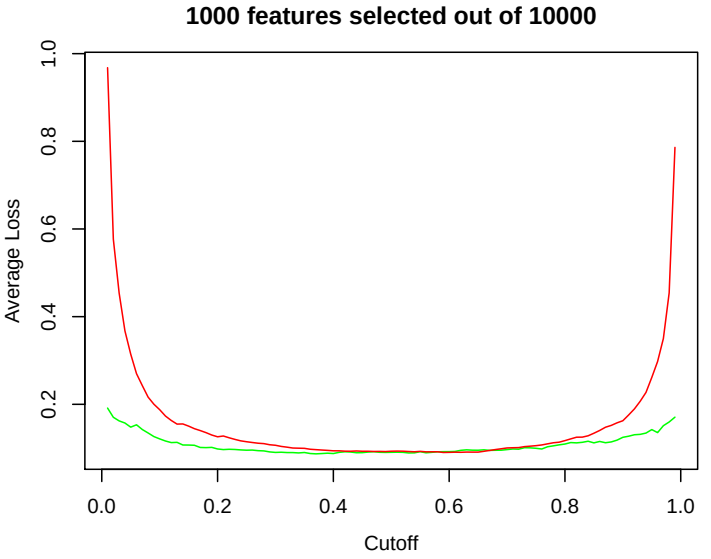
——— : Actual Error Rate = $\frac{1}{n}\sum_{i=1}^{n}I(y\neq\hat{y}^{(i)})$

Comparison of Average Loss









$$L_{0\to 1} = \frac{1+r}{2}$$

$$L_{1\to 0} = \frac{1+r}{2r}$$

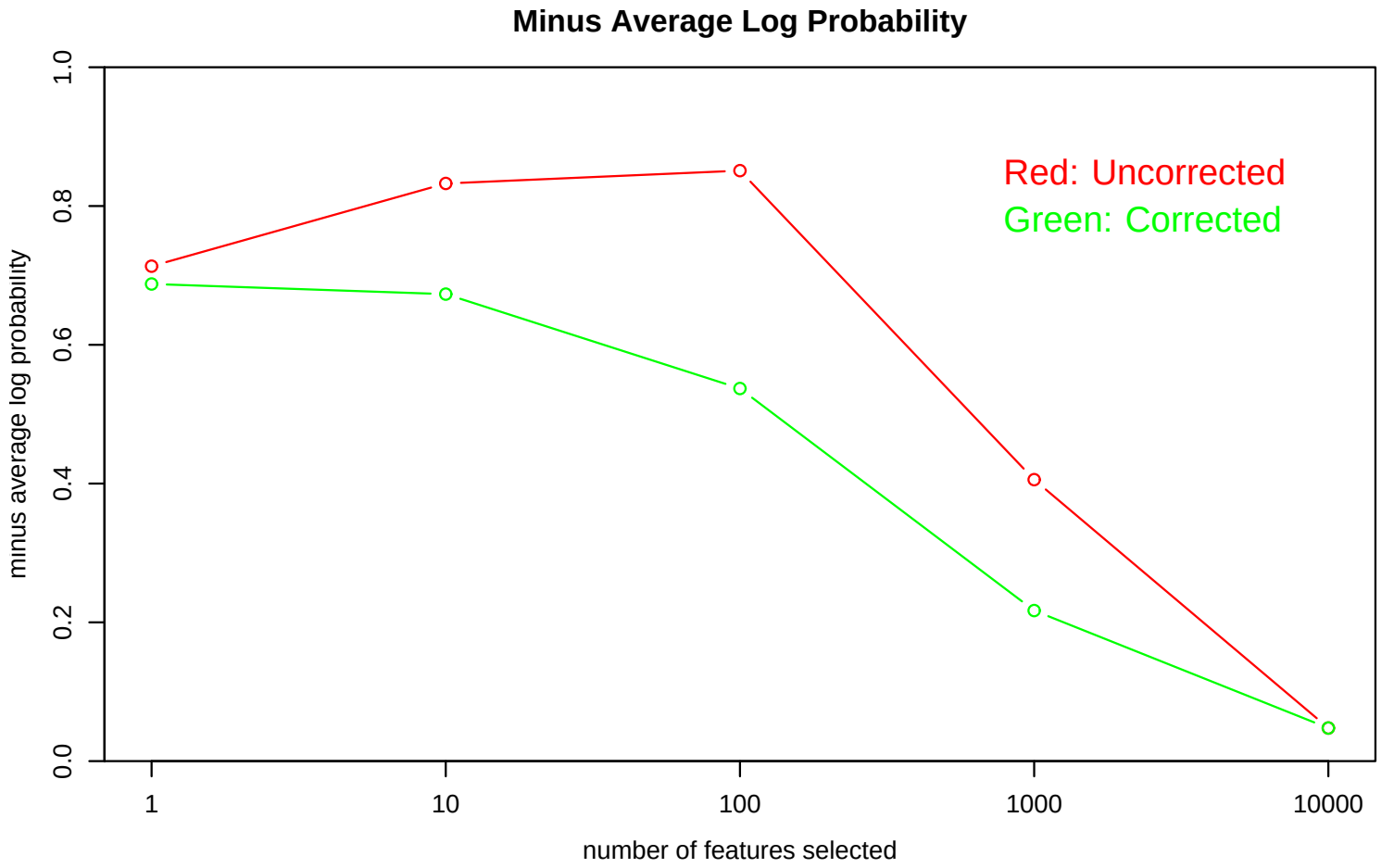
$$\text{cutoff} = 1-\frac{1}{1+r}$$

Red: Uncorrected

Green: Corrected

Comparison of Average Minus the Log Probabilities

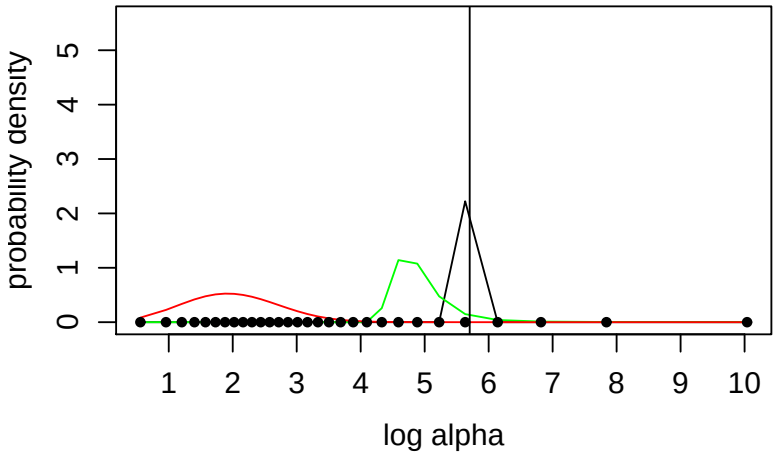
$$\text{AMLP} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}\log(\hat{p}^{(i)}) + (1-y^{(i)})\log(1-\hat{p}^{(i)})\right)$$



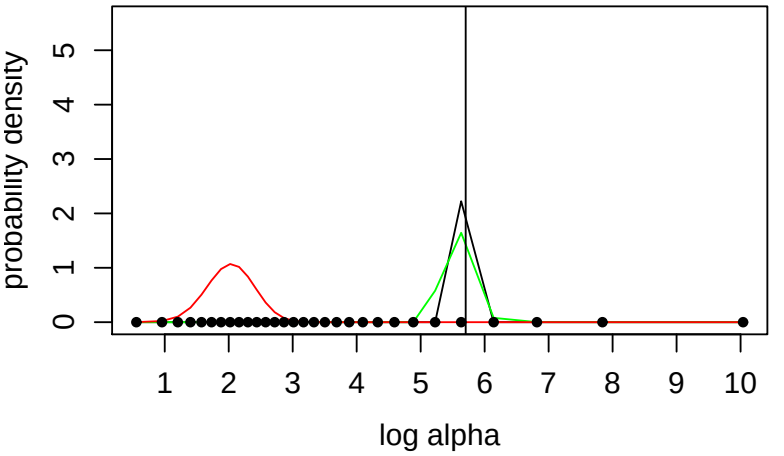
Posterior Distribution of $\log(\alpha)$

Red: Uncorrected Green: Corrected Black: Based on all features

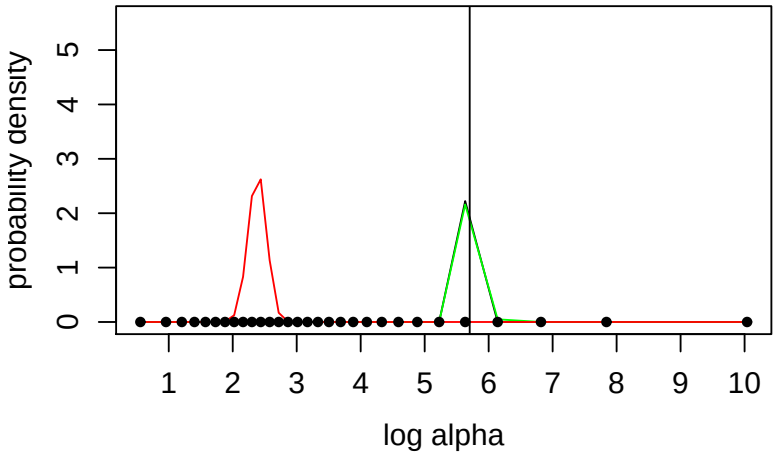
1 feature selected out of 10000



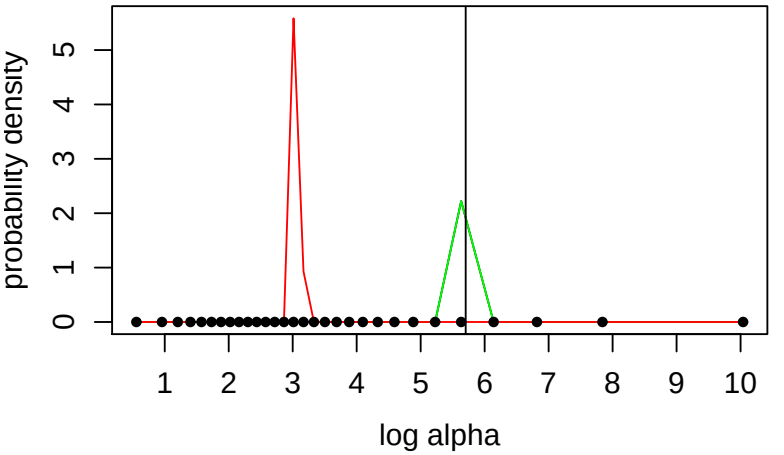
10 features selected out of 10000



100 features selected out of 10000



1000 features selected out of 10000



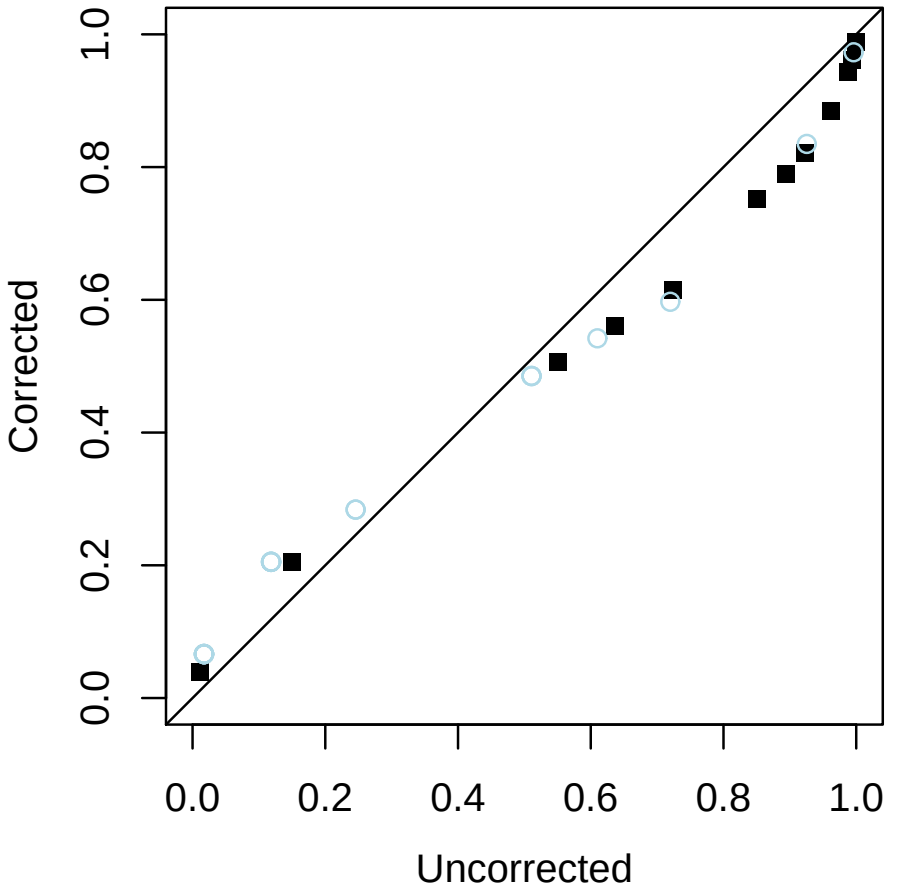
Computation Times

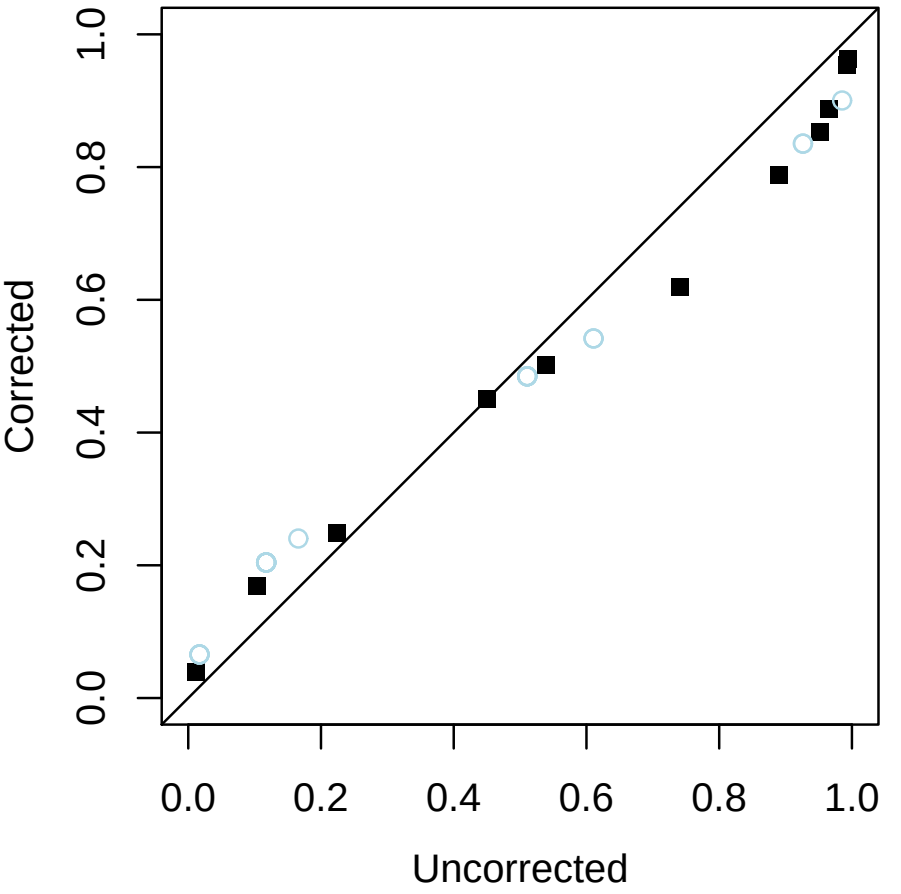
# of Features Selected	1	10	100	1000	10000
Uncorrected	12.66	28.80	203.60	2065.73	20627.71
Corrected	12.72	29.55	204.54	2076.21	

A Test with Gene Expression Data

- Data set: 2000 genes, 62 cases (40 Cancerous vs 22 Normal tissues)
- Converted into binary data by thresholding at medians of features
- Split 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

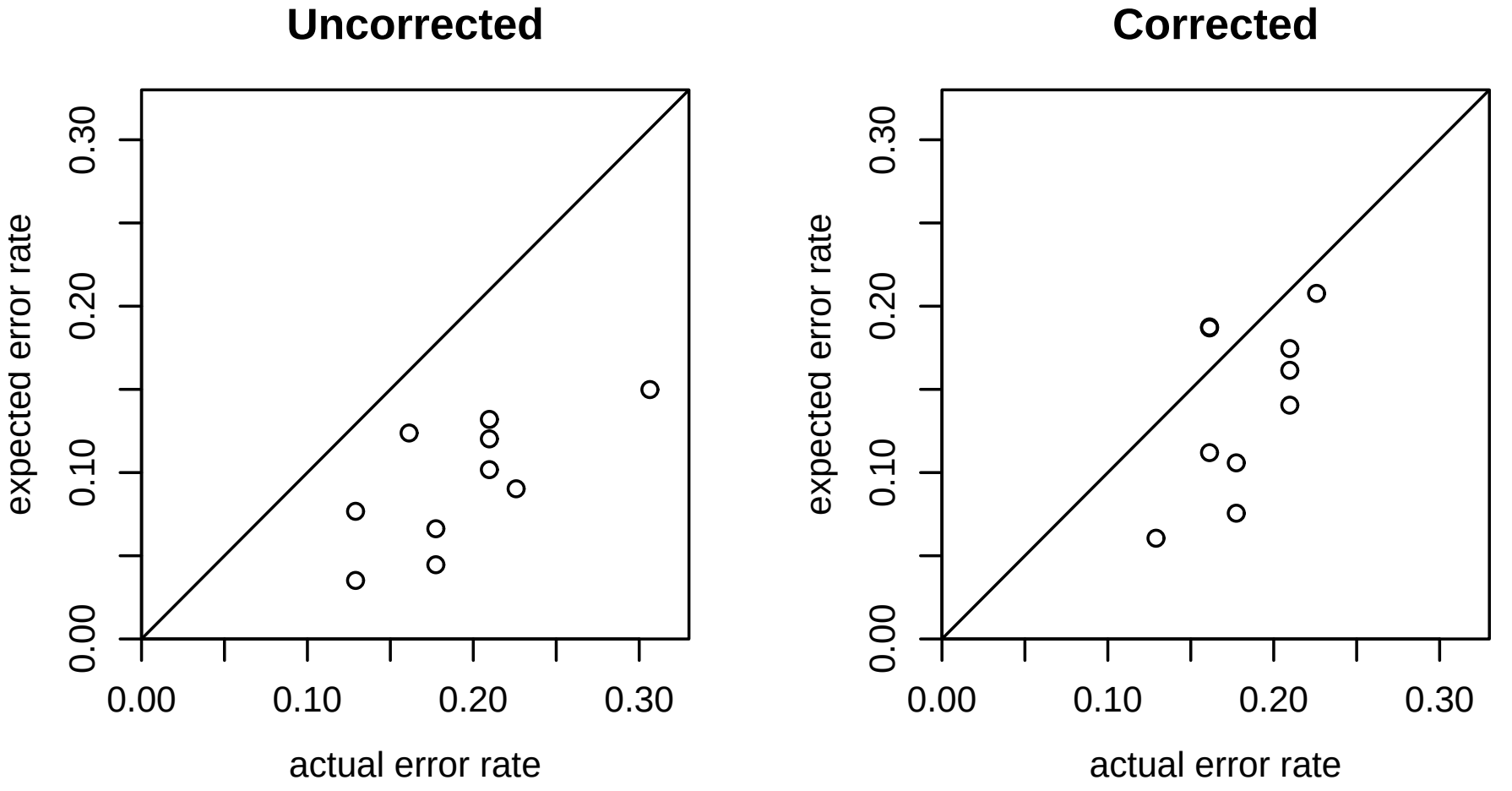
Plot of Predictive Probabilities



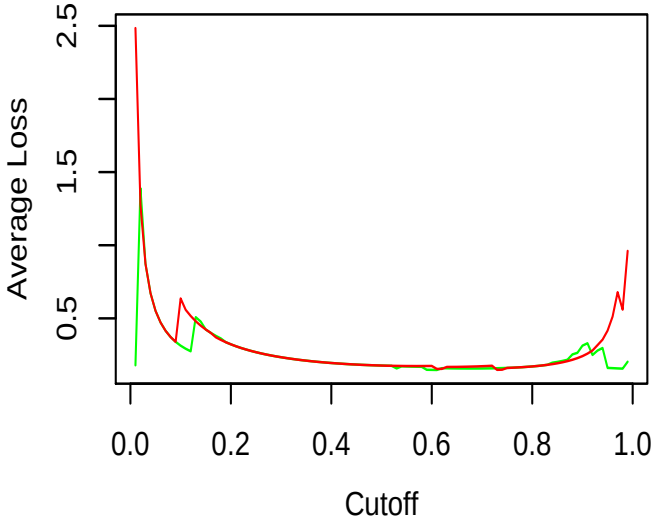


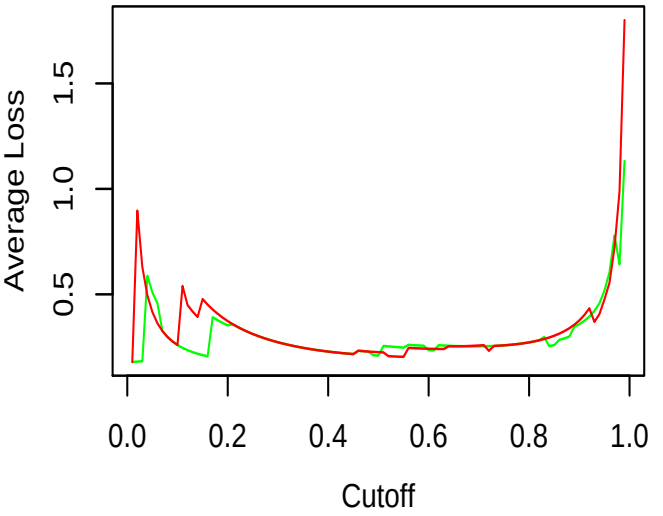
○: Class 0 ■: Class 1

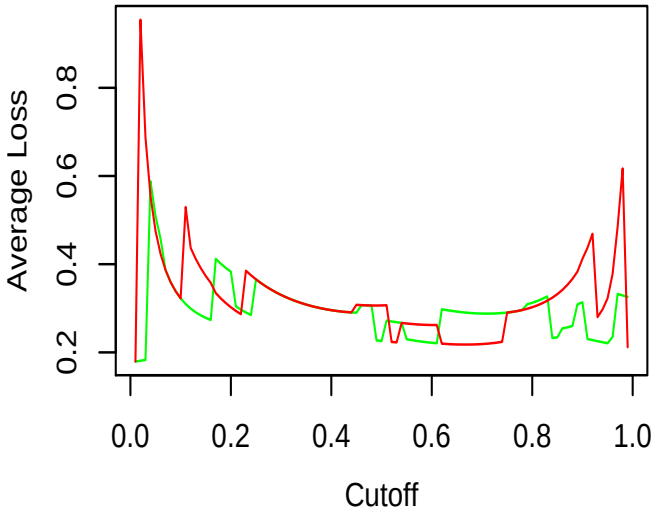
Expected versus Actual Error Rates

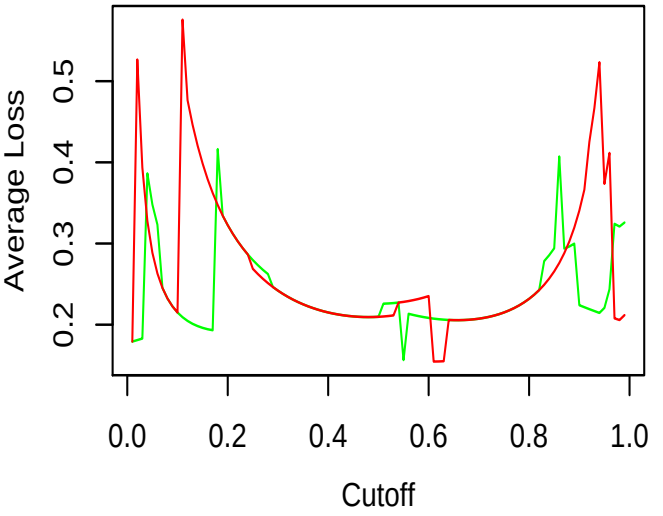


Comparison of Average Loss









$$L_{0\to 1} = \frac{1+r}{2}$$

$$L_{1\to 0} = \frac{1+r}{2r}$$

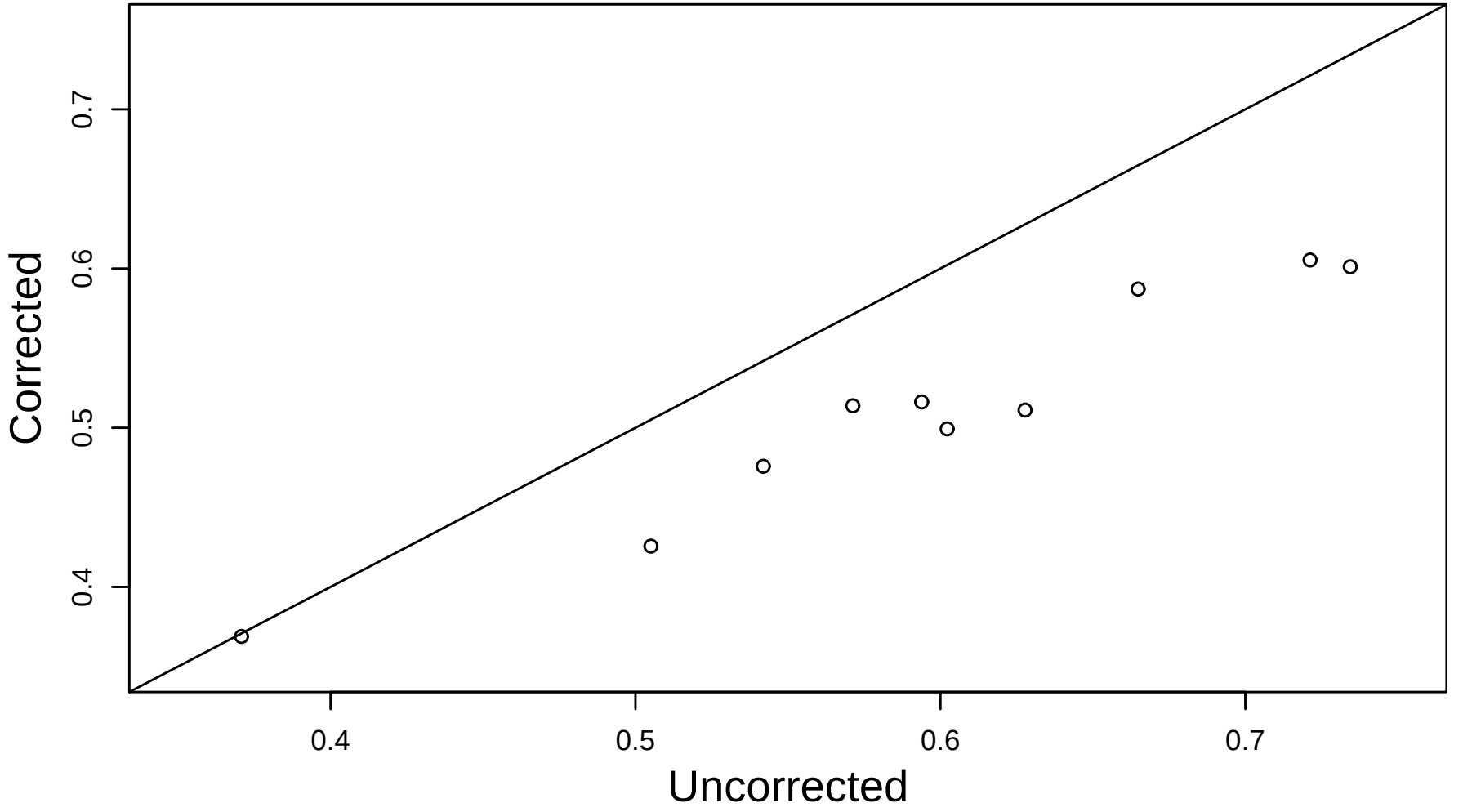
$$\text{cutoff} = 1 - \frac{1}{1+r}$$

Red: Uncorrected

Green: Corrected

Comparison of Average Minus the Log Probabilities

$$\text{AMLP} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}\log(\hat{p}^{(i)}) + (1-y^{(i)})\log(1-\hat{p}^{(i)})\right)$$



Conclusions and Discussions

- We propose a correction method to avoid the bias from feature selection.

Conclusions and Discussions

- We propose 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.

Conclusions and Discussions

- We propose 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.
- We have devised a fast method for computing it in binary naive Bayes models. This method can be generalized to all discrete naive Bayes models. However, the computation of the adjustment factor is difficult generally.

Conclusions and Discussions

- We propose 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.
- We have devised a fast method for computing it in binary naive Bayes models. This method can be generalized to all discrete naive Bayes models. However, the computation of the adjustment factor is difficult generally.
- When there is enough data, we better use different subsets to select features and fit the models, if we do not know how to correct for the feature selection bias.

Conclusions and Discussions

- We propose 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.
- We have devised a fast method for computing it in binary naive Bayes models. This method can be generalized to all discrete naive Bayes models. However, the computation of the adjustment factor is difficult generally.
- When there is enough data, we better use different subsets to select features and fit the models, if we do not know how to correct for the feature selection bias.
- More generally, if we do not know how to correct for the feature selection bias, we better avoid doing feature selection.

Future Work

- For real data sets, such as gene expression data, the independence assumption may not be true. We must apply the bias correction method to the independent factors governing the observed data. Some simple methods, such as principal component analysis, may be appropriate. Further work will be done to find a good way to do this.

Future Work

- For real data sets, such as gene expression data, the independence assumption may not be true. We must apply the bias correction method to the independent factors governing the observed data. Some simple methods, such as principal component analysis, may be appropriate. Further work will be done to find a good way to do this.
- One could pretty easily extend the method to other simple models (e.g. Gaussian models for continuous data) and other selection criteria (e.g. t statistic).

Future Work

- For real data sets, such as gene expression data, the independence assumption may not be true. We must apply the bias correction method to the independent factors governing the observed data. Some simple methods, such as principal component analysis, may be appropriate. Further work will be done to find a good way to do this.
- One could pretty easily extend the method to other simple models (e.g. Gaussian models for continuous data) and other selection criteria (e.g. t statistic).
- 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. Further work is necessary to improve efficiency of computing adjustment factor for such models, since we need to perform a large amount of this computation (e.g. in each iteration of Markov chain sampling). Some numerical methods, such as using importance sampling framework, are very promising.

Thank You! Questions and Comments?

References:

[1] Li, L., Zhang, J., and Neal, R. M. (2008), A Method for Avoiding Bias from Feature Selection with Application to Naive Bayes Classification Models, *Bayesian Analysis*, volume 3, number 1, pp 171-196.

[2] Li, L., (2007), *Bayesian Classification and Regression with High Dimensional Features*, Ph.D. thesis, University of Toronto.