Avoiding bias from feature selection in regression and classification models

Longhai Li

Department of Statistics
University of Toronto

Joint work with Radford Neal and Jianguo Zhang

Joint Statistical Meeting, Seattle, 9 Aug 2006

The Bias from Feature Selection

- We want to predict a response y using predictors $x_1, \ldots, x_p$.
- When p is big we may have to select a much smaller number, say k, of features to use due to computational or other reasons.
- We may decide to keep only the features having high correlation with y.
- **However**, this may result in overconfident prediction on test data.

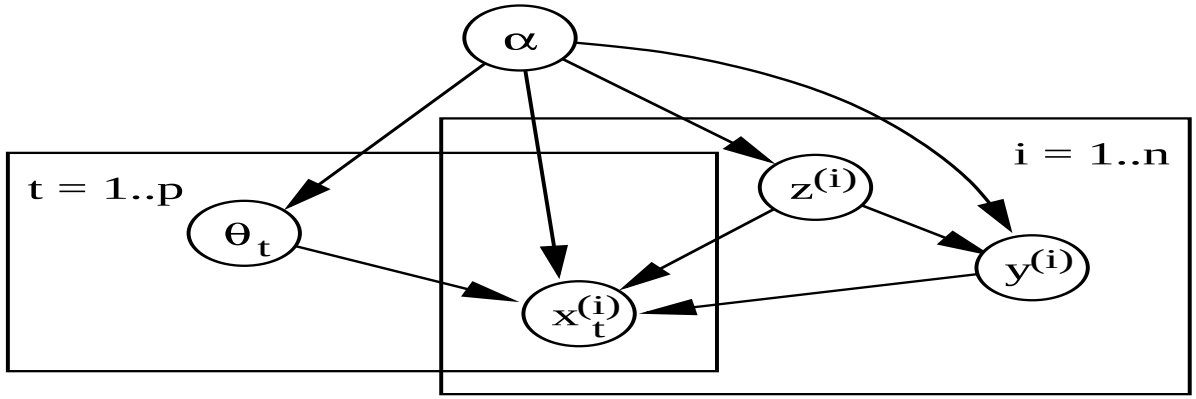
  As an extreme example, even when **all** the features are unrelated to y, there may seem to be strong correlations between y and the selected features.

Our approach to avoiding selection bias

- **General Idea**: Retain the information that $p - k$ features were discarded because of their weak correlation with the response.
- **Joint modeling**: The response and the predictors are modeled jointly. Given the response y, perhaps some latent values z, and model parameters α and $\theta_1, \ldots, \theta_p$, the predictors are modeled to be independent:

$$P(x_1, \cdots, x_p, y, z, \theta_1, \cdots, \theta_p, \alpha) = P(y, z, \alpha) \prod_{t=1}^{p} P(x_t \mid y, z, \theta_t, \alpha) P(\theta_t \mid \alpha)$$

The graphical model for independent cases, $i = 1, \ldots, n$:



Our approach to avoiding selection bias (continued)

- **Feature Selection**: Suppose we discard a feature x_t^{train} if it has sample correlation (COR) with y^{train} less than γ.
  Let $\mathcal{B}_\gamma = \{\tilde{x} : |\text{COR}(\tilde{x}, y^{\text{train}})| < \gamma\}$ be the set of all possible such features.

- **Modified Likelihood function**: The likelihood function of α, z should be based on the selected features $x_1^{\text{train}}, \ldots, x_k^{\text{train}}$ and the fact, $\mathcal{S}$, that $p-k$ features are in set $\mathcal{B}_\gamma$. If we don't use latent values, this is:

$$
\begin{aligned}
& L(\alpha; y^{\text{train}}, x_1^{\text{train}}, \cdots, x_k^{\text{train}}, \mathcal{S}) \\
&= \left[\prod_{t=1}^{k} P(x_t^{\text{train}} \mid y^{\text{train}}, \theta_t, \alpha) P(\theta_t \mid \alpha)\right] \cdot P(\mathcal{S} \mid \alpha, y^{\text{train}}) \\
&= \left[\prod_{t=1}^{k} P(x_t^{\text{train}} \mid y^{\text{train}}, \theta_t, \alpha) P(\theta_t \mid \alpha)\right] \cdot \left[P(x_{k+1}^{\text{train}} \in \mathcal{B}_\gamma \mid \alpha, y^{\text{train}})\right]^{p-k}
\end{aligned}
$$

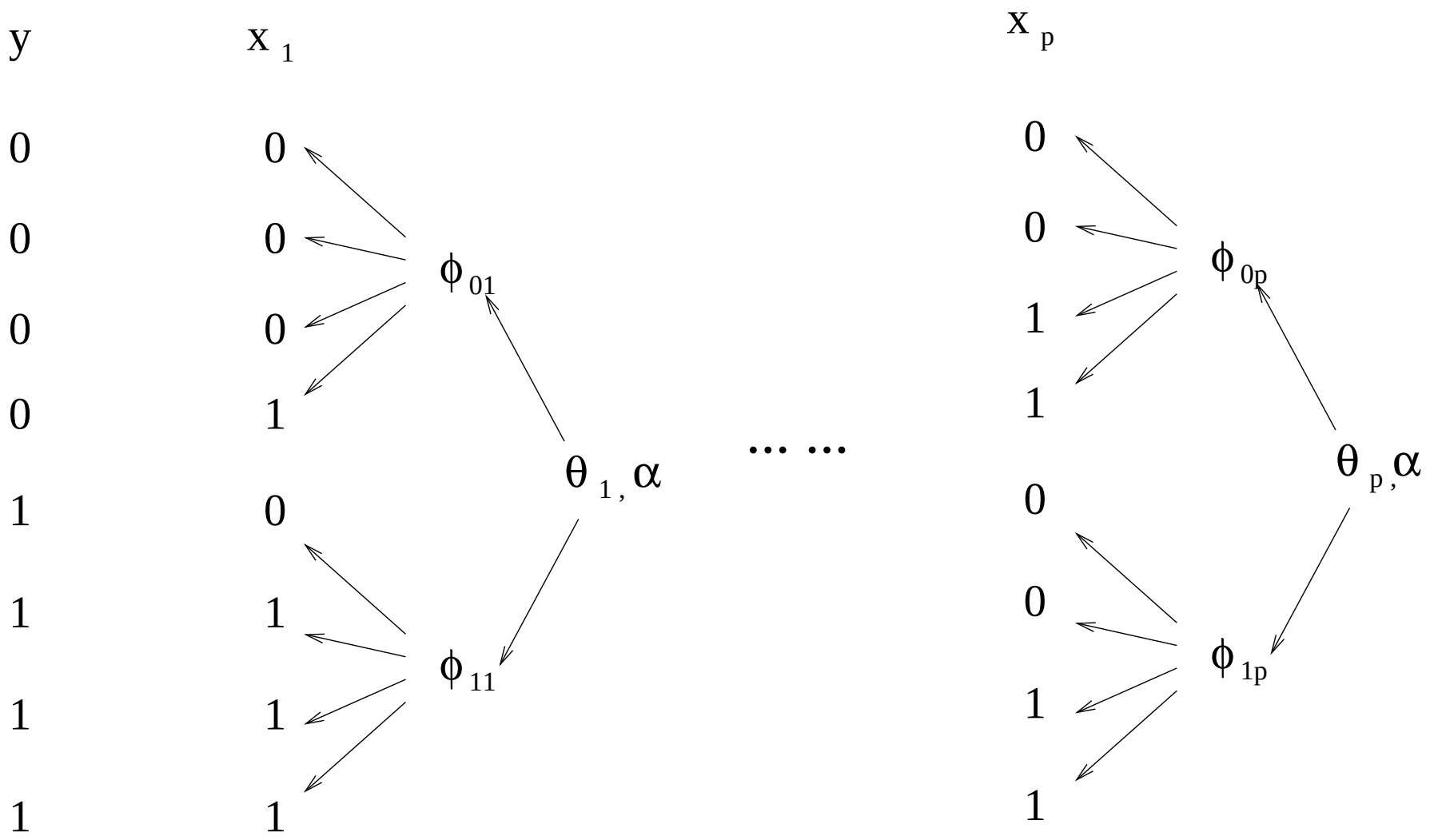
  Note that the probability in the last factor needs to be calculated only once regardless how many features are discarded, since $P(x_t^{\text{train}} \in \mathcal{B}_\gamma \mid \alpha, y^{\text{train}})$ is the same for all $t > k$.

Binary naive Bayes Models

$$
\begin{aligned}
\phi_y &\sim \text{Beta}(f_1, f_0) \\
y^{(1)}, \ldots, y^{(n)} \mid \phi_y &\overset{\text{IID}}{\sim} \text{Bernoulli}(\phi_y) \\
\alpha &\sim \text{Inverse-Gamma}(\alpha_0, \lambda_0) \\
\theta_1, \ldots, \theta_p &\overset{\text{IID}}{\sim} \text{Uniform}(a_0, b_0) \\
\phi_{0j}, \phi_{1j} \mid \alpha, \theta_j &\overset{\text{IID}}{\sim} \text{Beta}(\alpha\theta_j, \alpha(1-\theta_j)), \quad \text{for } j = 1, \ldots, p \\
x_j^{(i)} \mid y^{(i)}, \phi_{y^{(i)}} &\sim \text{Bernoulli}(\phi_{y^{(i)}j}), \quad \text{for } i = 1, \ldots, n
\end{aligned}
$$

Note that $\text{Var}(\phi_{yj}) = \theta_j(1-\theta_j)/\alpha + 1$. When α is bigger, the ϕ_{0j} and ϕ_{1j} are more likely to be close, and feature j is less likely to be correlated with the response y. Therefore α controls the overall degree of correlation between y and the features.

Picture of the binary naive Bayes models



Training the models with bias correction

- We use MCMC with α and $\theta_1, \ldots, \theta_k$ as the state
- For $j = 1, \cdots, k$, the posterior of θ_j given α is based only on x_j^{train} and the prior. ϕ_{0j} and ϕ_{1j} can be integrated away using the Polya urn scheme.
- The posterior of α is based on $x_1^{\text{train}}, \ldots, x_k^{\text{train}}$ as well as $\mathcal{S}$. Suppose we draw ϕ_{0j} and ϕ_{1j}, $j = 1, \ldots, k$ for temporary use, the posterior of α is:

$$
\begin{aligned}
& P(\alpha \mid \mathcal{S}, \phi_{0j}, \phi_{1j}, j = 1, \ldots, k) \\
= \quad & P(\alpha) \left[\prod_{j=1}^{k} \prod_{y=0}^{1} \text{beta}(\phi_{yj}; \alpha\theta_j, \alpha(1-\theta_j)) \right] \cdot [P(x_{k+1}^{\text{train}} \in \mathcal{B}_\gamma \mid \alpha, y^{\text{train}})]^{p-k}
\end{aligned}
$$

  where $\mathcal{B}_\gamma = \{\tilde{x} : |\text{COR}(\tilde{x}, y^{\text{train}})| < \gamma\}$.

- Note that the adjustment factor is an increasing function of α

Evaluation of the adjustment factor

- Evaluating the adjustment factor precisely is difficult. We need to sum the probability of feature x_{k+1}^{train} over all points in $\mathcal{B}_\gamma$ and integrate over θ_{k+1}:

$$\begin{aligned} P(x_{k+1}^{\text{train}} \in \mathcal{B}_\gamma \mid \alpha, y^{\text{train}}) &= \sum_{x_{k+1}^{\text{train}} \in \mathcal{B}_\gamma} P(x_{k+1}^{\text{train}} \mid \alpha, y^{\text{train}}) \\ &= \int \sum_{x_{k+1}^{\text{train}} \in \mathcal{B}_\gamma} P(x_{k+1}^{\text{train}} \mid \alpha, \theta_{k+1}, y^{\text{train}}) P(\theta_{k+1}) d\theta_{k+1} \end{aligned}$$

- For binary naive Bayes model, ways are available to do this faster:
  - $\mathcal{B}_\gamma$ is determined only by $n_{01} = \sum_{i=1}^{n} I(y^{(i)} = 0, x_{k+1}^{(i)} = 1)$ and $n_{11} = \sum_{i=1}^{n} I(y^{(i)} = 1, x_{k+1}^{(i)} = 1)$
  - $P(x_{k+1}^{\text{train}} \mid \alpha, \theta_{k+1}, y^{\text{train}})$ is determined also only by n_{01} and n_{11}.
- Since we can evaluate the integrand for a particular θ_{k+1}, then we can approximate the adjustment factor using Simpson's Rule.

A Simulation Study

- **Generating the data set**: with $\alpha = 400$, $a_0 = 0$, $b_0 = 1$, and $p = 10000$, we generated 350 cases with $y = 1$ and 350 cases with $y = 0$. From each group we chose 50 cases as the training data.

- **Training the model**: Priors used had $a_0 = 0, b_0 = 1$, $\alpha_0 = 5, \lambda_0 = 500$. 300 iterations of the Markov chain were run.

- **Feature selection**: Subsets of 1, 10, 100, 1000, and 2000 features were selected based on the absolute correlation with y^{train}.

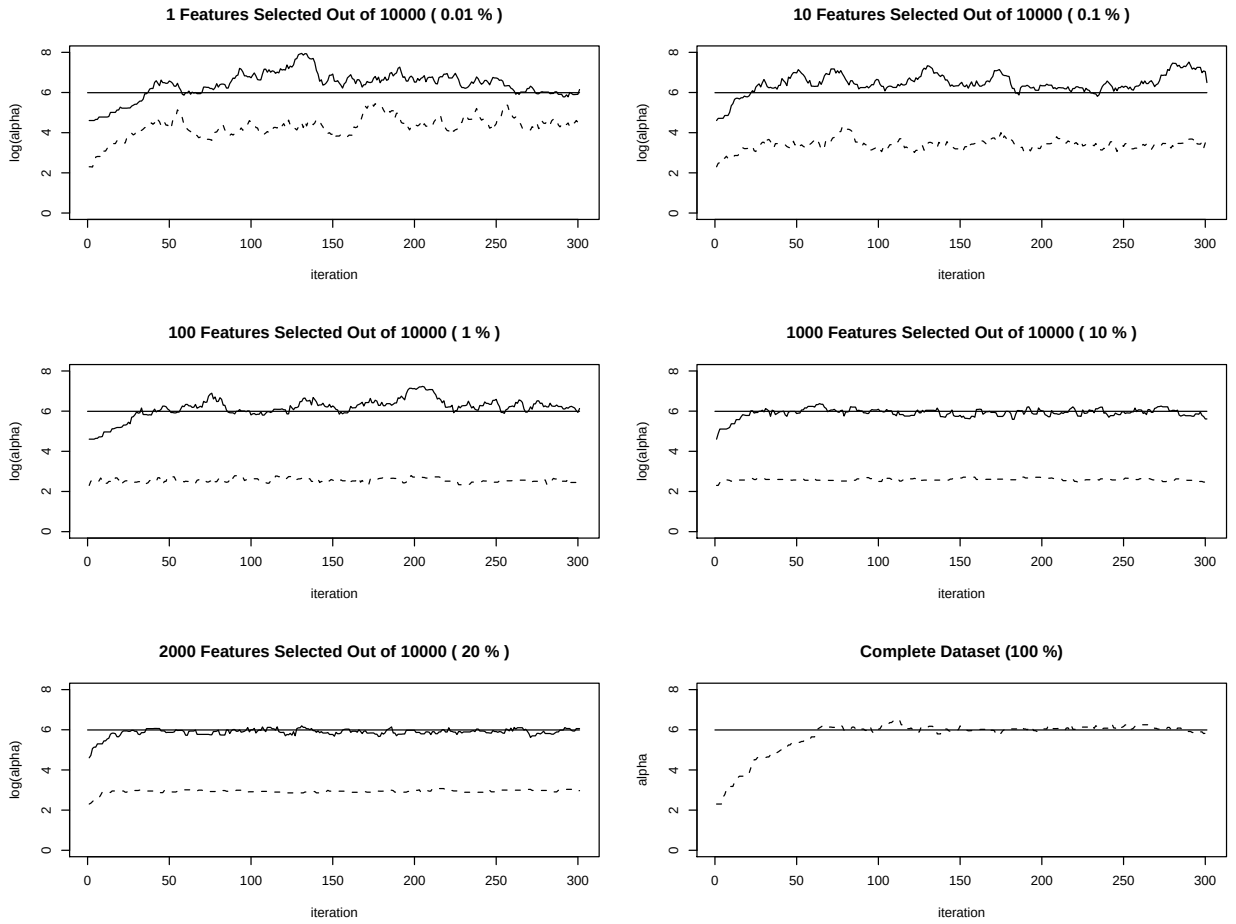
- **Assessment**: After obtaining the predictive probabilities for the 600 test cases, we categorized them by the first decimal of predictive probabilities into 10 groups. Within each group, we calculated the average predictive probability and compared it with the actual fraction of class 1.
  We also compared the posterior samples of α with the true value of 400.

Predictive probabilities for 600 test cases

	2000 Features Selected Out of 10000 (20%)						Complete Data Set		
	Uncorrected			Corrected					
Category	NO.	Pred.	Frac.1	NO.	Pred.	Frac.1	NO.	Pred.	Frac.1
0.0-0.1	276	0.005	0.16	125	0.04	0.09	187	0.03	0.03
0.1-0.2	14	0.14	0.36	61	0.15	0.10	46	0.15	0.26
0.2-0.3	2	0.26	0.00	44	0.24	0.27	39	0.24	0.23
0.3-0.4	10	0.35	0.80	39	0.35	0.28	17	0.34	0.35
0.4-0.5	3	0.45	1.00	31	0.46	0.52	15	0.44	0.20
0.5-0.6	5	0.52	0.60	35	0.55	0.63	27	0.56	0.63
0.6-0.7	7	0.65	0.43	37	0.65	0.59	21	0.66	0.81
0.7-0.8	5	0.75	0.60	42	0.74	0.86	26	0.76	0.85
0.8-0.9	11	0.85	0.64	68	0.86	0.79	55	0.85	0.91
0.9-1.0	267	0.99	0.84	118	0.96	0.93	167	0.97	0.95

The Markov chain traces of $\log(\alpha)$

Dashed line: Uncorrected method, Solid line: Corrected method, Horizonal line: the true value



Comparison of computational time (seconds)

Number of Features	1	10	100	1000	2000	10000
Fraction of Selection	0.01%	0.1%	1%	10%	20%	100%
Corrected	4597	5246	6515	12134	17473	
Uncorrected	5	48	479	4780	9544	48368

Note: these times are for an implementation in R.

Illustration on a real data set

- **Data sets**: Colon cancer data set, which has 62 cases and 2000 features. Divided into 10 smaller datasets, each with 200 randomly chosen features. We transformed the features to binary by thresholding at the medians. Five of the 200 features were selected based on the absolute correlation.

- **Assessment**: We used 10-fold cross-validation to compare the prediction performance of corrected and uncorrected methods. We compared the Mean Square Error (MSE) of the predictive probabilities from the true responses. The MSE values are shown by the following table:

Uncorrcted	.1324	.1351	.1655	.1763	.1525	.1696	.1583	.1432	.1681	.1667
Corrected	.1294	.1306	.1621	.1678	.1403	.1625	.1534	.1480	.1637	.1668
Difference	.0030	.0045	.0034	.0085	.0122	.0071	.0050	-.0048	.0044	-.0002

- The paired t-test on the two rows of MSE values gives a p-value of 0.017, showing that our bias correction method improves prediction.

Conclusion and Future Work

- 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 evaluation of the adjustment factor is difficult. For binary naive Bayes models, the method we used is fairly fast and precise. This method can be generalized to all discrete naive Bayes models.
- We've also applied the correction method to binary mixture models and factor analysis models. Evaluation of the adjustment factor for these models is more difficult but still feasible.
- Future work is needed to improve efficiency and extend the method to more models and other selection criteria.