The Journal of Prediction Markets 2012 Vol 7 No 1 pp 43-53

PREDICTING HORSE RACE WINNERS THROUGH REGULARIZED CONDITIONAL LOGISTIC REGRESSION WITH FRAILTY

Noah Silverman
Department of Statistics
University of California
Noahsilverman@ucla.edu

Marc A Suchard
Department of Biomathematics
Department of Biostatistics
Department of Human Genetics
University of California

ABSTRACT

Since first proposed by Bill Benter in 1994, the Conditional Logistic Regression has been an extremely popular tool for estimating the probability of horses winning a race. We propose a new prediction process that is composed of two innovations to the common CLR model and a unique goal for parameter tuning . First, we modify the likelihood function to include a “frailty" parameter borrowed from epidemiological use of the Cox Proportional Hazards model. Secondly, we use a LASSO penalty on the likelihood, where *profit* is the target to be maximized. (As opposed to the much more common goal of maximizing likelihood.) Finally, we implemented a Cyclical Coordinate Descent algorithm to fit our model in high-speed parallelized code that runs on a Graphics Processing Unit (GPU), allowing us to rapidly test many tuning parameter settings. Historical data from 3681 races in Hong Kong were collected and a 10-fold cross validation was used to find the optimal outcome. Simulated betting on a hold out set of 20% of races yielded a return on investment of 36.73%.

1 INTRODUCTION

Conditional logistic regression has remained a mainstay in predicting horse racing outcomes since the 1980's. In this paper, we propose and apply novel modifications of the regression model to include parameter regularization and a frailty contribution that exploits winning dividends. The model is trained using 4 years of horse racing data from Hong Kong, and then tested on a hold-out sample of races. Simulated betting produces a return on investment significantly higher than other published methods with Hong Kong races.

2 BACKGROUND

2.1 Horse Racing

Horse racing is one of the oldest forms of gambling in existence. It is conducted as a parimutuel style wagering contest. For a given race, each person bets on his choice to win the race. All monies bet are grouped into a pool. The racetrack takes a fixed percentage of the money pool in each race as its profit. The remaining funds are proportionally distributed amongst the bettors, who selected the winning horse, according to the amount they bet. A bettor's potential winnings, if a chosen horse wins, are erroneously named the “odds” on a horse. These “odds” do not represent the true odds or probability in the statistical sense, but are simply an indication of the percentage of bettors who favor the given horse. We prefer to think of these “odds” as *bettor implied confidence*.

This poses an interesting opportunity. Traditional casino games generally exist in a setting where the player bets against the establishment in a negative expectation game. Even with perfect skill, the player is bound to eventually lose his or her bankroll. With parimutuel betting, players are betting against each other, and it is possible to achieve a positive expectation bet if a winning horse can be chosen with more accuracy than the public. Many books and academic papers have been published in an attempt to model this system. [1] [5] [4] [8] [10]

2.2 Conditional Logistic Regression

Logistic regression is a form of a generalized linear model for working with binary outcomes [11]. In simple terms, logistic regression models the probability of an event occurring. The regression is linear in the log of the event odds. With the covariates X represented by an $N \times K$ dimensional matrix, with each row representing the covariates of a single outcome (Here a single horse's features.), β as a column of k regression coefficients, and p_i as the probability of a positive event outcome, the format of logistic regression may be represented as:

$$\ln\left(\frac{p_i}{1-p_i}\right) = x_i\beta \tag{1}$$

$$\left(\frac{p_i}{1-p_i}\right) = e^{x_i\beta} \tag{2}$$

The probability of an event occurring may be then be represented by the inverse logit transform

$$p = \frac{1}{1 + e^{-X\beta}} \tag{3}$$

Horse racing is an event where there is a single winner within a group of race competitors. A race $r = \{1,2,\dots,R\}$ is run between several horses $h = \{1,2,\dots,H\}$ with a single horse winning. The features of horse h in race r may be represented in a k dimensional vector identified by X_{rh}, The coefficients of the winning horse may then be represented by X_{rh}^{w}, with the superscript w indicating that horse h won race r. For each horse h in a race r, the estimated probability of winning that race is “conditioned” on the winning probabilities within race r summing to 1.

$$p_{rh} = \frac{e^{X_{rh}\beta}}{\sum_{h\in r} e^{X_{rh}\beta}} \tag{4}$$

The likelihood and log likelihood over all races then become:

$$\ell(\beta) \propto \prod_{r=1}^{R} \frac{e^{X_{rh}^{w}\beta}}{\sum_{h\in r} e^{X_{rh}\beta}} \tag{5}$$

$$\ln(\beta) \propto \sum_{r=1}^{R} \left[x_{rh}^{w}\beta - \ln(\sum_{h\in r} e^{X_{rh}\beta}) \right] \tag{6}$$

2.3 L_1 penalized conditional logistic regression

Tibshirani [14] proposed a method for variable selection for a Cox proportional hazards model through penalization of the regression coefficients under an L_1 norm. As the partial likelihood of the Cox model takes the similar form to the conditional logistic regression likelihood, and has the same partial likelihood. So we apply the same likelihood and shrinkage method.

Tibshirani describes penalizing the log likelihood, with a LASSO prior on β such that $\sum |\beta_j| \leq \lambda$ where λ is a user selected parameter. As λ is increased, the values of $beta$ are pushed toward zero. [15] By combining this shrinkage technique with cross validation, a model may be derived that maximizes likelihood while resisting over-fitting.

The log likelihood with this shrinkage factor is then:

$$\ln(\beta) \propto \sum_{r=1}^{R} \left[x_{rh}^{w}\beta - \ln(\sum_{h \in r} e^{x_{rh}\beta}) \right] - \lambda \sum_{k} |\beta_k| \quad (7)$$

3 CURRENT MODELS FOR PREDICTING HORSE RACING

Applying a conditional logistic regression to a horse race was first suggested by Boltman and Chapman [5] in 1986. Subsequently, it has been used, in a modified form by a majority of published work afterward. [2] [3] [10] [9] Each successive author has used a "two stage" form of this model. A "strength" factor, indicated by α, is calculated and then combined with the payoff dividend (generally in the form of a conditional logistic regression.) The likelihood of this conditional logit is:

$$\prod_{r=1}^{R} \frac{e^{\alpha_{rh}^{w}\beta_1 + p_{rh}^{w}\beta_2}}{\sum_{h \in r} e^{\alpha_{rh}\beta_1 + p_{rh}\beta_2}} \quad (8)$$

The concept here is to combine a previous model predicted "strength" of a horse, represented by $a = \{1,2, \dots, A\}$ with the "odds implied probability" for a horse $p = \{1,2, \dots, P\}$. Payoff dividends for a horse can be converted to implied probabilities with the formula: $p(x_h) = \frac{1}{d_h}$. Recall that this value is best described as the *bettor implied confidence*, as it is not a true probability of outcome, but an aggregate estimate of the public's opinion. Additionally, the "odds" reported at the racetrack represent the payoff with a track-take deducted. (This deduction is the profit of the racetrack hosting the event.) Subsequently, the dividend implied probability can be be calculated with the adjusted formula $p(x_h) = \frac{1-\tau}{d_h}$, with τ representing the track take from the parimutuel pool.

A series of authors have published successive variations to the original Boltman and Chapman model. Each publication has attempted to improve the calculation of the horse ``strength" measure that is represented by a in the conditional logistic regression equation (8).

- in 1994 Benter [2] used a conditional logistic regression to estimate horse strength.
- in 2006 Edelman [3] used a support vector regression on the horse's finishing position to estimate horse strength.

- in 2008 Lessmann and Sung [10] use a support vector classifier, and subsequent distance from the hyperplane to estimate horse strength.
- in 2010 Lessmann and Sung [9] use CART to estimate horse strength.

Despite different methods for calculating the "strength" of a horse, all of the authors used equation (8), a standard conditional logistic regression, as a second stage of their prediction algorithm.

4 APPLICATION OF A FRAILTY MODEL TO HORSE RACING

As noted previously, the conditional logistic regression likelihood shares similarities to a Cox Proportional Hazards model. The Cox model has been extended to account for the "frailty" of a subject, first suggested by Gillick [6] . This frailly was introduced to account for the fact that while a hazard may be proportionally equal for the population being studied, an individual may be more or less sensitive. This individual sensitivity is termed "frailty" and is represented here by ω. The likelihood involving a frailty term is then:

$$\ell(\beta) \propto \prod_{r=1}^{R} \frac{e^{X_{rh}^{w}\beta+\omega_{rh}^{w}}}{\sum_{h\in r} e^{X_{rh}\beta+\omega_{rh}}} \tag{9}$$

Our empirical analysis shows about a 40% correlation between public estimated probability and the true probability of a horse winning. In can be reasoned that the public is not completely naive, but has some degree of knowledge in picking winners. Subsequently, we propose using the public's knowledge to strengthen model accuracy. The "desirability" index w_{rh} for a horse h in race r may be calculated as a function of the posted "dividend" d_{rh} of a horse.

$$\omega_{rh} = 1 - (\frac{1}{d_{rh}}) \tag{10}$$

In essence, this can be considered a reverse-frailty model, as we are giving the horses with higher paying dividends a higher score. We are, in effect, creating a "strength" model instead of a frailty model. Profit from a race is maximally realized when betting on a horse that the public does not expect to win. Horses with the highest paying dividends, by definition, have the lowest public confidence. Looking for betting opportunities where the public is wrong is a strategy toward profit. By setting our strength(frailty) as the opposite of public confidence, we are looking for those horses we have a high confidence of winning that also pay high dividends. (Low public confidence of winning.)

The modified likelihood and log likelihood then become:

$$L(\beta) \propto \prod_{r=1}^{R} \frac{e^{X_{rh}^{w}\beta+\omega_{rh}^{w}}}{\sum_{h\in r} e^{X_{rh}\beta+\omega_{rh}}} \tag{11}$$

$$L(\beta) \propto \sum_{r=1}^{R} \left[X_{rh}^{w}\beta + \omega_{rh}^{w} - \log(\sum_{h\in r} e^{X_{rh}\beta+\omega_{rh}}) \right] \tag{12}$$

Our model includes 186 variable, many of which may be correlated. In an effort to avoid over-fitting, and and reduce the effect of colinearity, a regularization parameter is included. Applying the L1 shrinkage factor as described by Tibshirani gives a regularized log likelihood of:

$$L(\beta) \propto \sum_{r=1}^{R} \left[X_{rh}^{w}\beta + \omega_{rh} - \log(\sum_{h\in r} e^{X_{rh}\beta+\omega_{rh}}) \right] + \lambda \sum_{k=1}^{K} |\beta_j| \tag{13}$$

Using cyclical coordinate descent to fit this log likelihood requires the calculation of the first and second partial derivatives for each β_j

$$\frac{\partial L}{\partial \beta_j} = \sum_{r=1}^{R} x_{rh}^{w} - \frac{\sum_{h\in r} x_{rh} e^{x_{rh}\beta+\omega_{rh}}}{\sum_{h\in r} e^{x_{rh}\beta+\omega_{rh}}} + \lambda \tag{14}$$

$$\frac{\partial^2 L}{\partial \beta_j^2} = \sum_{r=1}^{R} \left[\frac{\sum_{h\in r} x_{rh}^2 e^{x_{rh}\beta+\omega_{rh}}}{\sum_{h\in r} e^{x_{rh}\beta} + \omega_{rh}} - \left(\frac{\sum_{h\in r} x_{rh} e^{x_{rh}\beta+\omega_{rh}}}{\sum_{h\in r} e^{x_{rh}\beta+\omega_{rh}}} \right)^2 \right] \tag{15}$$

5 CHOICE OF SHRINKAGE FACTOR λ

Ultimately, the goal of this process is not to predict winners with the most accuracy, but to maximize profit, which is not necessarily the same. This paper uses a frailty modified conditional logistic regression that is regularized by a shrinkage factor λ. While all of the coefficients may be calculated using the aforementioned cyclical coordinate descent, the choice of λ is left up to the user. We performed a 10-fold cross-validated training process, with a random 20% of the data held out for testing each time, for different values of λ, calculating the mean return-on-investment for the hold out set of each 10-fold run. λ is chosen to maximize ROI, which is defined as:

$$ROI = \frac{1}{10}\sum_{i=1}^{10}\left[\sum_{r}^{R} b_{rh}^{w} \cdot d_{rh}^{w} - \sum_{h\in r} b_{rh}\right] \tag{16}$$

with b_{rh}^{w} representing the amount bet on the winning horse h in race r, b_{rh} representing the amount bet on horse h in race r, and d_{rh}^{w} representing the dividend collected if horse h wins race r.

6 PARALLEL COMPUTING

Given the large data set of 40,000 cases of 186 variables each, computing the regularized logistic regression can be slow even on state of the art hardware. Additionally, adding the polynomial expansion magnifies the 186 variables to 36100. The optimal LASSO penalty, λ can't be calculated ahead of time, so must be discovered by iterating over a wide range of values. We performed a 10-fold cross validation for each of 100 different levels of λ resulting in 1,000 runs of the model fitting regression.

In order to speed up the model fitting process, and to allow for faster experimentation, custom software was written in C++. This software takes advantage of the Graphics Processing Unit (GPU) in the author's desktop computer. A basic level GPU costs about $250 and provides hundred of CPU cores optimized for parallel processing and are quickly finding many uses in statistics [16] [13]. Using the Thrust[7] library provided a fast and easy way to harness this computing power. (Specifics of the software used will be discussed in a forthcoming paper.)

Fitting this form of model involves the cyclical update of each coefficient many times. Each of those updates may be described as a series of transformations and subsequent reductions. For example, one component needed is the sum of the exponent of the linear term for each race:

$$\sum_{h\in r} e^{x_{rh}\beta+\omega_{rh}} \tag{17}$$

A non-parallelized solution would be to create a loop that computes the exponent of each item, then a second loop that sums these exponents by race. An intermediate solution would be to use the OpenMP[12] library to take advantage of running multiple threads on the same CPU. However this is still limited in terms of parallelization benefit. Parallelization on the GPU is fairly simple with a few strategic calls to the Thrust library. As the exponent steps are independent, they may all be performed in parallel. The subsequent sums by race are also independent, so they may also be computed in parallel.

7 RESULTS

Data from 3681 races in Hong Kong were collected from the Hong Jockey Club's official race records. 2944 races were used to learn the model through cyclical coordinate descent as described in algorithm 8. The data were 186 covariates describing each horse's past performance. (i.e. days of rest since last race, change in weight carried, average speed in previous races, and if the horse had gained or lost weight.) 737 races were withheld to use for testing the predictive strength of the model.

First, a two stage conditional logistic model, as described by Benter [2] was fit to establish a base level of performance. Bets were placed on any horse with a positive expectation, defined as: $Pr_{rh}(win) \cdot dividend_{rh} > 1.0$.

Return on investment (ROI) was calculated using equation (16). The Benter two-stage model produced a return on investment of 14.2%. Next, we fit our proposed modified frailty model with the same data. λ was varied between 1 and 100 to find the best resulting profit. With a λ of 7, our model produced a maximum ROI of 36.73% which is a significant improvement over the Benter model. As third version was tried, by adding the square of each covariate to the data set x^2 in an attempt to capture some non-linearity. However, the ROI produced was not as good as the main model. A plot of ROI as a function of λ for both models is included as plot 1

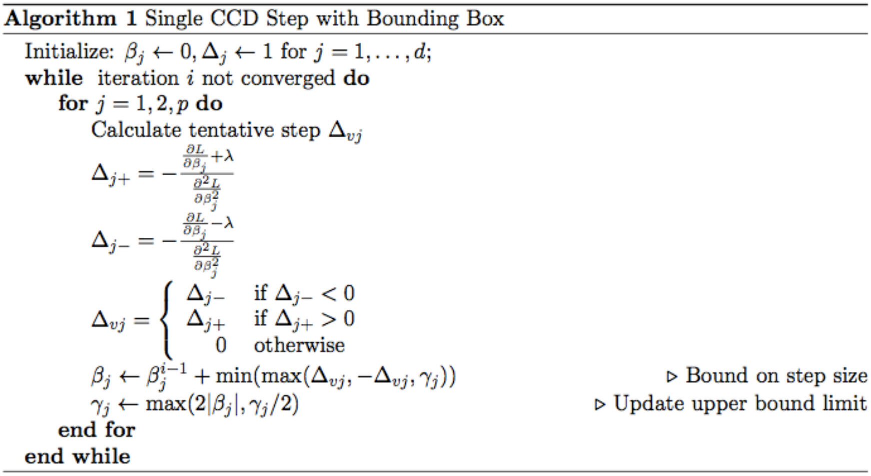
Parallelization of the model fitting algorithm via the GPU was compared to using the OpenMP library across a range of operating threads. The algorithm took 494 milliseconds to fit using the GPU solution. The openMP solution speed ranged from a high of 3995 milliseconds with a single thread to 1324 milliseconds using 8 threads. The GPU solution was faster than the best openMP solution by a factor of 2.68. A plot demonstrating the model fitting speeds is included as plot 2

8 CONCLUSION

The conditional logistic regression model has been a standard tool in horse race modeling. We have found that this continues to hold true. However, much of the past work revolves around how to best estimate a “strength" score that is fed into this form of model, and then maximizing accuracy probability estimates. By both combining a calculated inverse-frailty score, and changing the goal to directly maximizing profit, we are able to repeatedly generate a significantly higher return on investment.

Fitting a model with number of coefficients repeatedly as we iterate over a range of tuning parameters can be prohibitively slow. By using a CCD algorithm parallelized to take advantage of a GPU, we achieve a massive speedup, allowing us to quickly experiment with different forms of the model and tuning parameters.

Ultimately, our model innovations and tuning procedure produce a return on investment over 36% over repeated cross-validation trials.



Algorithm 1 Single CCD Step with Bounding Box

```
Initialize: $\beta_j \leftarrow 0, \Delta_j \leftarrow 1$ for $j = 1, \dots, d$;
while iteration $i$ not converged do
    for $j = 1, 2, p$ do
        Calculate tentative step $\Delta_{vj}$
        $\Delta_{j+} = -\frac{\frac{\partial L}{\partial \beta_j} + \lambda}{\frac{\partial^2 L}{\partial \beta_j^2}}$
        $\Delta_{j-} = -\frac{\frac{\partial L}{\partial \beta_j} - \lambda}{\frac{\partial^2 L}{\partial \beta_j^2}}$
        $\Delta_{vj} = \begin{cases} \Delta_{j-} & \text{if } \Delta_{j-} < 0 \\ \Delta_{j+} & \text{if } \Delta_{j+} > 0 \\ 0 & \text{otherwise} \end{cases}$
        $\beta_j \leftarrow \beta_j^{i-1} + \min(\max(\Delta_{vj}, -\Delta_{vj}, \gamma_j))$        ▷ Bound on step size
        $\gamma_j \leftarrow \max(2|\beta_j|, \gamma_j/2)$        ▷ Update upper bound limit
    end for
end while
```

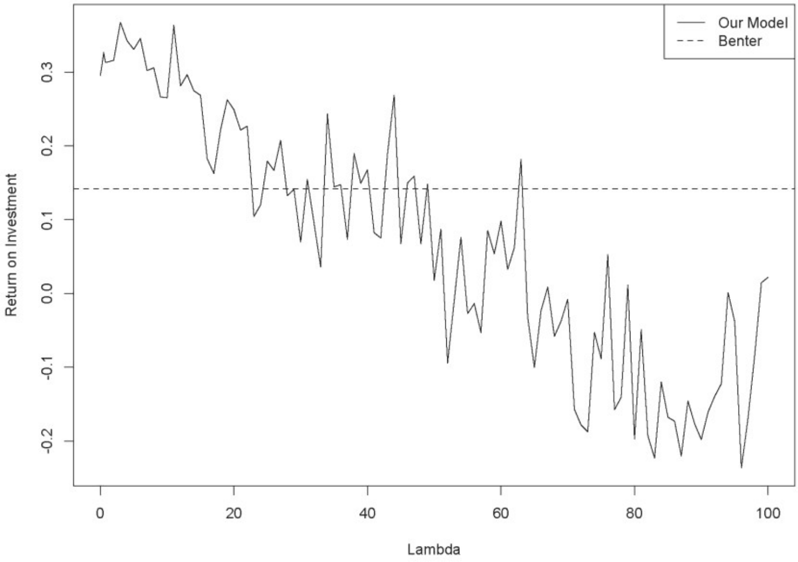


Figure 1: Mean ROI Over 10 Fold Cross Validation

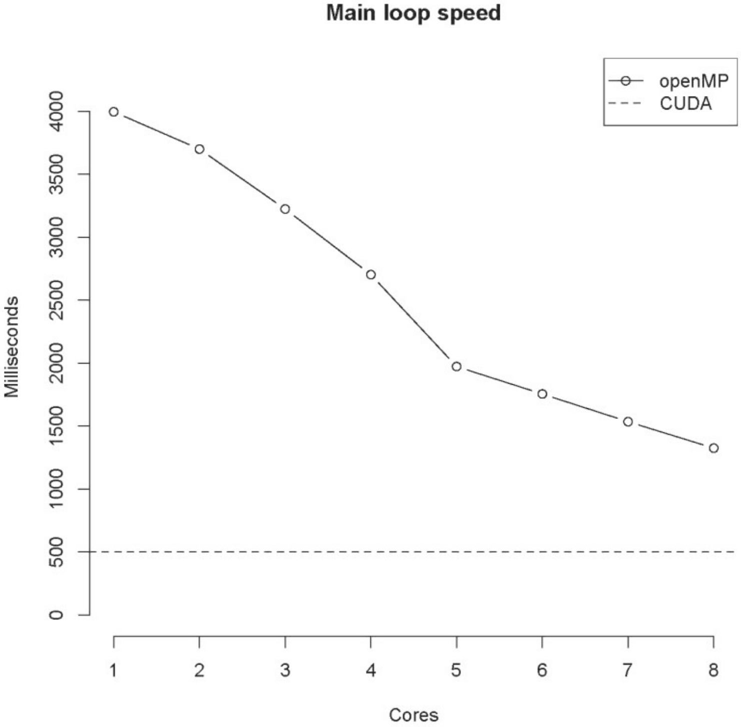


Figure 2: Speed of parallelization

9 REFERENCES

[1] Andrew Bayer. Bayer on Speed; New Strategies for Racetrack Betting. Mariner Books, 2007.

[2] Benter, William. Computer-Based Horse Race Handicapping and Wagering Systems: A Report. Efficiency of Racetrack Betting Markets, :183-198, 1994.

[3] Edelman, David. Adapting support vector machine methods for horserace odds prediction. *Annals of Operations Research*, 151:325-336, 2007. 10.1007/s10479-006-0131-7.

[4] Efficiency of Racetrack Betting Markets. World Scientific Publishing Company, 2008.

[5] Ruth N. Bolton and Randall G. CHapman. Searching for Positive Returns at the Track: A Multinomial Logit Model for Handicapping Horse Races. *Management Science,* 32(8):1040-1060, 1986.

[6] Gillick, Muriel. Guest Editorial: Pinning Down Frailty. The Journals of Gerontology Series A: *Biological Sciences and Medical Sciences*, 56(3):M134-M135, 2001.

[7] Hoberock, J. and Bell, N. Thrust: A parallel template library. Online at http://thrust. googlecode. com, 2010.

[8] Michael Kaplan. The High Tech Trifecta. *Wired Magazine*, 2003.

[9] Stefan Lessmann and Ming-Chien Sung and Johnnie E.V. Johnson. Alternative methods of predicting competitive events: An application in horserace betting markets. International *Journal of Forecasting*, 26(3):518 - 536, 2010.

<ce:title>Sports Forecasting</ce:title>.

[10] Stefan Lessmann and Ming-Chien Sung and Johnnie E.V. Johnson. Identifying winners of competitive events: A SVM-based classification model for horserace prediction. *European Journal of Operational Research*, 196(2):569 - 577, 2009.

[11] McCullagh, Peter and Nelder, J. A. Generalized Linear Models, Second Edition (Chapman & Hall/CRC Monographs on Statistics & Applied Probability). Chapman & Hall/CRC, London, United Kingdom, 2 edition, 1989.

[12] OpenMP Architecture Review Board. OpenMP Application Program Interface Version 3.1. 2011.

[13] Suchard, M.A. and Simpson, S.E. and Zorych, I. and Ryan, P. and Madigan, D. Massive parallelization of serial inference algorithms for a complex generalized linear model. arXiv preprint arXiv:1208.0945, 2012.

[14] Tibshirani, Robert. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1):267--288, 1996.

[15] Tibshirani, Robert. The Lasso Method for Variable Selection in the Cox Model. *Statist. Med.*, 16(4):385--395, 1997.

[16] Zhou, H. and Lange, K. and Suchard, M.A. Graphics processing units and high-dimensional optimization. Statistical science: a review journal of the Institute of *Mathematical Statistics*, 25(3):311, 2010.