Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

# Forecasting Ranking in Harness Racing Using Probabilities Induced by Expected Positions

Fredrik Armerin, Jonas Hallgren & Timo Koski

**To cite this article:** Fredrik Armerin, Jonas Hallgren & Timo Koski (2019) Forecasting Ranking in Harness Racing Using Probabilities Induced by Expected Positions, Applied Artificial Intelligence, 33:2, 171-189, DOI: 10.1080/08839514.2018.1536105

**To link to this article:** https://doi.org/10.1080/08839514.2018.1536105

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 2, 171–189
https://doi.org/10.1080/08839514.2018.1536105

Taylor & Francis
Taylor & Francis Group



# Forecasting Ranking in Harness Racing Using Probabilities Induced by Expected Positions

Fredrik Armerin, Jonas Hallgren, and Timo Koski

Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden

**ABSTRACT**

Ranked events are pivotal in many important AI-applications such as Question Answering and recommendations systems. This paper studies ranked events in the setting of harness racing.

For each horse there exists a probability distribution over its possible rankings. In the paper, it is shown that a set of expected positions (and more generally, higher moments) for the horses induces this probability distribution.

The main contribution of the paper is a method, which extracts this induced probability distribution from a set of expected positions. An algorithm is proposed where the extraction of the induced distribution is given by the estimated expectations. MATLAB code is provided for the methodology.

This approach gives freedom to model the horses in many different ways without the restrictions imposed by for instance logistic regression. To illustrate this point, we employ a neural network and ordinary ridge regression.

The method is applied to predicting the distribution of the finishing positions for horses in harness racing. It outperforms both multinomial logistic regression and the market odds.

The ease of use combined with fine results from the suggested approach constitutes a relevant addition to the increasingly important field of ranked events.

## Introduction

The problem of finding the outcome of a harness race is studied. Formally, that is the problem of obtaining a probability distribution from a set of ordered expectations.

A prediction of the outcome of a competitive event is sought. Predicting the outcome can, for instance, mean the winner but the approach here suggested predicts the full distribution of the outcome of the race. The method produces the probability that a certain horse finishes in a particular position.

**CONTACT** Jonas Hallgren jhallg@kth.se Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

The motivation for this methodology is that it allows more freedom for the user in specifying and building models.

Sometimes heuristics can be implemented in the model. Therefore, it can many times be easier to create a model for the positions rather than for the probabilities. As an example, it can be known that a horse always finishes before another horse. While mathematically different, this motivation is similar to that of Freund, Schapire, and Abe (1999) who introduced the AdaBoost algorithm, also with applications in horse racing.

### *Harness Racing and Efficient Markets*

Adelman (1981) gives the history of harness racing in, mainly, New York, and is the reference for this section. Harness racing, or trotting, has a long tradition dating back to at least the 19th century. Races were originally organized as impromptu contests at the beginning of the 19th century. The step to the modernization of harness racing is the formation in the Winter 1824–1825 of the New York Trotting Club. A trotter typically started twice as many races annually as a good thoroughbred, and this made it possible to commercialize the sport by increasing the number of races. By collecting statistics regarding trotters, it was possible to form opinions on how good a horse would perform in a race even though it had only competed for a few, if any, times at a given race track. Hence, from early on collecting statistics played an important part in harness racing.

When the algorithm presented in this paper is applied, data from harness races are used. One part of the data are the odds offered by a bookmaker. For the odds of an event to be useful, it must contain information about the probability of the event in question, and this leads to the concept of efficient markets. Efficiency in a financial market deals with how efficiently prices reflect different types of information. This, in turn, leads to the question of whether a market is efficient or not: The Efficient Market Hypothesis is a theory that the price of a security reflects all currently available information about its economic value. A market in which prices fully reflect all available information is said to be efficient. Elton et al. (2014). When considering betting markets, prices of assets are replaced with the odds of different events.

In betting markets, as in their financial equivalents, there are three forms of market efficiency: weak, semi-strong and strong. These forms of efficiency differ in the respect of information sets, i.e. what is included in the definition of "information", Elton et al. (2014).

Testing for different kinds of efficiency in betting markets goes back at least to the 1940s when both laboratory experiments and tests based on observed results from race tracks were used; see Vaughan Williams (2005). Both in laboratory experiments as well as in different types of actual betting markets, there is typically a tendency for bets on horses with lower odds to have a better return than bets put on horses with higher odds. This tendency

is often referred to as the "favorite-longshot bias", or just "longshot bias". There are several suggested explanations for this anomaly; see e.g. Coleman (2004), Thaler (1992), Vaughan Williams (2005) and references therein.

Although identifying the longshot-favorite bias together with inconsistent pricing in the show and place markets as two anomalies in the pari-mutual betting markets, Thaler (1992) concludes: The racetrack betting market is surprisingly efficient. Market odds are remarkably good estimates of winning probabilities.

## Competitive Events

There are $n$ horses participating in a race, and their expected finishing positions are given. What can then be said about the induced probability distribution? This paper proposes a method where the distribution is given as the solution to a convex optimization problem. A two-step procedure emerges: the expected positions are estimated and then the distribution is obtained by solving a convex optimization problem.

Ranking of Competitive events is a well-studied field with many recent contributions. There are many aspects of the problem but predicting the winner is most common in the literature. Ordered expectations are studied in for instance Gaines and Rice (1990) but there the interest is testing when expectations are ordered.

Recent approaches to predicting the winner use popular machine learning tools. Lessmann, Sung, and Johnson (2009) applies an SVM-based classification model indicating a presence of non-linear relationships among the variables. Another method is crowdsourcing used by Schumaker (2013) where several sources, such as multiple bookmakers, are used to create the prediction; this combined with a betting system produced good performance. The performance of different neural networks is evaluated by Davoodi and Khanteymoori (2010). The experimental results are similar to the ones presented in this paper. Silverman and Suchard (2013) use a regularized logistic regression. This paper also employs regularized regression but without the limitations of logistic regression.

To evaluate individual players performance relative to other opponents it is also possible to use a ranking system, Aldous (2015). These systems have been of mathematical interest for more than a century, Carroll (1883), but rose to fame more recently with the Elo-system, Elo (1978), developed for rating chess players. Glickman (1999) generalized Elo from a Bayesian perspective which was further developed to Microsofts' Trueskill in Herbrich, Minka, and Graepel (2006). All three systems relate to Bradley and Terry (1952).

Pieramati et al. (2010) develops an Elo-rating for trotters which would fit well into our framework. The ratings could be used as expectations producing a probability distribution.

### Ranking in AI

The ranking problem is important and well studied in AI-applications. A prominent example is DeepQA, featured in IBMs Watson, Ferrucci et al. (2010). Other examples are Ko, Si, and Nyberg (2010). In a similar field but with a different application Breese, Heckerman, and Kadie (1998) use a ranking approach for recommendation engines in e-commerce applications. Interesting future work would be to apply the methodology to these problems.

### Convex Optimization

Familiarity with a few popular concepts in convex optimization is assumed throughout this paper. CVX is a package for specifying and solving convex programs, see Grant and Boyd (2014), Grant and Boyd (2008). Another method, the alternating direction method of multipliers (ADMM), is also implemented, see Boyd et al. (2011) for the version used in the paper. See Fukushima (1992), Gabay and Mercier (1976), and Glowinski and Marroco (1975) for historical references, and Nishihara et al. (2015) for recent theory on ADMM. ADMM provides high performance and is—given the work by Boyd et al. (2011)—easy to implement for many standard problems. While our problem is one of the standard problems, the ADMM implementation does require more work than its CVX counterpart.

### Contributions

The main contribution of this paper is the two-step procedure, an algorithm which yields a probability distribution from a set of expected values in a ranked competitive event. The problem of finding the distribution is expressed as a convex optimization problem. Estimates of the expectations are required as input to solve the problem.

The expectations are obtained without restriction. The method is a relevant competitor to logistic regression which is considered a standard method. The suggested approach supersedes logistic regression both in performance and in speed.

The paper is organized as follows. Section 2 provides a mathematical formulation of the problem and present our solution as a two-step procedure. In Section 3 the developed method is applied to an application in regression analysis. The methodology is applied to real data in Section 4. The paper concludes with a discussion in Section 5. The dataset is described in Appendix A; a few theoretical justifications are given in Appendix B. MATLAB code is provided in Appendix C.

## The Induced Distribution

This section introduces necessary terminology and background. Then the main problem is formulated and solved.

### *Problem Formulation*

Throughout the paper it shall be assumed that the participants in the race are horses. Let $n$ be the number of horses and let $X_k$ be the finishing position for horse number $k$. Denote its expected position, $\mathbb{E}[X_k]$, by $\mu_k$.

Denote the probability that horse number $k$ will finish in the $j$'th position by $p_{kj}$. The element $p_{kj}$ of the matrix $P$ denotes the probability that horse number $k$ finishes in position $j$. That is, the $k$'th row of $p$ gives the distribution of the finishing position for the $k$'th horse while the $j$'th column gives the distribution for the $j$'th position. Let $\mu$ denote the vector $[\mu_1 \mu_2 \cdots \mu_n]$.

A distribution must sum to 1; that combined with the properties of expectation gives the following set of equations.

*i)* $\sum_{j=1}^{n} j p_{kj} = \mu_k$ for every $k$.

*ii)* $\sum_{k=1}^{n} p_{kj} = 1$ for every $j$.

*iii)* $\sum_{j=1}^{n} p_{kj} = 1$ for every $k$.

A matrix $P$ satisfying 2. and 3. is called a *doubly stochastic* matrix. Finding $P$ corresponds to solving the matrix equation

$$XP = m^{\mathrm{T}} \stackrel{\Delta}{=} [\mu \quad 1 \quad 1 \quad \cdots \quad 1]^{\mathrm{T}}$$

with the constraint [*i)*]

*i)* $0 \leq p_{kj} \leq 1$

for every $k$ and every $j$. Here $X$ is defined to satisfy equations *i-iii* as,

$$X = \begin{bmatrix} A \\ B \\ C \end{bmatrix}.$$

The capital letter matrices $A, B$ and $C$ correspond to equations *1, 2* and *3*, respectively; they are defined as direct sums of their lower case vectors:

$$A = \begin{bmatrix} \mathbf{a} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{a} \end{bmatrix}, B = \begin{bmatrix} \mathbf{b} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{b} \end{bmatrix}, C = \begin{bmatrix} I & \cdots & I \end{bmatrix}.$$

The vectors $\mathbf{a} \overset{\Delta}{=} [1\,2 \cdots n]$, and $\mathbf{b} \overset{\Delta}{=} [1\,1 \cdots 1]$ are both of length $n$. The matrix $C$ is $n$ copies of an $n \times n$ identity matrix. So the matrices $A$, $B$, $C$ are all of size $n \times n^2$ and the matrix $X$ is of size $3n \times n^2$

Thus, the problem is written as

$$\begin{array}{ll} \underset{P}{\text{minimize}} & XP - m \\ \text{subject to} & 0 \leq p_{kj} \leq 1,\ \forall j, k. \end{array} \tag{1}$$

The class containing the solutions to the problem, the $n \times n$ doubly stochastic matrices, is called the $n$th Birkhoff polytope. It is convex and has dimension $(n-1)^2$, see Pak (2000), and Beck and Pixton (2003). The polytope was named after Birkhoff (1946) and famously used by Von Neumann (1953).

If the number of participating horses is large there might exist several solutions. The set of equations can then be extended to include higher moments. That is,

$$m^{\mathrm{T}} = [\mu\ \mu^2 \ldots \mu^r\, 1 \ldots 1]^{\mathrm{T}}$$

with a corresponding $X$ matrix with an extended $A$. Let $A_k$ denote the matrix where $\mathbf{a}_k = [1^k, 2^k, \ldots, n^k]$. Then $A$ is given by the stacked matrices $[A_1 A_2, \ldots A_r]^{\mathrm{T}}$ and is of size $r \times n$. The size of the extended $X$ matrix is of size $(2+r)n \times n^2$. Thus, it is always possible to extend the problem and find a solution given higher moments. Proposition B.3 establishes that the solution is unique.

The optimization problem is convex since both the objective function and the feasible region—the Birkhoff polytope—are convex, Luenberger (1997). A more general formulation of the optimization problem would be to replace the norm with a loss function but this paper is restricted to work with the norm.

The upper bound of the constraint can be relaxed since any positive $P$ satisfying $XP = m$ will be a probability distribution. However, in Section 4 the real world example indicates that this relaxation impairs performance.

### Normalizing the Expectation

Typically the expectations are not known but estimated as $\hat{\mu}$. Obtaining the probability distribution from the estimates can be problematic. The relation

$$\begin{aligned} \sum_{k=1}^{n} \mu_k &= \sum_{k=1}^{n} \mathbb{E}[X_k] = \sum_{k=1}^{n} \sum_{j=1}^{n} j p_{kj} \\ &= \sum_{j=1}^{n} j \sum_{k=1}^{n} p_{kj} = \sum_{j=1}^{n} j = \tfrac{1}{2} n(n+1), \end{aligned} \tag{2}$$

must hold and inserting the estimate $\hat{\mu}$ does not necessarily satisfy the equality. To remedy this, consider the normalized estimate $\tilde{\mu}$ given by

$$\tilde{\mu}_k = \hat{\mu}_k \frac{1}{2} \frac{n(n+1)}{\sum_j^n \hat{\mu}_j}, \tag{3}$$

which satisfies

$$\sum_{k=1}^{n} \tilde{\mu}_k = \frac{1}{2} \frac{n(n+1)}{\sum_j^n \hat{\mu}_j} \sum_{k=1}^{n} \hat{\mu}_k = \frac{1}{2} n(n+1). \tag{4}$$

All elements in $\mu$ must lie in the interval $[1, n]$. That is, require that $1 \leq \tilde{\mu}_k \leq n$ for every $k$. If this condition is not met, the problem is not well specified and a probability distribution can not be found, see Proposition B.2. It is still possible to solve the optimization problem and find an approximative solution but it will not be a probability distribution.

### Examples

Given a vector

$$\hat{\mu} = [1.4 \quad 2.2 \quad 3.1]$$

the induced distribution is sought. To do this, normalize to find $\tilde{\mu}$ and then proceed by solving the convex optimization problem; see Figure 4 in Appendix C for Matlab code. This gives the estimated $\hat{P}$-matrix

$$\hat{P} = \begin{bmatrix} 0.88 & 0.10 & 0.02 \\ 0.10 & 0.83 & 0.07 \\ 0.02 & 0.07 & 0.91 \end{bmatrix},$$

which is a doubly stochastic matrix, i.e. it sums to 1 in all directions.

For a degenerate example the procedure is repeated with the vector

$$\hat{\mu} = [1.2 \quad 2.9 \quad 3.9],$$

which after applying the normalization from (3) becomes

$$[0.9 \quad 2.17 \quad 2.92].$$

Since 0.9 is smaller than 1 the problem is not well specified and a distribution can not be found. However, as mentioned in above, it is still possible to solve the optimization problem. This produces the estimate

$$\hat{P} = \begin{bmatrix} 0.98 & 0.00 & 0.00 \\ 0.01 & 0.85 & 0.15 \\ 0.01 & 0.15 & 0.86 \end{bmatrix}, \tag{5}$$

which is not a doubly stochastic matrix. This is problematic but if the object is to find the distribution of the horse placing first in the race it is possible to change the second constraint of the optimization:

*ii)* $\sum_k^n p_{k1} = 1$.

That is, require that the distribution is proper for the first position but not for the others.

## Application to a Regression Example

The previous section described a method for obtaining a probability distribution given a set of expectations. This section will estimate the expectations and plug them into the developed methodology.

Let $H$ be a corpus of features relating to targets $y$. The targets are assumed to be of the racing type described in the previous section. Let $T$ be the number of observed races. For each race indexed by $t$ $y_t^k$ denotes the finishing position for horse $k$. The vector $H_t^k$ contains the $m$ features for horse $k$ in race $t$. So each race is described by the resulting vector $y_t$ of length $n_t$ with finishing positions for the horses participating in the race, and the matrix $H_t$ of dimension $m \times n$ containing the input data to the race. The raw data are the same as Josefsson and Hellander (2014) and further described in Appendix A.

### *Two-Step Procedure*

A given model will be calibrated to a subset, $H_{\text{train}}$, of the data. Given the calibrated model an estimate or distribution of some unobserved $y_{\text{validation}}$ is desired. It is created using $H_{\text{validation}}$ which always is available before the race occurs. The algorithm works in two steps, first the expected positions of the race are estimated and then the induced probability distribution is extracted from the expected positions. The procedure is described in Algorithm 1 below.

**Algorithm 1** Two-step procedure

(1) Calibrate the model to the data.

$$\theta_{\text{train}} \leftarrow \underset{\theta}{arg\,min}\, L[f(H_{\text{train}}, \theta) - y_{\text{train}}]$$

(2) Predict the probability distribution for a race in the validation corpus:

### Training Methods

Three methods are used in the paper. Ridge regression, neural networks and logistic regression. The methods and details on their implementations are given below.

The linear regression model is given by,

$$\hat{y} = f(h, w) = w^{\mathrm{T}} H.$$

Ridge regression, introduced by Hoerl and Kennard (1970), adds an $\ell_2$ regularization on the parameters in ordinary least squares regression. That is, implementing ridge regression in our calibration step gives the following estimate of $w$

$$w_{\text{train}} \leftarrow \underset{w}{\arg\min} \left\| w^{\mathrm{T}} H_{\text{train}} - y_{\text{train}} \right\|_2^2 + \lambda \|w\|_2^2,$$

where $\lambda$ is called the *tuning parameter*.

In regression it is common to transform data. A popular data transformation is single layer feedforward neural network given by the tranformation $\Phi(H) \mapsto \sigma(\alpha^{\mathrm{T}} H)$. The prediction, given an input $H$, is then

$$\hat{y} = w^{\mathrm{T}} \Phi(H) = \sum_{j=1}^{N_{\text{nodes}}} \Phi(H) w_j = \sum_{j=1}^{N_{\text{nodes}}} \sigma(\alpha_j^{\mathrm{T}} H) w_j = \sum_{j=1}^{N_{\text{nodes}}} \sigma\left( \sum_{i=1}^{m} \alpha_{j,i} H_i \right) w_j.$$

The activation function is chosen to be a ramp, i.e. $\sigma(x) = \max(x, 0)$. This is a popular choice referred to as a rectifing linear unit (ReLU), LeCun, Bengio, and Hinton (2015).

To train the network a naive, but efficient, approach is employed: the vectors $\alpha$ are initialized randomly and normalized so that their euclidean norm is 1. Then the weights, $w$, are estimated using ridge regression

$$w_{\text{train}} \leftarrow \underset{w}{\arg\min} \left\| w^{\mathrm{T}} \Phi(H_{\text{train}}) - y_{\text{train}} \right\|_2^2 + \lambda \|w\|_2^2.$$

The computational cost of a net trained this way is low; the costliest parts are ridge regression and the evaluation of the activation function. The ReLu is cheap to evaluate, most of the time is spent computing the dot product $\alpha^{\mathrm{T}} H$.

The two-step procedure will be compared to multinomial logistic regression. The prediction for the probability that a horse finishes in position $k$ out of $n$ is then given as

$$\hat{\mathbb{P}}(y_{\text{validation}} = k) = \hat{\mathbb{P}}(y_{\text{validation}} = n) e^{H_{\text{validation}} w_k},$$

$$\hat{\mathbb{P}}(y_{\text{validation}} = n) = \frac{1}{1 + \sum_{k=1}^{n-1} e^{H_{\text{validation}} w_k}}.$$

See Hastie et al. (2009). Training this model is a nonlinear problem which, compared to linear regression, is difficult.

## Application to Harness Racing

This section studies an example in which there is a model for predicting placement of horses in a trot race.

The induced probabilities will be compared to both the market odds and to the probabilities obtained from logistic regression.

### *Models*

Three predictive models are compared

(1) The two-step procedure
(2) Multinomial logistic regression
(3) The ranking and distribution induced by the odds

In the two-step procedure the training is done using ADMM. The two models are ridge regression and the neural network from Section 3.2. The distribution is obtained using the attached code from Figure 4. The Multinomial logistic regression problem is solved using MATLAB. The odds induce a ranking used as a benchmark: The horse with the lowest odds is ranked first, the horse with second lowest odds is ranked second and so on.

### *Description of Data*

The data are the same as was used in Josefsson and Hellander (2014). It comprises over 900 races from 2009 to 2012. Features of the data are described in Appendix A. The data are divided into two parts, calibration and validation. Several experiments with different divisions of the data are made, see Table 1.

Consider three metrics: the first is the accuracy or the amount of correct predictions of the winners. The second is the mean absolute deviation (MAD) from the predicted order against the observed order for each of the three predictions. Finally, consider the MAD for the winner. The odds has the smallest MAD for the winner since the ranking predicts a single horse while the other methods will always give a distribution over all the horses.

**Table 1.** Experiments.

| Experiment | Initial Calibration | Online update |
|---|---|---|
| 1 | 50% | |
| 2 | 10% | |
| 3 | 10% | True |

The data are described in the Appendix. Factors which should be taken into account have been studied, for instance Entin (2007) concludes that gender does make a difference. The genetic trend is examined in Gaffney and Cunningham (1988).

Factors for which there are no data in this study have been examined by others. For instance, Pfau et al. (2009) show that the riding style is important for the performance and Ratzlaff et al. (2005) study hoof-acceleration patterns in detail.

## Method

The data consist of nearly a thousand races and a number of features described in the appendix. The data are divided into two parts: calibration and validation. In the two-step procedure ridge regression is used for estimating the model; the tuning parameter $\lambda$ is set to 0.01.

The induced distribution is found using ADMM; both the tuning parameter and the tolerance are set to 0.001. Substituting CVX for ADMM improves the metrics, but by less than one percent.

MATLABs function for ordinal logistic regression is used to fit the logistic model. Sorting the odds from highest to lowest gives a ranking which is used as their prediction.

Three experiments were conducted and are presented in Table 1. The first experiment is a crossvalidation where 50% of the data are used to calibrate the model and the other half to validate. The second experiment is done in the same way but only 10% of the data are used for calibration. The third experiment differs from the other two. Initially the model is calibrated against 10% of the data but then, as new data become available, the model is recalibrated. In forecasting this is referred to as backtesting and simulates how the model works in practice.

## Results

The results are presented in Table 2. The two-step procedure is implemented in two versions: 2-ridge and 2-neural. The two-step procedure ridge surpasses the Multinomial logistic with respect to all three metrics. The neural network excels with respect to the MAD-metric while having poor performance on the other metrics. Clearly the odds are superior to the other methods when it comes to finding the winner but for the other positions it performs worse.

In Figures 1 and 2 the crossvalidations described in the experiments are presented. The correct prediction ratio is plotted with its exponential moving average, Holt (1957). In Figure 3 the result of the backtest, starting with 10% of the observed data are given. The induced distribution is computed using regression. The neural network is not in the figures.

**Table 2.** Results from prediction.

| Method | Experiment | %Accuracy | MAD | MAD-winner |
|---|---|---|---|---|
| 2-Ridge | 1 | **0.3866** | 3.3281 | 0.1424 |
| 2-Neural | 1 | 0.3048 | **3.3146** | 0.1729 |
| Logistic | 1 | 0.3755 | 3.3329 | 0.1572 |
| Odds | 1 | 0.3717 | 3.3438 | **0.1104** |
| 2-Ridge | 2 | **0.4008** | 3.2761 | 0.1453 |
| 2-Neural | 2 | 0.1840 | **3.2402** | 0.1828 |
| Logistic | 2 | 0.3476 | 3.3245 | 0.1597 |
| Odds | 2 | 0.3947 | 3.2728 | **0.1121** |
| 2-Ridge | 3 | **0.3967** | 3.2784 | 0.1416 |
| 2-Neural | 3 | 0.1472 | **3.0927** | 0.1885 |
| Logistic | 3 | 0.3906 | 3.2906 | 0.1585 |
| Odds | 3 | 0.3947 | 3.2728 | **0.1121** |

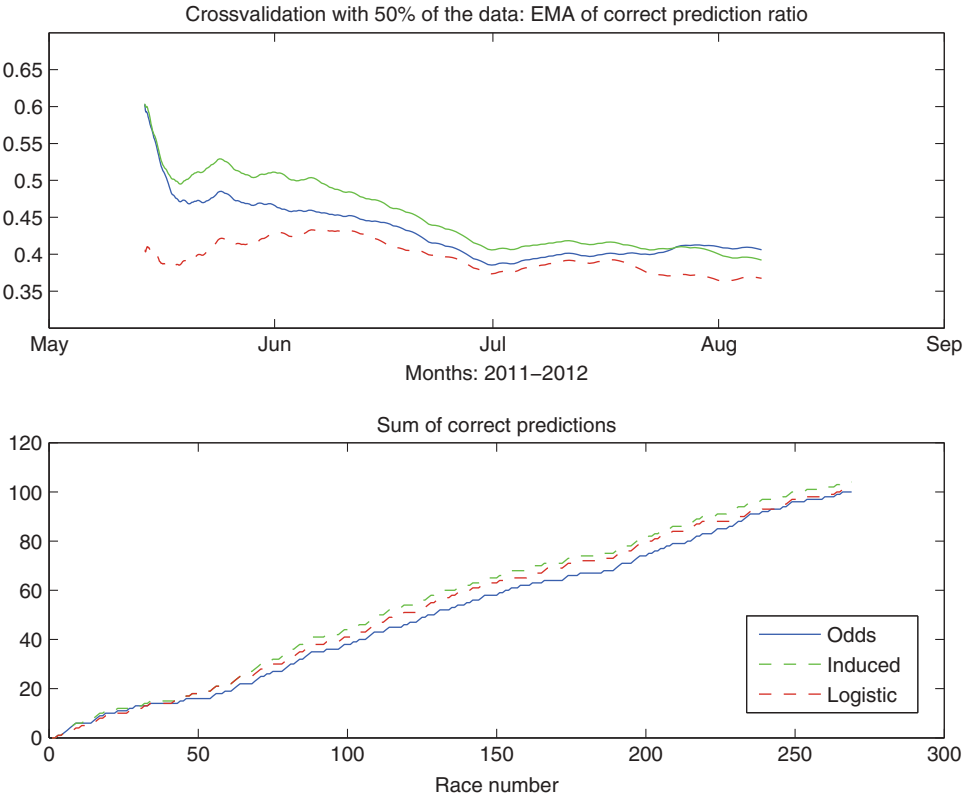


**Figure 1.** Cross validation 50%.

The ADMM-solver is much faster than both the CVX and the calibration of the Multinomial logistic model. It is implemented such that the probabilities are not required to be smaller than 1. This restriction is imposed by the implementation in CVX resulting in a minor improvement in the metrics but it is not included in the table.

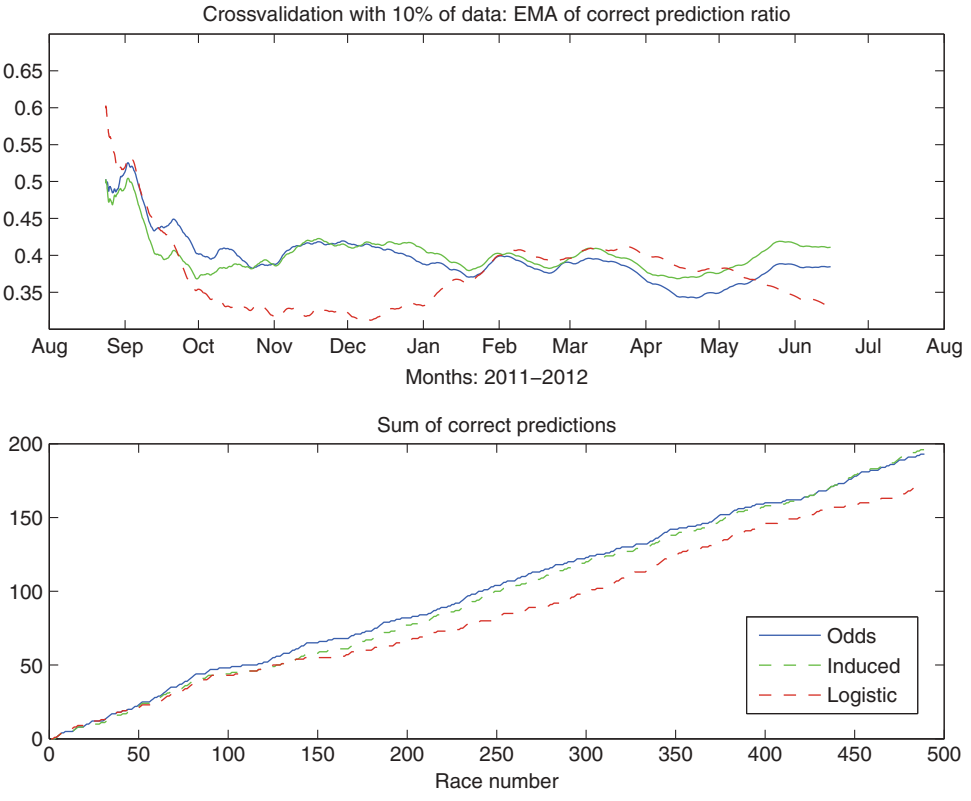


**Figure 2.** Cross validation 10%.

## *Comments*

It should be noted that ordinal logistic regression was implemented but performed worse than multinomial logistic regression with respect to all three metrics. It was therefore discarded from the study.

A drawback of logistic regression is that the model is nonlinear and therefore difficult to train; another drawback is that the interaction between inputs is restricted to be linear. This in contrast to the two-step procedure where a very complex model can be trained in the first step and in the second step extract the probabilities independently of how the model was estimated. An advantage of the multinomial logistic regression is that once it is calibrated it can produce predictions fast. However, this advantage is obliterated since estimation of the probability distribution in the two-step procedure can be done very fast using ADMM.

Calibration of the ridge regression executes in less than one millisecond; the logistic regression is more than 6000 times slower. Experiments suggest an increase in execution time by the square root of the size of the data. This means that ridge regression would be $60,000$ times faster if the size of the

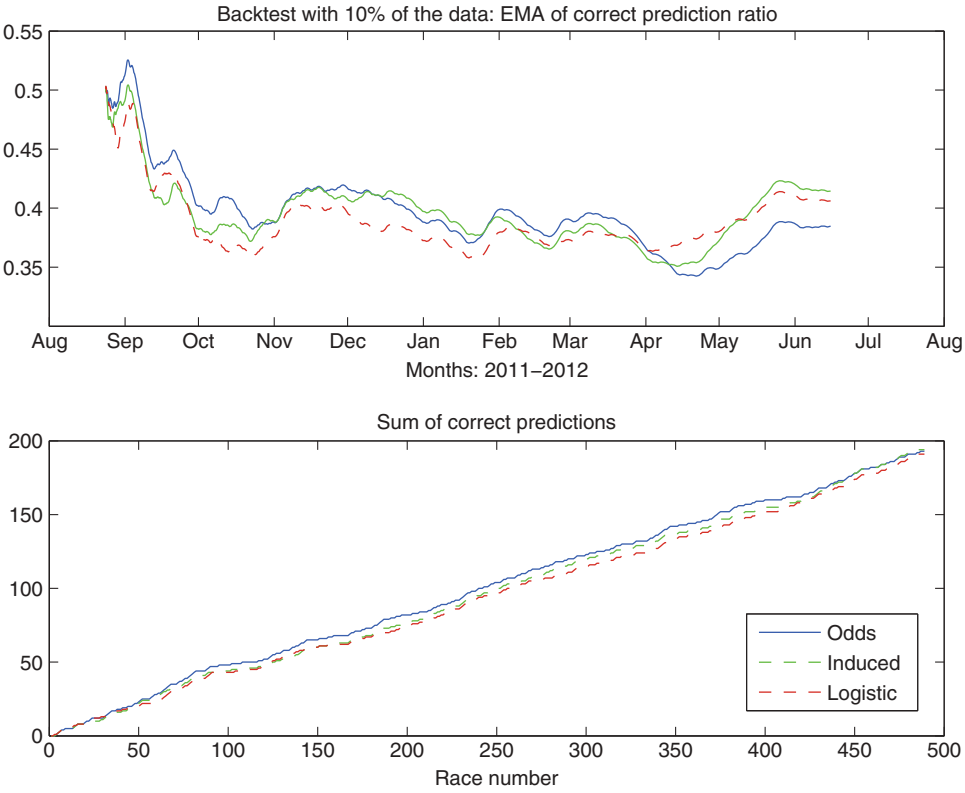


**Figure 3.** Backtesting 10%.

data were a hundred times larger. These figures depend on the implementation and can possibly be improved but never reconciled.

## Discussion

In this paper a two-step procedure for estimating a probability distribution induced by a ranking is presented. The clear advantage compared to logistic regression is the ease and speed of calibration. Furthermore, it allows more freedom and more sophisticated modeling of the data while keeping the probability distribution describing the ranking non-parametric. Another interesting attribute is that the method can accept any ranking as input and can, therefore, be more intuitive for humans. It is easier to imagine one horse finishing first, one in place 1.5 one in 2 and the rest in lower places than to think of the implied probabilities.

With this framework it might be tempting to use the method for more general classification; however, the idea is based on the fact that there is an ordering between the different classes. The probabilities must be computed in relation to each other which is a limitation.

Interesting future work would be to benchmark or combine the method with other ranking engines.

The method is preferable to logistic regression when the problem is large or when frequent re-estimation of the model is needed. This is for instance definitely the case in e-commerce problems.

The method is easy to use and the results are good. The field of ranked events is becoming increasingly important and we consider the proposed method a relevant addition to the field.

## Acknowledgments

The authors are grateful to Martin Hellander and Jonas Josefsson for their work with data collection. Jonas Hallgren and Timo Koskis funding was provided by the Swedish Research Council (Grant Number 2009- 5834).

## References

Adelman, M. L. 1981. The first modern sport in America: Harness racing in New York city. *Journal of Sport History* 80 (1): 5–32.

Aldous, D. 2015. Data science for everyone, and probability models meet player ratings. *Bernoulli News* 220 (1):0 6–7.

Beck, M., and D. Pixton. 2003. The Ehrhart polynomial of the Birkhoff polytope. *Discrete & Computational Geometry* 300 (4):0623–637. doi:10.1007/s00454-003-2850-8.

Birkhoff, G. 1946. Three observations on linear algebra. *University Nac Tucumán Revista A* 5:0 147–151.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 30 (1):01–122.

Bradley, R. A., and M. E. Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 390 (3/4):0324–345.

Breese, J. S., D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 43–52, Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998.

Carroll, L. 1883. Lawn tennis tournaments. *St. James's Gazette* 1: 5–6.

Coleman, L. 2004. New light on the longshot bias. *Applied Economics* 360 (4):0315–326. doi:10.1080/0003684041000167424O.

Davoodi, E., and A. R. Khanteymoori. 2010. Horse racing prediction using artificial neural networks. *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing*, 2010:155–160.

Elo, A. E. 1978. *The rating of chessplayers, past and present*. New York: Arco Pub..

Elton, E. J., M. J. Gruber, S. J. Brown, and W. N. Goetzmann. 2014. *Modern portfolio theory and investment analysis*. New Jersey: John Wiley & Sons.

Entin, P. 2007. Do racehorses and Greyhound dogs exhibit a gender difference in running speed? *Equine and Comparative Exercise Physiology* 40 (3–4):0135–140.

Ferrucci, D., E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. William Murdock, E. Nyberg, J. Prager, et al. 2010. Building Watson: An overview of the DeepQA Project. *AI Magazine*. 310(3):059–79. doi:10.1609/aimag.v31i3.2303.

Freund, Y., R. Schapire, and N. Abe. 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence* 140 (771–780):0 1612.

Fukushima, M. 1992. Application of the alternating direction method of multipliers to separable convex programming problems. *Computational Optimization and Applications* 10 (1):093–111. doi:10.1007/BF00247655.

Gabay, D., and B. Mercier. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* 20 (1):017–40. doi:10.1016/0898-1221(76)90003-1.

Gaffney, B., and E. P. Cunningham. 1988. Estimation of genetic trend in racing performance of thoroughbred horses. *Nature* 3320 (6166):0722–724. doi:10.1038/332722a0.

Gaines, S. D., and W. R. Rice. 1990. Analysis of biological data when there are ordered expectations. *American Naturalist* 310–317. doi:10.1086/285047.

Glickman, M. E. 1999. Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics* 48:377–394.

Glowinski, R., and A. Marroco. 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique Et Analyse Numérique* 90 (R2):041–76.

Grant, M., and S. Boyd. 2008. Graph implementations for nonsmooth convex programs. In *Recent advances in learning and control, lecture notes in control and information sciences*, ed. V. Blondel, S. Boyd, and H. Kimura, 95–110. New York: Springer-Verlag Limited. http://stanford.edu/boyd/graph_dcp.html

Grant, M., and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

Hastie, T., R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. 2009. *The elements of statistical learning*. New York: Springer-Verlag.

Herbrich, R., T. Minka, and T. Graepel. 2006. Trueskill: A Bayesian skill rating system. In *Advances in neural information processing systems*, 569–576. Vancouver, Canada: NIPS 2006.

Hoerl, A. E., and R. W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 120 (1):055–67.

Holt, C. C. Forecasting seasonals and trends by exponentially weighted moving averages. Technical report, DTIC Document, 1957.

Josefsson, J., and M. Hellander. Prediction of swedish harness racing. *KTH Bachelor thesis, OAI: oai: DiVA.org:kth-140703*, 2014.

Ko, J., L. Si, and E. Nyberg. 2010. Combining evidence with a probabilistic framework for answer ranking and answer merging in question answering. *Information Processing & Management* 460 (5):0541–554. doi:10.1016/j.ipm.2009.11.004.

LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 5210 (7553):0436–444. doi:10.1038/nature14539.

Lessmann, S., M.-C. Sung, and J. E. V. Johnson. 2009. Identifying winners of competitive events: A SVM-based classification model for horserace prediction. *European Journal of Operational Research* 1960 (2):0569–577. doi:10.1016/j.ejor.2008.03.018.

Luenberger, D. G. 1997. *Optimization by vector space methods*. New York: Springer-Verlag.

Nishihara, R., L. Lessard, B. Recht, A. Packard, and M. I. Jordan. 2015. A general analysis of the convergence of ADMM. International Conference on Machine Learning, Volume 37, pp. 343-352, Lille, France, 7-9 July 2015. *arXiv preprint arXiv:1502.02009*.

Pak, I. 2000. Four questions on Birkhoff polytope. *Annals of Combinatorics* 40 (1):083–90. doi:10.1007/PL00001277.

Pfau, T., A. Spence, S. Starke, M. Ferrari, and A. Wilson. 2009. Modern riding style improves horse racing times. *Science* 3250 (5938):0289–289. doi:10.1126/science.1174605.

Pieramati, C., L. Fusaioli, L. Scacco, L. Buttazzoni, and M. Silvestrelli. 2010. On the use of Elo rating on harness racing results in the genetic evaluation of trotter. *Italian Journal of Animal Science* 60 (1s):0189–191.

Ratzlaff, M. H., P. D. Wilson, D. V. Hutton, and B. K. Slinker. 2005. Relationships between hoof-acceleration patterns of galloping horses and dynamic properties of the track. *American Journal of Veterinary Research* 660 (4):0589–595. doi:10.2460/ajvr.2005.66.589.

Schumaker, R. P. 2013. Machine learning the harness track: Crowdsourcing and varying race history. *Decision Support Systems* 540 (3):0 1370–1379. doi:10.1016/j.dss.2012.12.013.

Silverman, N., and M. Suchard. 2013. Predicting horse race winners through a regularized conditional logistic regression with frailty. *The Journal of Prediction Markets* 70 (1):043–52.

Thaler, R. 1992. *The winner's curse. paradoxes and anomalies of economic life*. Princeton, NY: Princeton University Press.

Vaughan Williams, L. 2005. *Information efficiency in financial and betting markets*. Cambridge, England: Cambridge University Press.

Von Neumann, J. 1953. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games* 2:0:5–12.

## Appendix A. Data Description

The features of the data are given in Table 3.

**Table 3.** Variable description.

| Variable name | Description | Binary |
|---|---|---|
| Heavy track | Stronger horses perform better | True |
| Odds | Odds given by the betting company | |
| Autostart | Car used for start | |
| Gallop | Leads to disqualification | True |
| Shoes Front | Helps on heavy tracks | True |
| Shoes Back | Helps on heavy tracks | True |
| Age | Age of horse | |
| Stallion | Male horse | True |
| Gelding | Castrated horse | True |
| Starting number | Lower starting number is better | |
| Good start number | Top half | True |
| Starting score | Based on the last five races. | |
| Time | Best historical time on the distance | |
| Result | Finishing position, last if failed to finish | |

## Appendix B. Conditions on the Expectations

The notation in this section is as before, but here repeated. That is, $X_k$ denotes the finishing position of horse number $k$ and $\mu_k$ its expected position. The vector containing the expected positions is denoted $\mu$ and its estimate by $\hat{\mu}$. The normalized expectations, obtained from equation eq:normalization are denoted $\tilde{\mu}$.

There is a total of $n$ horses. The element $p_{kj}$ of the matrix $P$ denotes the probability that horse number $k$ finishes in position $j$. That is, the $k$'th row of $p$ gives the distribution of the finishing position for the $k$'th horse while the $j$'th column gives the distribution for the $j$'th position. Since $P$ sums to 1 in all directions it is a stochastic matrix.

**Proposition B.1.**

- *Given a set of estimated expectations $\hat{\mu}$ there exists a normalized estimate $\tilde{\mu}$, given by (3), which satisfies*
$$\sum_{k=1}^{n} \tilde{\mu}_k = \frac{1}{2} n(n+1).$$
- *For the set of normalized estimated expectations $\tilde{\mu}$ there exists a probability distribution only if the sum of the expectations is equal to $n(n+1)/2$.*

*Proof.*

- By (2) the sum of the expectations, $\sum_{k=1}^{n} \mu_k$, is equal to
$$\frac{1}{2} n(n+1).$$
That any normalized sum of expectations is equal to $n(n+1)/2$ follows from (4).
- For any probability distribution it must hold that
$$\sum_{j=1}^{n} j \sum_{k=1}^{n} p_{kj} = \sum_{j=1}^{n} j = \frac{1}{2} n(n+1).$$
On the other hand, by definition every set of ordered expectations must satisfy
$$\sum_{k=1}^{n} \tilde{\mu}_k = \sum_{k=1}^{n} \sum_{j=1}^{n} j\, p_{kj} = \sum_{j=1}^{n} j \sum_{k=1}^{n} p_{kj}.$$
Therefore, there exists a probability distribution for $\tilde{\mu}$ only if
$$\sum_{k=1}^{n} \tilde{\mu}_k = \frac{1}{2} n(n+1).$$

□

**Proposition B.2**. *If $P$ is a stochastic matrix then all elements in $\mu$ must lie in the interval $[1, n]$.*

*Proof.* For each element $\mu_k$ in $\mu$
$$\mu_k = \sum_{j=1}^{n} j p_{kj} \geq \sum_{j=1}^{n} p_{kj} = 1,$$
and
$$\mu_k = \sum_{j=1}^{n} j p_{kj} \leq \sum_{j=1}^{n} n p_{kj} = n.$$

The argument above can be used for columns of $P$ in the same way. □

**Proposition B.3**. *Let $X$ be a random variable with distribution $p$ taking values in $1, \ldots, n$. Denote its moments $\mathbb{E}[X^k]$ by $\mu^{(k)}$. Then $p$ is uniquely determined by its moments.*
*Proof.* First observe that

$$\mu^{(k)} = \sum_{j=1}^{n} j p_j \leq n^k \sum_{j=1}^{n} = n^k$$

for all $k$. Thus, for every $M$

$$\sum_{k=0}^{M} \frac{\mu^{(k)}|t|^k}{k!} \leq \sum_{k=0}^{M} \frac{(n|t|)^k}{k!} \leq e^{n|t|}$$

so the moment generating function $g(t) = \sum_{k=0}^{\infty} \frac{\mu^{(k)} t^k}{k!}$ exists and is differentiable for all $t$. The $k$th moment is obtained by evaluating the $k$th derivative of $g$ at the origin: $g^{(k)}(0) = \mu^{(k)}$. For two distinct set of moments there would be two distinct set of derivatives of $g$. Therefore $\mu$ uniquely determines $g$. By the uniquness of moment generating functions $g$ determines $p$. We conclude that $\mu$ uniquely determines $p$. □

## Appendix C. MATLAB Code

The function described in Figure 4 estimates the distribution given a set of expectations.

```
function p = Expectation2Probability(mu)
% Input is a row-vector
% Pre-Normalize
mu = mu - min(mu) + 1;
n = length(mu);
R_ = 0.5 * (n^2 + n); S_ = sum(mu); R = R_/S_;
mut = mu * R;
if min(mut) < 1 || max(mut) > n
    warning('Misspecification! Normalized mu must have ...
        elements in [1, n]')
end
a = 1:n;
b = ones(1, n);
Ca = [1 zeros(1, n-1)];
C0 = []; for j = 1:n, C0 = [C0 Ca]; end
A = []; B = []; C = [];
for j = 1:n
    A = blkdiag(A, a);
    B = blkdiag(B, b);
    C = [C; zeros(1, j-1) C0(1:end-j+1)];
end
X = [A; B; C];
[mux nx] = size(X);
muOpt = [mut ones(1, mux-n)]';
p = ADMMsolver(X, muOpt);
p = reshape(p, n, n)';
end
```

**Figure 4.** MATLAB-function.