Supervised Aggregation of Classifiers using Artificial Prediction Markets

Nathan Lay   NLAY@FSU.EDU

Department of Scientific Computing, Florida State University, Tallahassee, Florida 32306, USA

Adrian Barbu   ABARBU@STAT.FSU.EDU

Department of Statistics, Florida State University, Tallahassee, Florida 32306, USA

Abstract

Prediction markets are used in real life to predict outcomes of interest such as presidential elections. In this work we introduce a mathematical theory for Artificial Prediction Markets for supervised classifier aggregation and probability estimation. We introduce the artificial prediction market as a novel way to aggregate classifiers. We derive the market equations to enforce total budget conservation, show the market price uniqueness and give efficient algorithms for computing it. We show how to train the market participants by updating their budgets using training examples. We introduce classifier specialization as a new differentiating characteristic between classifiers. Finally, we present experiments using random decision rules as specialized classifiers and show that the prediction market consistently outperforms Random Forest on real and synthetic data of varying degrees of difficulty.

1. Introduction

Prediction markets, also known as information markets, are forums that trade contracts that yield payments dependent on the outcome of future events of interest. They have been used in the US Department of Defense (Polk et al., 2003), health care (Polgreen et al., 2006), to predict presidential elections (Wolfers & Zitzewitz, 2004) and in large corporations to make informed decisions (Cowgill et al., 2008). The prices of the contracts traded in these markets are good approximations for the probability of the

outcome of interest (Manski, 2006; Gjerstad & Hall, 2005). Prediction markets are capable of fusing the information that the market participants possess through the contract price. For more details, see (Arrow et al., 2008).
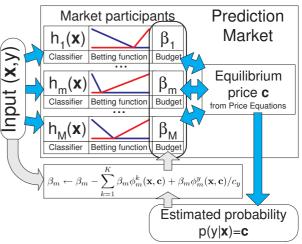

Figure 1. The Artificial Prediction Market. Given a feature vector $\mathbf{x}$, the market equilibrium price vector $\mathbf{c}$ is computed from the Price Equations (11), with c_k an estimator of $P(Y = k|\mathbf{x})$. Online training on an example $(\mathbf{x}, y)$ is achieved through **Budget Update** $(\mathbf{x}, y, \mathbf{c})$ shown with upward gray arrows.

In this paper we develop a mathematical theory for Artificial Prediction Markets for the purpose of supervised aggregation of classifier and or probability estimators. Starting from the total budget conservation condition we derive the market equations. We show that under certain mild conditions, the market price is unique and give efficient algorithms for computing it. This market price will be the estimated probability given the evidence presented to the market participants though a feature vector $\mathbf{x}$. It is the result of the fusion of the information possessed by the market participants in their classifiers.

We show how the Artificial Prediction Market gen-

eralizes linear aggregation, the basis of Boosting (Friedman et al., 2000; Schapire, 2003) and Random Forest (Breiman, 2001). It turns out that for linear aggregation, each market participant will purchase contract for the class it predicts, regardless of the market price for that contract. We introduce other *betting functions* that make buying decisions not only based on the classifier prediction but also on the market price.

We introduce a new type of classifiers that are *specialized* in modeling certain regions of the feature space. Such classifiers have good accuracy in their region of specialization and are not used in predicting outcomes for observations outside this region. This means that for each observation, a different subset of classifiers will be aggregated to obtain the estimated probability, making the whole approach become a sort of *ad-hoc* aggregation. This is contrast to the general trend in Boosting where the same classifiers are aggregated for all observations.

We give examples of generic specialized classifiers as the leaves of Random Trees from a Random Forest. Experimental validation on thousands of synthetic datasets with Bayes errors ranging from 0 (very easy) to 0.5 (very difficult) as well as on real UCI data show that the Prediction Market using the specialized classifiers outperforms the Random Forest in prediction and in estimating the true underlying probability.

2. Artificial Prediction Market for Classifier Aggregation

This work simulates a real prediction market named the *Iowa Electronic Market* (Wolfers & Zitzewitz, 2004), on the web at http://www.biz.uiowa.edu/iem/. The Iowa Electronic Market works as follows: contracts are sold for all possible outcomes at the market price. A winning contract (that predicted the correct outcome) pays \$1 after the outcome is known.

2.1. Setup of the Artificial Prediction Market

If the possible classes (outcomes) are $1, ..., K$, we assume there exist contracts for each class, whose prices form a K-dimensional vector $\mathbf{c} = (c_1, ..., c_K) \in [0, 1]^K$. Let $\Omega \subset \mathbb{R}^F$ be the instance or feature space containing all the available information that can be used in making outcome predictions $p(Y = k|\mathbf{x}), \mathbf{x} \in \Omega$.

We define a *market participant* as a pair $(\beta, \phi(\mathbf{x}, \mathbf{c}))$ of a *budget* β and a *betting function* $\phi(\mathbf{x}, \mathbf{c}) : \Omega \times [0, 1]^K \to [0, 1]^K$. The budget β represents the weight of the the participant in the aggregation. The betting function tells what percentage of its budget the

market participant will allocate to purchase contracts for each class, based on the instance $\mathbf{x} \in \Omega$ and the market price $\mathbf{c}$. In this paper, we will only focus on betting functions based on trained classifiers $h(\mathbf{x}) : \Omega \to [0, 1]^K, \sum_{k=1}^K h^k(\mathbf{x}) = 1$. In order to bet at most the budget β, the betting functions must satisfy $\sum_{k=1}^K \phi^k(\mathbf{x}, \mathbf{c})) \leq 1$. We use one of the following three betting functions, also shown in Figure 2:
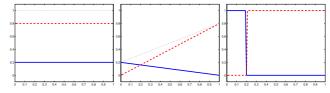

Figure 2. Betting functions. Left: Constant betting, Middle: Linear betting, Right: Aggressive betting. Shown are $\phi^0(\mathbf{x}, 1-c)$ (red), $\phi^1(\mathbf{x}, c)$ (blue), and the total amount bet $\phi^0(\mathbf{x}, 1-c) + \phi^1(\mathbf{x}, c)$ (black dotted). In this example, the classifier probability is $h^1(\mathbf{x}) = 0.2$.

- Constant betting functions
$$\phi^k(\mathbf{x}, \mathbf{c}) = h^k(\mathbf{x}) \qquad (1)$$

- Linear betting functions
$$\phi^k(\mathbf{x}, \mathbf{c}) = (1 - c_k)h^k(\mathbf{x}) \qquad (2)$$

- Aggressive betting functions
$$\phi^k(\mathbf{x}, \mathbf{c}) = \begin{cases} 1 & \text{if } c_k \leq h^k(\mathbf{x}) - \epsilon \\ 0 & \text{if } c_k > h^k(\mathbf{x}) \\ \frac{h^k(\mathbf{x}) - c_k}{\epsilon} & \text{otherwise} \end{cases} \qquad (3)$$

We suspect that the betting functions play a similar role to the *potential functions* from MRF modeling (Zhu et al., 1998).

The market consists of a number of market participants $(\beta_m, \phi_m(\mathbf{x}, \mathbf{c}))$.

2.2. Aggregation Using the Artificial Prediction Market

The aggregation result of the prediction market is the contract price vector $\mathbf{c}$ at equilibrium. In order for the contract prices to approximate a probability over the class labels and similar to (Gjerstad & Hall, 2005), we enforce the condition $\sum_{k=1}^K c_k = 1$. We use the price c_k as an estimator of $p(Y = k|\mathbf{x})$.

Algorithm 1 Prediction with the Market

Input: Feature vector $\mathbf{x} \in \Omega$.
Compute equilibrium price $\mathbf{c} = (c_1, ..., c_K)$ using Thm. 2.1
Output: Class conditional probability estimates $\hat{p}(Y = k|\mathbf{x}) = c_k, k = 1, ..., K$.

To simplify the behavior of the prediction market and without loss of generality, the contract prices in the Artificial Prediction Market are started at an equilibrium

price that is determined numerically and all transactions are made instantaneously at the this price. Hence in our setup, the market price does not fluctuate. This is possible because, as opposed to a real prediction market, we know the betting strategy of each market participant and we can find the equilibrium price numerically.

2.3. Training the Artificial Prediction Market

Training the market involves initializing all participants with the same budget β_0 and running the budget update on a set of training examples $(\mathbf{x}_i, y_i)$. For each example $(\mathbf{x}_i, y_i)$ the budgets β_m are updated based on the contracts purchased and the outcome y_i. After all training examples have been presented, the participants will have budgets that depend on how well they predicted the correct class y for each training example $\mathbf{x}$. This procedure is illustrated in Figure 1.

Algorithm 2 Prediction Market Training

   Input: Training examples $(\mathbf{x_i}, y_i)$
   Initialize all budgets $\beta_m = \beta_0$.
   for each training example $(\mathbf{x}_i, y_i)$ **do**
      Compute equilibrium price $\mathbf{c}_i$ using Thm. 2.1
      Run **Budget Update** $(\mathbf{x}_i, y_i, \mathbf{c}_i)$
   end for

The budget update procedure subtracts from the budget of each participant the amounts it bets for each class, then rewards the participant based on how many contracts it purchased for the correct class.

Participant m purchased $\beta_m \phi_m^k(\mathbf{x}, \mathbf{c})$ worth of contracts for class k, at price c^k. Thus the number of contracts purchased for class k is $\beta_m \phi_m^k(\mathbf{x}, \mathbf{c})/c_k$. In total, participant m's budget is decreased by the amount $\sum_{k=1}^K \beta_m \phi_m^k(\mathbf{x}, \mathbf{c})$ invested in contracts. Since participant m bought $\beta_m \phi_m^y(\mathbf{x}, \mathbf{c})/c_y$ contracts for the correct class y, he is rewarded the amount $\beta_m \phi_m^y(\mathbf{x}, \mathbf{c})/c_y$.

We will multiply all betting functions by a factor $0 < \eta < 1$ to control the percentage of their budget the market participants are allowed to bet for each training example. This does not change the equilibrium price but it makes the training less sensitive. In this work we fixed $\eta = 0.1$.

Algorithm 3 Budget Update $(\mathbf{x}, y, \mathbf{c})$

   Input: Training example $(\mathbf{x}, y)$, price vector $\mathbf{c}$
   for $m = 1$ **to** M **do**
      Update participant m's budget as
$$\beta_m \leftarrow \beta_m - \eta \sum_{k=1}^K \beta_m \phi_m^k(\mathbf{x}, \mathbf{c}) + \eta \frac{\beta_m}{c_y} \phi_m^y(\mathbf{x}, \mathbf{c}) \quad (4)$$
   end for

2.4. The Price Equations

The main principle the proposed artificial prediction market follows is *budget conservation*, i.e. the conservation of the sum of all participants budgets $\sum_{m=1}^M \beta_m$. Since for each instance, any of the outcomes are theoretically possible, we assume that *the total budget $\sum_{m=1}^M \beta_m$ must be conserved independent of the outcome y.*

This condition transforms into a set of equations that constrain the market price, which we call the price equations.

Theorem 2.1 Price Equations. *The total budget $\sum_{m=1}^M \beta_m$ is conserved after the* **Budget Update**$(\mathbf{x}, y, \mathbf{c})$, *independent of the outcome y, if and only if there exists $n \in \mathbb{R}_+$ such that*

$$\sum_{m=1}^M \beta_m \phi_m^k(\mathbf{x}, \mathbf{c}) = nc_k, \quad \forall k = 1, ..., K \quad (5)$$

The proof is given in the Appendix.

2.5. Price Uniqueness

The price equations together with the equation $\sum_{k=1}^K c_k = 1$ are enough to uniquely determine the market price $\mathbf{c}$, under mild assumptions on the betting functions $\phi^k(\mathbf{x}, \mathbf{c})$.

Observe that if $c_k = 1$ for some k, then the contract costs 1 and pays 1, so there is nothing to win. Hence in this case, it is reasonable to assume $\phi^k(\mathbf{x}, \mathbf{c}) = 0$. Also, if $c_k = 0$ for some k, then the contract costs 0 and pays 1, so there is everything to win. In this case, one should have $\phi^k(\mathbf{x}, \mathbf{c}) > 0$. This suggests a class of betting functions $\phi^k(\mathbf{x}, c_k)$ depending only on the price c_k that satisfy the conditions $\phi^k(\mathbf{x}, 0) > 0, \phi^k(\mathbf{x}, 1) = 0$ and are continuous and monotonically decreasing in c_k. For such betting functions we have

Remark 2.2 *If all $\phi_m^k(\mathbf{x}, c_k), m = 1, ..., M$ are continuous and monotonically decreasing in c_k with $\phi_m^k(\mathbf{x}, 0) > 0$ and $\phi_m^k(\mathbf{x}, 1) = 0$ then*

$$f_k(c_k) = \frac{1}{c_k} \sum_{m=1}^M \beta_m \phi_m^k(\mathbf{x}, c_k) \quad (6)$$

is continuous and strictly decreasing in c_k and for every $n \geq 0$ there is a unique $c_k = c_k(n)$ that satisfies $f_k(c_k) = n$.

The proof is given in the Appendix. From here we obtain the following, also proved in the Appendix.

Theorem 2.3 Monotonic Betting Functions. *If all betting functions $\phi_m^k(\mathbf{x}, c_k), m = 1, ..., M, k = 1, ..., K$ are continuous and monotonically decreasing, then for each $(\mathbf{x}, y)$ there is a unique price $\mathbf{c} = (c_1, ..., c_K)$ such that the total budget $\sum_{m=1}^M \beta_m$ is conserved after* **Budget Update**$(\mathbf{x}, y, \mathbf{c})$.

In practice, each $c_k(n)$ can be found by the bisection method, while n can also be found by the bisection method. This gives Algorithm 4 below.

Algorithm 4 Equilibrium Price Computation

Initialize $n_0 = \epsilon, n_1 = \eta \sum_{m=1}^{M} \beta_m$.
for $k = 0$ **to** K **do**
   Compute $c_k(n_0), c_k(n_1)$ using bisection method on eq (6).
end for
repeat
   Let $n = (n0 + n1)/2$.
   for $k = 0$ **to** K
      Compute $c_k(n)$ by bisection on eq (6).
   end for
   if $\sum_k c_k(n) > 1$ **then**
      $n_0 = n$
   else
      $n_1 = n$
   end if
until $|\sum_k c_k(n) - 1| < \epsilon$ or $n_1 - n_0 < \epsilon$

The n_1 initialization is due to the bound

$$n = \sum_{k=1}^{K} nc_k = \sum_{k=1}^{K} \sum_{m=1}^{M} \beta_m \phi_m^k(\mathbf{x}, \mathbf{c}) \leq \eta \sum_{m=1}^{M} \beta_m \quad (7)$$

because for each m, $\sum_{k=1}^{K} \phi_m^k(\mathbf{x}, \mathbf{c}) \leq 1$.

2.6. Constant Betting is Linear Aggregation

The simplest betting functions are constant, where the same amount is invested in contracts for any contract price, i.e. $\phi_m^k(\mathbf{x}, \mathbf{c}) = h_m^k(\mathbf{x})$. A constant betting function with $h^1(\mathbf{x}) = \mathbf{0.2}$ is illustrated in Figure 2, left.

Theorem 2.4 Linear Aggregation. *If all betting function are constant* $\phi_m^k(\mathbf{x}, \mathbf{c}) = h_m^k(\mathbf{x})$, *then the equilibrium price is obtained by linear aggregation*

$$\mathbf{c} = \frac{\sum_{m=1}^{M} \beta_m h_m(\mathbf{x})}{\sum_{m=1}^{M} \beta_m} = \sum_m \alpha_m h_m(\mathbf{x}) \quad (8)$$

This way the Artificial Prediction Market generalizes the linear aggregation of classifiers existent in Adaboost (Friedman et al., 2000; Schapire, 2003) and Random Forest (Breiman, 2001).

In particular, the Random Forest (Breiman, 2001) is a Prediction Market with Constant Betting (linear aggregation) where all participants are random trees with the same budget $\beta_m = 1, m = 1, ..., M$.

For Constant Betting, the budget update has an analytic form:

$$\beta_m \leftarrow \beta_m(1 - \eta) + \eta \beta_m \frac{h_m^y(\mathbf{x}) \sum_{j=1}^{M} \beta_j}{\sum_{j=1}^{M} \beta_j h_j^y(\mathbf{x})} \quad (9)$$

This is a novel online update rule for linear aggregation.

2.7. Two-class Formulation

For the two-class problem, i.e. $K = 2$, the budget equation can be simplified by writing $\mathbf{c} = (1 - c, c)$ and obtaining the *two-class budget equation*

$$(1-c) \sum_{m=1}^{M} \beta_m \phi_m^1(\mathbf{x}, c) - c \sum_{m=1}^{M} \beta_m \phi_m^0(\mathbf{x}, 1-c) = 0 \quad (10)$$

This can be solved numerically directly in c using the bisection method. Again, the solution is unique if $\phi_m^k(\mathbf{x}, c_k), m = 1, ..., M, k = 1, ..., K$ are continuous and monotonically decreasing. Moreover, the solution is guaranteed to exist if there exist m, m' such that $\beta_m > 0, \beta_{m'} > 0$ and $\phi_m^1(\mathbf{x}, 0) > 0, \phi_{m'}^0(\mathbf{x}, 1) > 0$.

3. Specialized Classifiers

Classifiers are usually suboptimal, due to not using the entire feature vector $\mathbf{x} \in \mathbf{\Omega}$, the way they are trained, or other reasons. In Boosting, the same classifiers are aggregated for each instance $\mathbf{x} \in \mathbf{\Omega}$. In many situations however, there exist rules that hold on subsets of Ω but not on the entire space Ω. Classifiers trained on such subsets $D_i \subset \Omega$, would have small misclassification error on D_i but unpredictable behavior outside of D_i. The Artificial Prediction Market can aggregate such classifiers, transformed into participants that don't bet anything outside of their domain of expertise $D_i \subset \Omega$. This way, for different observations $\mathbf{x} \in \mathbf{\Omega}$, different subsets of participants will contribute to the resulting probability estimate. We call these *specialized classifiers* since they only give their opinion through betting on observations that fall in their domain of specialization. This idea is illustrated on the following simple 2D example of a triangular region, shown in Figure 3, with positive examples inside the triangle and negatives outside. An accurate classifier for that region can be constructed using six market participants, one for each half-plane determined by each side of the triangle.

Three of these classifiers correspond to the three half planes that are outside the triangle. These participants have 100% accuracy in predicting the observations, all negatives, that fall in their half planes and don't bet anything outside of their half planes. The other three classifiers are not very good, and will have smaller budgets. On observations outside of the triangle, one or two of the high-budget classifiers will enforce the correct class through high bids. On observations inside the triangle, only the small-budget classifiers will participate but will be in agreement on the correct class. Running this market on 1000 positives and 1000 negatives showed that the market has a prediction accuracy of 100%.
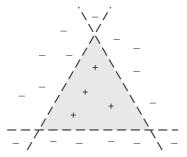
Figure 3. The triangular region can be perfectly classified by a market of six specialized classifiers that only bid on a half-plane determined by one side of the triangle. Three of these specialized classifiers have 100% accuracy while the other three have low accuracy. Nevertheless, the market is capable of obtaining 100% overall accuracy.

There are many ways to construct specialized classifiers, depending on the problem setup. In natural language processing for example, a specialized classifier could be based on grammar rules, which work very well in many cases, but not always.

We propose a generic set of specialized classifiers that are the leaves of the Random Trees of a Random Forest. Each leaf is a rule and will only work in a subdomain of the instance space Ω, but in that domain it will achieve good accuracy. In (Friedman & Popescu, 2008) these rules were combined using a linear aggregation method similar to Boosting. One could also use nodes of the random tree, not necessarily leaves, for the same purpose.

4. Related Work

Recent work in Economics (Manski, 2006; Perols et al., 2009; Plott et al., 2003) investigates the information fusion of the Prediction Markets. However, none of these works aims at using the prediction markets as a tool for learning class probability estimators in a supervised manner.

Some works (Perols et al., 2009; Plott et al., 2003) focus on parimutuel betting for combining classifiers. In parimutuel betting, contracts are sold for all possible outcomes (classes) and the entire budget (minus fees) is divided between the participants that purchased contracts for the winning outcome. Parimutuel betting has a different way of fusing information than the Iowa Prediction Market. The Information Based Decision Fusion (Perols et al., 2009) aggregates classifiers through parimutuel betting, using a loop that updates the odds for each outcome and takes updated bets until convergence. This ensures a stronger information fusion than without updating the odds. Our work is

different since it uses the Iowa Electronic Market instead of parimutuel betting, it presents a multi-class formulation instead of a two-class approach. Foremost, our work trains the market participants in a supervised way, whereas (Perols et al., 2009) does not train the market participants. We experimentally observed statistically significant improvements in prediction and probability estimation by using the trained market participants. Finally, our work evaluates the prediction market not only in terms of classification accuracy but also in the accuracy of predicting the exact class conditional probability given the evidence.

Specialization is a type of reject rule (Chow, 1970; Tortorella, 2004), but not for the aggregated classifier. Instead, each market participant has his own reject rule to decide on what observations to contribute to the aggregation. ROC-based reject rules (Tortorella, 2004) could be found for each market participant and used for defining its domain of specialization.

An overall reject rule can also be obtained for instances outside the specialization domain of all participants. No participant will bet for such an instance and this can be detected as an overall rejection. If the overall reject option is not desired, one could include in the market a set of participants that are all the leaves of a number of Random Trees. This way, by design it is guaranteed that each instance will fall into at least one leaf, i.e. participant, hence the instance will not be rejected.

A simplified specialization is present in delegated classifiers (Ferri et al., 2004). A first classifier would decide on the relatively easy instances and would delegate more difficult examples to a second classifier. This approach can be seen as a market with two non-overlapping participants. The specialization domain of the second participant is the complement of the first participant's domain.

The leaves of random trees (named rules) have also been used in (Friedman & Popescu, 2008) for linear aggregation. However, our work presents a more generic aggregation method through the prediction market, with linear aggregation as a particular case, and we view the rules as one sort of specialized classifiers that only bid in a subdomain of the feature space.

Linear aggregation of classifiers was also presented in (Bunea & Nobel, 2008), with an exponential weighting scheme. Our work goes beyond linear aggregation and presents a novel linear aggregation update that differs from the exponential weighting by being nonlinear and recursive.
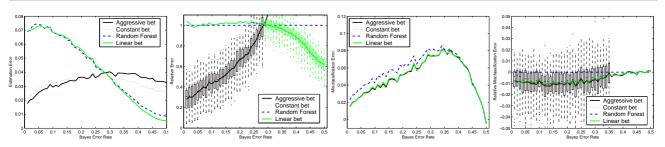
Figure 4. Evaluation on 5000 100D problems. Left: Class probability estimation error vs problem difficulty. Mid-left: Probability estimation errors relative to Random Forest. The Aggressive and Linear Betting are shown with box plots. Mid-right: Misclassification error minus Bayes error vs problem difficulty. Right: Misclassification errors relative to Random Forest. The Aggressive Betting is shown with box plots.

5. Experimental Results

We present results on real and synthetic datasets and compare with Random Forest. For each dataset, random trees are trained on bootstrap samples of the training data. The binary split of each node is selected using Information Gain from $\sqrt{F}$ randomly selected features. These trained random trees are used to construct the Random Forest and the three markets described below. This way only the aggregation capabilities of the different markets will be compared.

We evaluated four artificial prediction markets, having the same classifiers, namely the leaves of the trained random trees:

1. The RF market has constant betting and equal budgets for all participants. We proved in Section 2.6 that this is a Random Forest (Breiman, 2001).

2. The CB market has constant betting and the budgets are trained on the training set.

3. The LB market has linear betting and the budgets are trained on the training set.

4. The AB market has aggressive betting and the budgets are trained on the training set.

5.1. Evaluation of the Probability Estimation and Classification Accuracy

A series of experiments on synthetic datasets are performed to evaluate the market's ability to predict class conditional probabilities $P(Y|\mathbf{x})$. The markets are trained with 50 trees. The experiments are performed on 5000 binary datasets with 50 levels of Bayes error

$$E = \int \min\{p(\mathbf{x}, Y = 0), p(\mathbf{x}, Y = 1)\}d\mathbf{x},$$

ranging from 0.01 to 0.5 with equal increments. For each dataset, the two classes have equal frequency. Both $p(\mathbf{x}|Y = k), k = 0, 1$ are normal distributions $\mathcal{N}(\mu_k, \sigma^2 I)$, with $\mu_0 = 0, \sigma^2 = 1$ and μ_1 chosen in some random direction at such a distance to obtain the desired Bayes error.

For each of the 50 Bayes error levels, 100 datasets of size 200 were generated using the Bisection method to find an appropriate μ_1 in a random direction.

For each observation $\mathbf{x}$, the class conditional probability can be computed analytically using the Bayes rule

$$p^*(Y = 1|\mathbf{x}) = \frac{p(\mathbf{x}|Y = 1)p(Y = 1)}{p(\mathbf{x}, Y = 0) + p(\mathbf{x}, Y = 1)}$$

Each market is an estimator $\hat{p}(y = 1|\mathbf{x})$ that is compared to the truth $p^*(Y = 1|\mathbf{x})$ using the L_2 norm. This error is approximated using a sample of size 1000.

The probability estimates errors and the misclassification rates of the four markets are shown in Figure 4 for a 100D problem setup. Also shown on the right are the errors relative to the Random Forest. The relative probability estimation error is obtained by dividing each error to the corresponding Random Forest error. The probability estimates are significantly smaller ($\alpha < 0.01$) than the Random Forest for Bayes errors up to 0.28 for Aggressive and Constant betting markets and for Bayes error from 0.34 to 0.5 for Linear betting. It is interesting to note that all markets behave the same at Bayes error 0.3. The difference between these misclassification errors and the Bayes error are shown in Figure 4, mid-right. The difference between these misclassification errors and the Random Forest error are shown in Figure 4, right. All markets with trained participants predict significantly better ($\alpha < 0.01$) than Random Forest for Bayes errors up to 0.3, and behave similar to Random Forest for the remaining errors.

5.2. Evaluation on Real Data

We present in Table 1 an evaluation on 21 datasets from the UCI machine learning repository. Similar to (Breiman, 2001), the training and test sets are randomly subsampled from the available data, with 90% for training and 10% for testing. The exceptions are `satimage` and `poker` with test sets of size 2000 and 10^6. All results are averaged over 100 runs. The ADB

Table 1. Misclassification errors in percent (%) for 21 UCI datasets from the UC Irvine Repository. The markets evaluated are Random Forest (RF), and Constant (CB), Linear (LB) and Aggressive (AB) betting.

Data	Train Size	Test Size	Feat.	Cls	ADB	RFB	RF	CB	LB	AB
cancer	699	–	9	2	3.2	2.9	3.0	2.9	2.9	2.9
sonar	208	–	60	2	15.6	15.9	14.8	14.1	14.3	14.1
vowel	990	–	10	11	4.1	3.4	3.3	3.1 ●	3.2	3.1 ●
diabetes	768	–	8	2	26.6	24.2	23.4	23.4	23.4	23.5
ecoli	336	–	8	8	14.8	12.8	13.1	13.0	13.0	13.1
german	1000	–	20	2	23.5	24.4	23.7	23.7	23.6	23.7
glass	214	–	9	6	22.0	20.6	20.0	20.1	20.1	20.2
ionosphere	351	–	34	2	6.4	7.1	**5.8**	**5.7**	**5.7**	**5.7**
letter-recognition	20000	–	16	26	3.4	3.5	3.3	3.2 ●	3.2 ●	3.2 ●
satimage	4435	2000	36	6	8.8	8.6	*8.8*	8.6 ●	8.7 ●	8.6 ●
image	2310	–	19	7	1.6	2.1	**1.8**	**1.6** ●	**1.6** ●	**1.6** ●
vehicle	846	–	18	4	23.2	25.8	24.8	**24.5**	**24.6**	**24.5**
voting-records	435	–	16	2	4.8	4.1	3.0	3.0	3.0	3.0
car	1728	–	6	4	–	–	2.4	1.2 ●	1.4 ●	1.2 ●
poker	25010	1000000	10	10	–	–	38.0	35.7 ●	36.0 ●	35.7 ●
cylinder-bands	540	–	39	2	–	–	20.3	20.2	20.1	20.0
yeast	1484	–	9	10	–	–	35.9	35.8	35.7	35.8
magic	19020	–	10	2	–	–	12.0	11.7 ●	11.8 ●	11.8 ●
king-rook-vs-king	28056	–	6	18	–	–	21.6	11.0 ●	11.8 ●	11.0 ●
connect-4	67557	–	42	3	–	–	19.9	16.7 ●	16.9 ●	16.7 ●
splice-junction-gene	3190	–	59	3	–	–	4.9	4.6 ●	4.6	4.6 ●

and RFB columns are Adaboost and RF results taken from (Breiman, 2001). Significant mean differences ($\alpha < 0.01$) from RFB are **bold** for when RFB is worse and ***italic*** for when RFB is better. Significant paired t-tests (Demšar, 2006) ($\alpha < 0.01$) that compare the markets with our RF implementation are shown with ●, † for when RF is worse or better respectively. Both AB and CB significantly outperform RF on 10 datasets out of the 21 evaluated, while the LB outperforms RF on 8 datasets. None of the three markets was significantly outperformed by RF on any dataset.

6. Conclusion

In this paper we introduced a theory for Artificial Prediction Markets for supervised aggregation of probability estimators. We derived the equilibrium price equations, gave algorithms for computing it and showed that the Prediction Market generalizes linear aggregation. We also introduced specialized classifiers that bet on subsets of the instance space Ω. Experimental results real and synthetic data show that the Prediction Market usually outperforms the Random Forest in both prediction and probability estimation.

The Artificial Prediction Market is *updated online*, naturally multi-class and can obtain probability estimates when subsets of participants are involved for any instance $\mathbf{x} \in \Omega$. This feature could be useful for learning on manifolds (Agrafiotis & Xu, 2002; Belkin, 2003; Belkin & Niyogi, 2004; Elgammal & Lee;

Rahimi et al., 2005; Saul & Roweis, 2003), where the location on the manifold decides which participants should be involved. For example, in face detection, different face part classifiers (eyes, mouth, ears, nose, hair, etc) can be involved in the market, depending on the orientation of the head hypothesis.

References

Agrafiotis, D.K. and Xu, H. A self-organizing principle for learning nonlinear manifolds. *Proceedings of the National Academy of Sciences*, 99(25):15869–15872, 2002.

Arrow, K. J., Forsythe, R., Gorham, M., Hahn, R., Hanson, R., Ledyard, J. O., Levmore, S., Litan, R., Milgrom, P., and Nelson, F. D. The promise of prediction markets. *Science*, 320(5878):877, 2008.

Belkin, M. Problems of learning on manifolds. 2003.

Belkin, M. and Niyogi, P. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56(1):209–239, 2004.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Bunea, F. and Nobel, A. Sequential procedures for aggregating arbitrary estimators of a conditional mean. *IEEE Trans. Inf. Theory*, 54(4):1725–1734, 2008.

Chow, C. On optimum recognition error and reject trade-off. *IEEE Transactions on Information Theory*, 16(1): 41–46, 1970.

Cowgill, B., Wolfers, J., and Zitzewitz, E. Using prediction markets to track information flows: Evidence from Google. *Dartmouth College*, 2008.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:30, 2006.

Elgammal, A. and Lee, C.S. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR 2004*.

Ferri, C., Flach, P., and Hernández-Orallo, J. Delegating classifiers. In *ICML*, 2004.

Friedman, J., Hastie, T., and Tibshirani, R. Additive logistic regression: a statistical view of boosting. *Ann. Statist*, 28(2):337–407, 2000.

Friedman, J.H. and Popescu, B.E. Predictive learning via rule ensembles. *Ann. Appl. Stat.*, 2(3):916–954, 2008.

Gjerstad, S. and Hall, M.C. Risk aversion, beliefs, and prediction market equilibrium. *Economic Science Laboratory, University of Arizona*, 2005.

Manski, C.F. Interpreting the predictions of prediction markets. *Economics Letters*, 91(3):425–429, 2006.

Perols, J., Chari, K., and Agrawal, M. Information Market-Based Decision Fusion. *Management Science*, 55(5):827–842, 2009.

Plott, C.R., Wit, J., and Yang, W.C. Parimutuel betting markets as information aggregation devices: Experimental results. *Economic Theory*, 22(2):311–351, 2003.

Polgreen, P.M., Nelson, F.D., and Neumann, G.R. Use of prediction markets to forecast infectious disease activity. *Clinical Infectious Diseases*, 44(2):272–279, 2006.

Polk, C., Hanson, R., Ledyard, J., and Ishikida, T. The policy analysis market: an electronic commerce application of a combinatorial information market. In *ACM Conf. on Electronic Commerce*, pp. 272–273, 2003.

Rahimi, A., Darrell, T., and Recht, B. Learning appearance manifolds from video. In *CVPR*, 2005.

Saul, L.K. and Roweis, S.T. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.

Schapire, R.E. The boosting approach to machine learning: An overview. *Lect. Notes in Stat.*, pp. 149–172, 2003.

Tortorella, F. Reducing the classification cost of support vector classifiers through an ROC-based reject rule. *Pattern Analysis & Applications*, 7(2):128–143, 2004.

Wolfers, J. and Zitzewitz, E. Prediction markets. *Journal of Economic Perspectives*, pp. 107–126, 2004.

Zhu, S.C., Wu, Y., and Mumford, D. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

Appendix: Proofs

Proof of Theorem 2.1. From eq. (4), the total budget $\sum_{m=1}^{M} \beta_m$ is conserved if and only if

$$\sum_{m=1}^{M} \sum_{k=1}^{K} \beta_m \phi_m^k(\mathbf{x}, \mathbf{c}) = \sum_{m=1}^{M} \beta_m \phi_m^y(\mathbf{x}, \mathbf{c})/c_y \qquad (11)$$

Denoting $n = \sum_{m=1}^{M} \sum_{k=1}^{K} \beta_m \phi_m^k(\mathbf{x}, \mathbf{c})$, and since the above equation must hold for all y, we obtain that eq. (5) is a necessary condition.

Suppose now that there exists $n \in \mathbb{R}_+$ such that $\sum_{m=1}^{M} \beta_m \phi_m^k(\mathbf{x}, \mathbf{c}) = nc_k$ holds for all $k = 1, ..., K$. Summing over k we get $\sum_{m=1}^{M} \sum_{k=1}^{K} \beta_m \phi_m^k(\mathbf{x}, \mathbf{c}) = \sum_{k=1}^{K} nc_k = n$ since $\sum_{k=1}^{K} c_k = 1$. Hence eq. (11) holds and the total budget is conserved. ∎

Proof of Remark 2.2. Since $\sum_{m=1}^{M} \beta_m \phi_m^k(\mathbf{x}, c_k)$ is monotonically decreasing in c_k, then $\frac{1}{c_k} \sum_{m=1}^{M} \beta_m \phi_m^k(\mathbf{x}, c_k)$ is strictly decreasing in c_k. Since the total budget is conserved and is positive, there exists a $\beta_m > 0$, therefore $\sum_{m=1}^{M} \beta_m \phi_m^k(\mathbf{x}, 0) > 0$, which implies $\lim_{c_k \to 0} f_k(c_k) = \infty$. From the fact that $f_k(c_k)$ is continuous and strictly decreasing, with $\lim_{c_k \to 0} f_k(c_k) = \infty$ and $\lim_{c_k \to 1} f_k(c_k) = 0$, it implies that for every $n \geq 0$ there exists a unique c_k that satisfies $f_k(c_k) = n$. ∎

Proof of Theorem 2.3. From the above remark we get that for every $n \geq 0$ there is a unique $c_k(n)$ such that $f_k(c_k(n)) = n$. Moreover, following the proof of the above remark we see that $c_k(n)$ is continuous and monotonically strictly decreasing and $c_k(0) = 1, \lim_{n \to \infty} c_k(n) = 0$. Thus there is a unique n such that $\sum_{k=1}^{K} c_k(n) = 1$. For this n, from Theorem 2.1 follows that the total budget is conserved for the price $\mathbf{c} = (c_1(n), ..., c_K(n))$. Uniqueness follows from the uniqueness of $c_k(n)$ and the uniqueness of n. ∎

Proof of Theorem 2.4. The price equations (5) become:

$$\sum_{m=1}^{M} \beta_m h_m^k(\mathbf{x}) = nc_k, \quad \forall k = 1, ..., K.$$

From the constraint $\sum_{k=1}^{K} c_k = 1$ we obtain

$$n = \sum_{k=1}^{K} nc_k = \sum_{k=1}^{K} \sum_{m=1}^{M} \beta_m h_m^k(\mathbf{x}) = \sum_{m=1}^{M} \beta_m$$

since $\sum_{k=1}^{K} h_m^k(\mathbf{x}) = 1$. Finally we get

$$c_k = \frac{\sum_{m=1}^{M} \beta_m h_m^k(\mathbf{x})}{\sum_{m=1}^{M} \beta_m} = \sum_m \alpha_m h_m^k(\mathbf{x}), \; \forall k = 1, ..., K. \blacksquare$$