The Artificial Prediction Market

Adrian Barbu
Department of Statistics
Florida State University
Joint work with Nathan Lay, Siemens Corporate Research

Overview

Main Contributions

- A mathematical theory for Artificial Prediction Markets
  - Loss function.
  - Relation to existing methods:
    - Linear Aggregation
    - SVM
    - Logistic Regression
  - Extension to regression estimation.
  - Experimental comparison with Random Forest and Adaboost

Motivation

Main goal: Classification

- Let $\Omega \subset R^F$ be the instance space
- K possible classes (outcomes) {1,…,K}

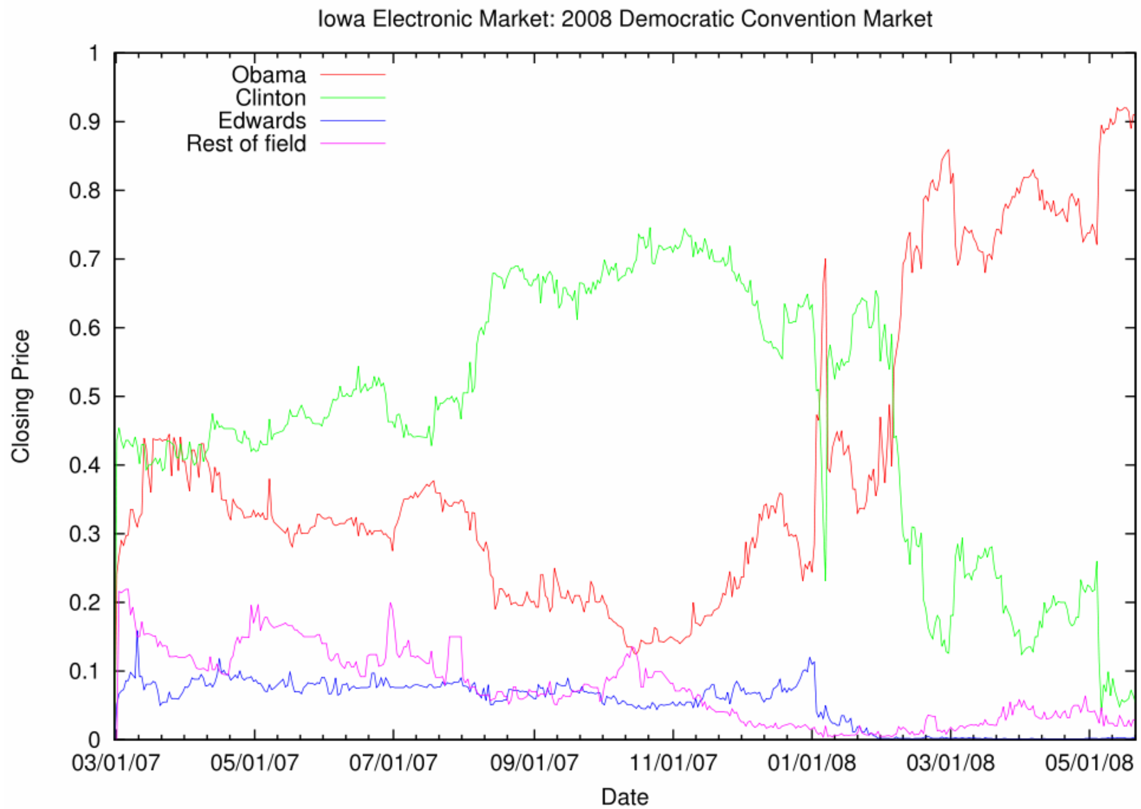
Supervised learning:

- Given training examples:
  - $(x_i, y_i) \in \Omega \times \{1,\ldots,K\}$
- Learn a function

$$\mathbf{f}(\mathbf{x}) : \Omega \rightarrow [0,1]^K, \; \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}))$$

such that $f_k(x)$ is a good approximation of $p(Y=k|x)$

The Iowa Electronic Market



- Market setup:
  - Contracts for each outcome are bought and sold at market price $0 < c < 1$
  - Each contract pays $1 if outcome is realized.
  - Market price of contract represents a good approximation of the probability that the corresponding event occurs

The Artificial Prediction Market

- Goal: predict class probability $p(y|x)$
- Market formulation:
  - Simulate the Iowa Electronic Market
  - Market participants = classifiers
  - Solve market price equations
    - Obtain total budget conservation
    - No price fluctuations
  - Train the market using training examples $(x_i,y_i) \in \Omega \times \{1,\ldots,K\}$
    - Participants bet on instance x_i
    - Wins are based on contracts purchased for correct class y_i
    - Participants become rich or poor based on prediction ability
  - The trained market predicts better

Other Prediction Markets

- Perols 2009
  - Parimutuel betting with odds update
  - Participants are not trained (have equal budgets)
  - Evaluated on UCI datasets
- Using the Market Maker
  - Chen and Vaughan, 2010, Abernethy et al, 2011
  - Participants enter the market sequentially
  - Are paid according to a scoring rule
  - See Tuesday’s tutorial
- Machine Learning Markets (Storkey 2011)
  - Participants bet to maximize a utility function
  - Equilibrium price is computed by optimization

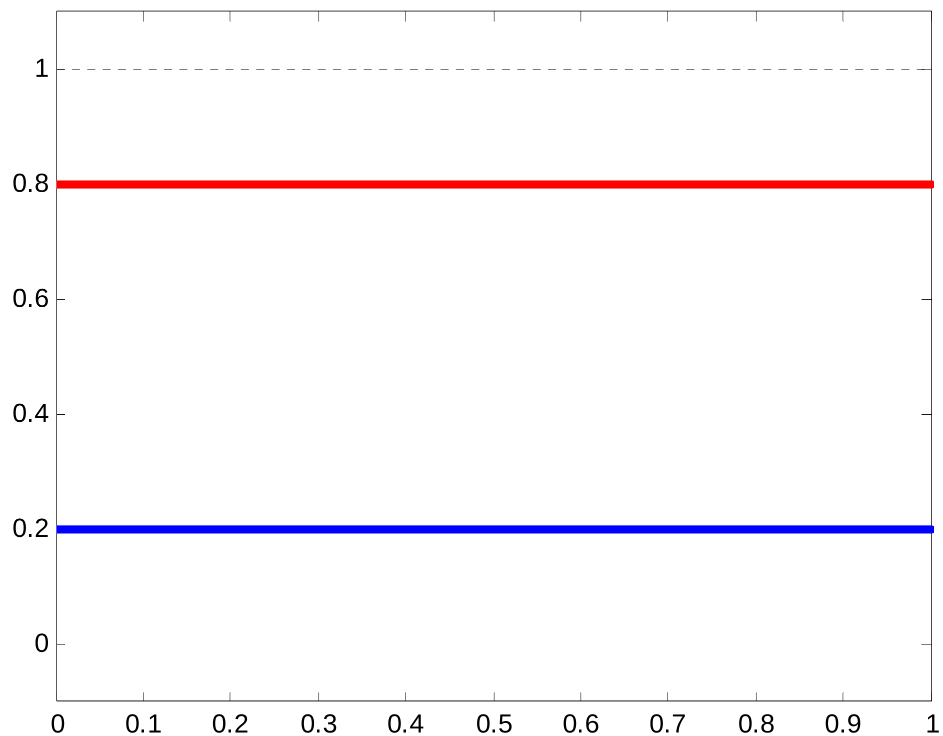
The Artificial Prediction Market

- A simulation of the Iowa Electronic Market:
  - Each class k = 1, ...,K corresponds to a contract type
  - Market price is a vector **c** = (c_1,..., c_K). We enforce $\sum c_k$=1
  - Contract for class k sells at market price $0<c_k<1$ and pays 1 if the outcome is k.
- A market participant is not a human, but a pair of:
  1. A budget (or weight) β_m
     - Based on past ability in predicting correct class
  2. A betting function $\phi(\mathbf{x}, \mathbf{c}) : \Omega \times [0, 1]^K \rightarrow [0, 1]^K$
  3. Percentage of the budget on each class a participant allocates.

Constant Betting Functions

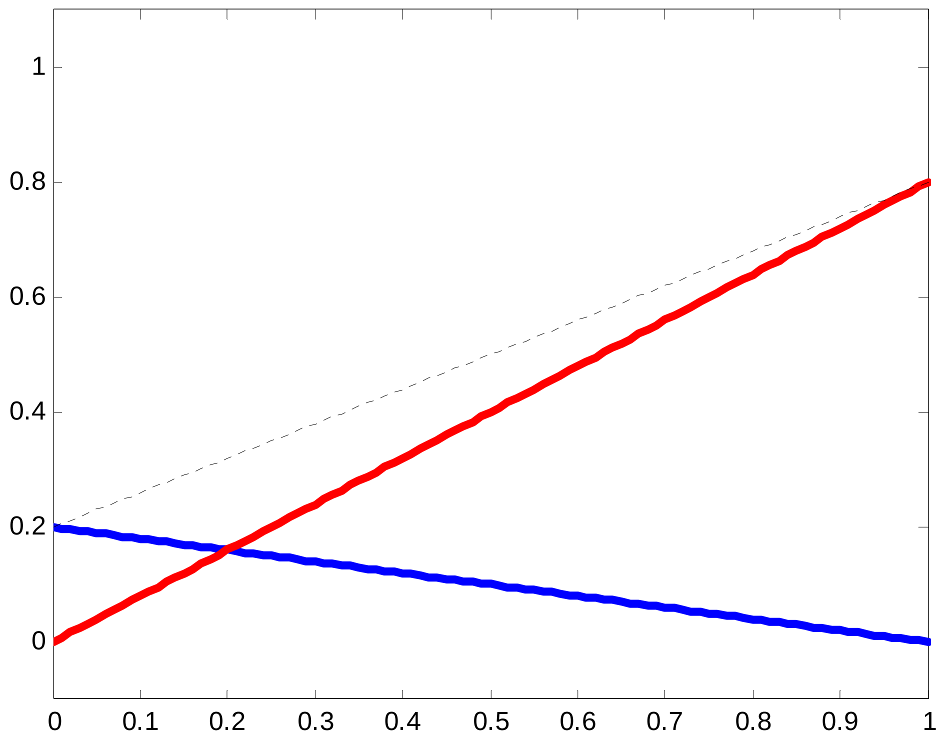
- Allocate same amount independent of the price

$$\phi_k(\mathbf{x}, \mathbf{c}) = h_k(\mathbf{x})$$

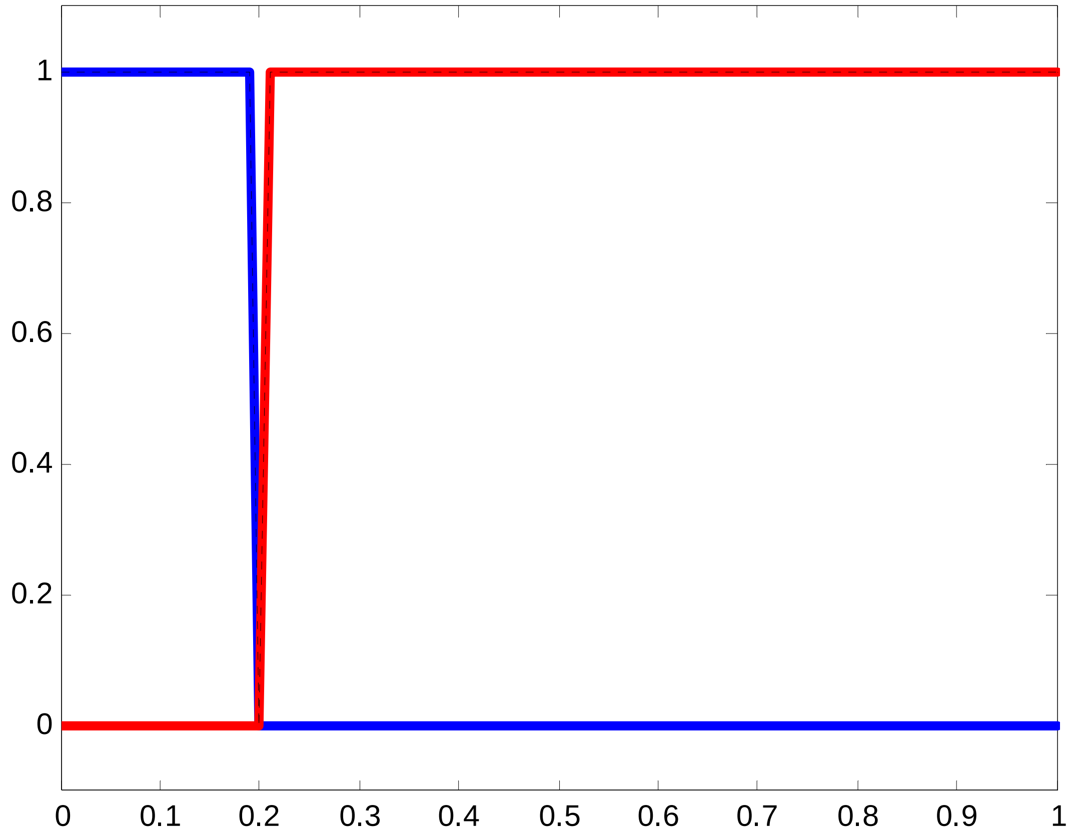


Linear Betting Functions

$$\phi_k(\mathbf{x}, \mathbf{c}) = (1 - c_k)h_k(\mathbf{x})$$

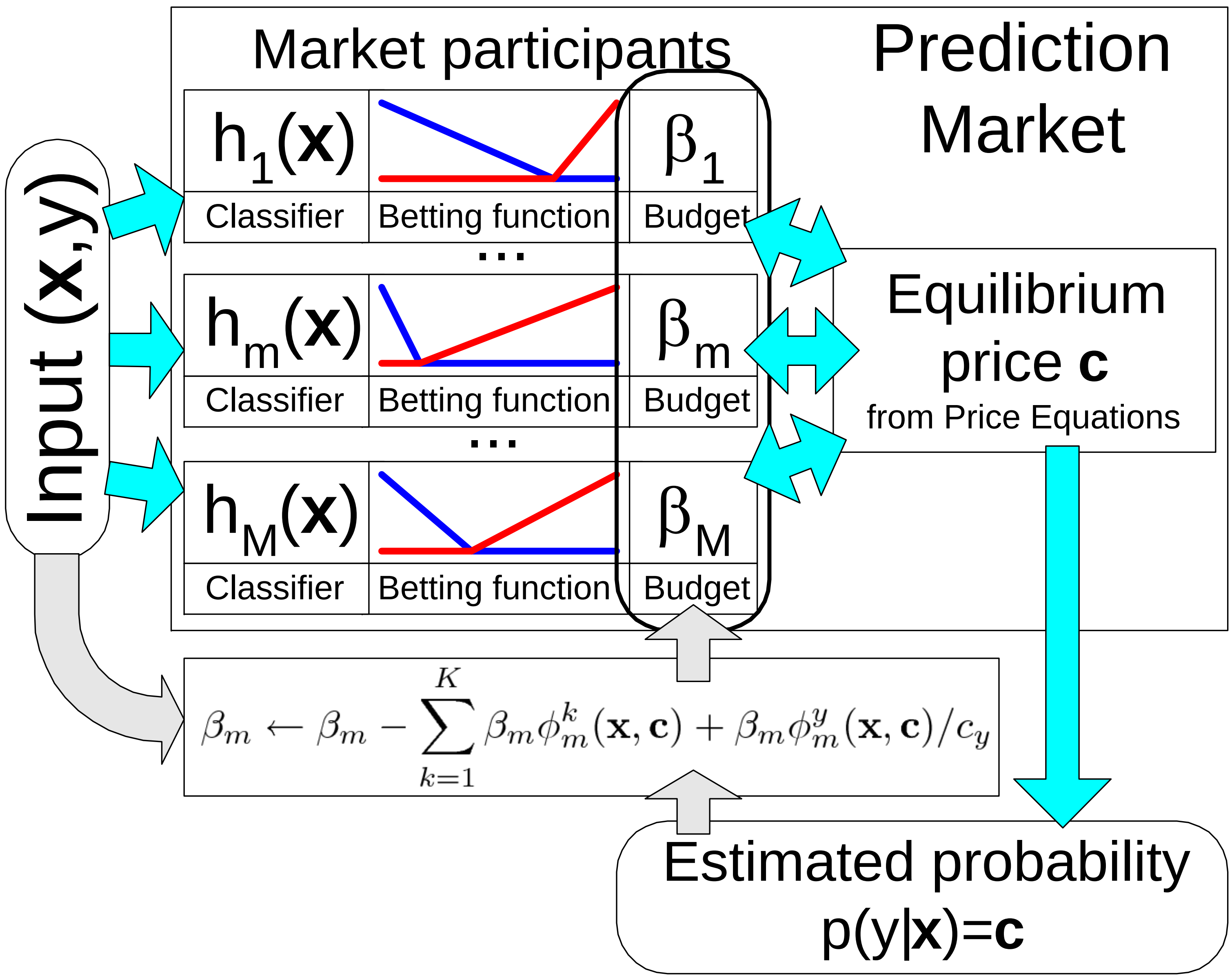


Aggressive Betting Functions



- Buy/sell based on classifier estimation of p(y|x)

Artificial Prediction Market Diagram



Market Update (x,y)

1. Compute equilibrium price **c** based on the price equations.
2. For each m=1,...,M
   - Update participant m's budget as

$$\beta_m \leftarrow \beta_m - \sum_{k=1}^{K} \beta_m \phi_{km}(\mathbf{x},\mathbf{c}) + \frac{\beta_m \phi_{ym}(\mathbf{x},\mathbf{c})}{c_y}$$

Price Equations

Main requirement:

- The total budget must remain the same after each market update, independent of the outcome y.
- This means:

$$\sum_{m=1}^{M} \sum_{k=1}^{K} \beta_m \phi_{km}(\mathbf{x}, \mathbf{c}) = \sum_{m=1}^{M} \frac{\beta_m \phi_{ym}(\mathbf{x}, \mathbf{c})}{c_y}$$

- This must hold for any y, since the market price c must depend only on x for prediction purposes.
- We also have

$$\sum_{k=1}^{K} c_k = 1$$

Solving the Price Equations

- Price Uniqueness

  If $\phi_k(\mathbf{x}, \mathbf{c})/c_k$ are monotonic, the price **c** is unique
- Holds for our betting functions.
- Solving the price equations
  - Analytically when possible:
    - For Constant Market
    - Two class linear market.
  - Numerically:
    - Double bisection method
    - Mann Iteration (faster)

Constant Betting is Linear Aggregation

- In the case of constant betting functions

$$\phi_k(\mathbf{x},\mathbf{c}) = h_k(\mathbf{x})$$

  we obtain linear aggregation of classifiers

$$\mathbf{c} = \frac{1}{\|\beta\|_1}\sum_{m=1}^{M}\beta_m h_m(\mathbf{x}) = \sum_{m=1}^{M}\alpha_m h_m(\mathbf{x})$$

  existent in Adaboost, Random Forest, etc.

- We obtain a new online learning rule for linear aggregation:

$$\beta_m \leftarrow \beta_m(1-\eta)) + \eta\|\beta\|_1 \frac{\beta_m h_{ym}(\mathbf{x})}{\sum_{i=1}^{M}\beta_i h_{yi}(\mathbf{x})}$$

Logistic Regression Market

- If $x \in R^M$, then picking the betting functions

$$\phi_{1m}(\mathbf{x}, 1-c) = (1-c)(x_m^+ - \log(1-c)) \quad x_m^+ = x_m I(x_m > 0)$$
$$\phi_{2m}(\mathbf{x}, c) = c(x_m^- - \log c) \quad x_m^- = x_m I(x_m \leq 0)$$

- Gives the price equilibrium equation

$$\sum_{m=1}^{M} \beta_m c(1-c)(x_m - \log(1-c) + \log c) = 0$$
$$\Rightarrow \log \frac{1-c}{c} = \sum_{m=1}^{M} \beta_m x_m = \mathbf{x}\beta$$

- Which gives the logistic regression model

$$p(Y=1|\mathbf{x}) = c = \frac{1}{1 + \exp\left(\sum_{m=1}^{M} \beta_m x_m\right)}$$

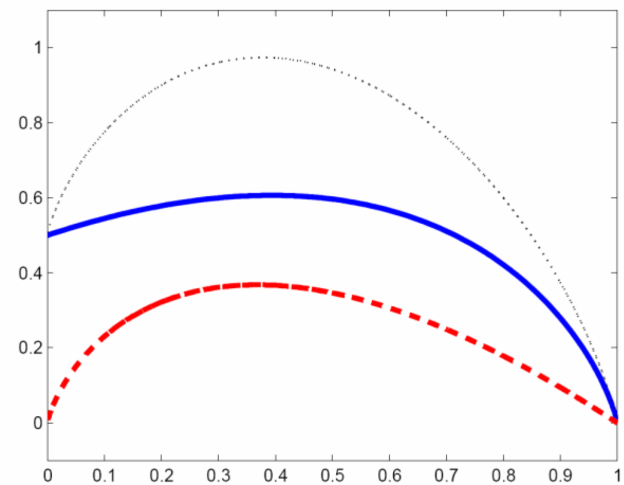
Logistic Regression Market Update

- This has the update rule that conserves $\sum_{m=1}^{M} \beta_m$

$$\beta_m^{t+1} = \beta_m^t - \eta \beta_m^t \left(x_m - \mathbf{x}\beta^t\right) \left(y - \frac{1}{1+\exp(\mathbf{x}\beta^t)}\right)$$

- It resembles the online logistic regression update rule

$$\beta_m^{t+1} = \beta_m^t - \eta x_m \left(y - \frac{1}{1+\exp(\mathbf{x}\beta^t)}\right)$$



An example of Logistic betting

Kernel Method for the Market

- Each instance x_i is a participant
- Each participant given as

$$\mathbf{h}_m(\mathbf{x}) = \begin{cases} \mathbf{e}_1 \cos\theta & \cos\theta \geq 0 \\ -\mathbf{e}_2 \cos\theta & \cos\theta < 0 \end{cases} \qquad \cos\theta = \frac{\mathbf{x}^T \mathbf{x}_m}{||\mathbf{x}||\,||\mathbf{x}_m||}$$

- Has decision boundary

$$h(\mathbf{x}) = \mathrm{sgn}\left(\sum_{m=1}^{M} \frac{\beta_m}{||\mathbf{x}_m||} (2y_m - 3)\mathbf{x}_m^T \mathbf{x} \right)$$

Kernel Method for the Market

- Decision boundary

$$h(\mathbf{x}) = \mathrm{sgn}\left(\sum_{m=1}^{M} \frac{\beta_m}{\|\mathbf{x}_m\|}(2y_m - 3)\mathbf{x}_m^T\mathbf{x}\right)$$

- Can use the RBF Kernel Trick for nonlinear boundaries
  - No margin though

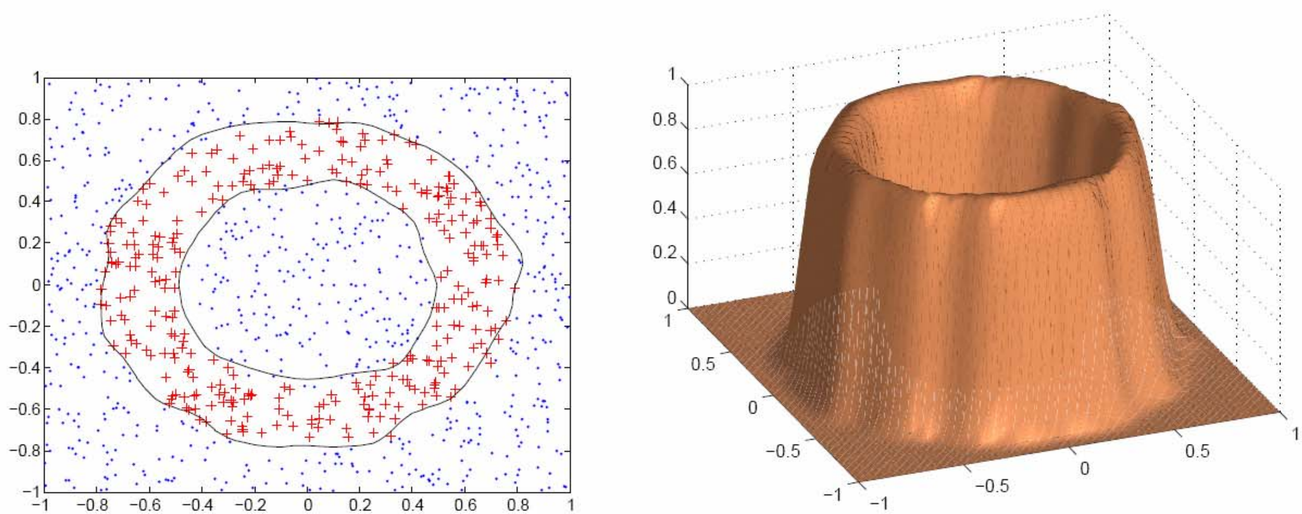


Figure 3: Left: 1000 training examples and learned decision boundary (right) for an RBF kernel-based market from eq. (8) with $\sigma = 0.1$. Right: estimated probability function.

Maximum Likelihood

- The Constant Market maximizes the log likelihood

$$L(\beta) = \frac{1}{N}\sum_{n=1}^{N} \log c_{y_n}(\mathbf{x}_n;\beta)$$

- The update

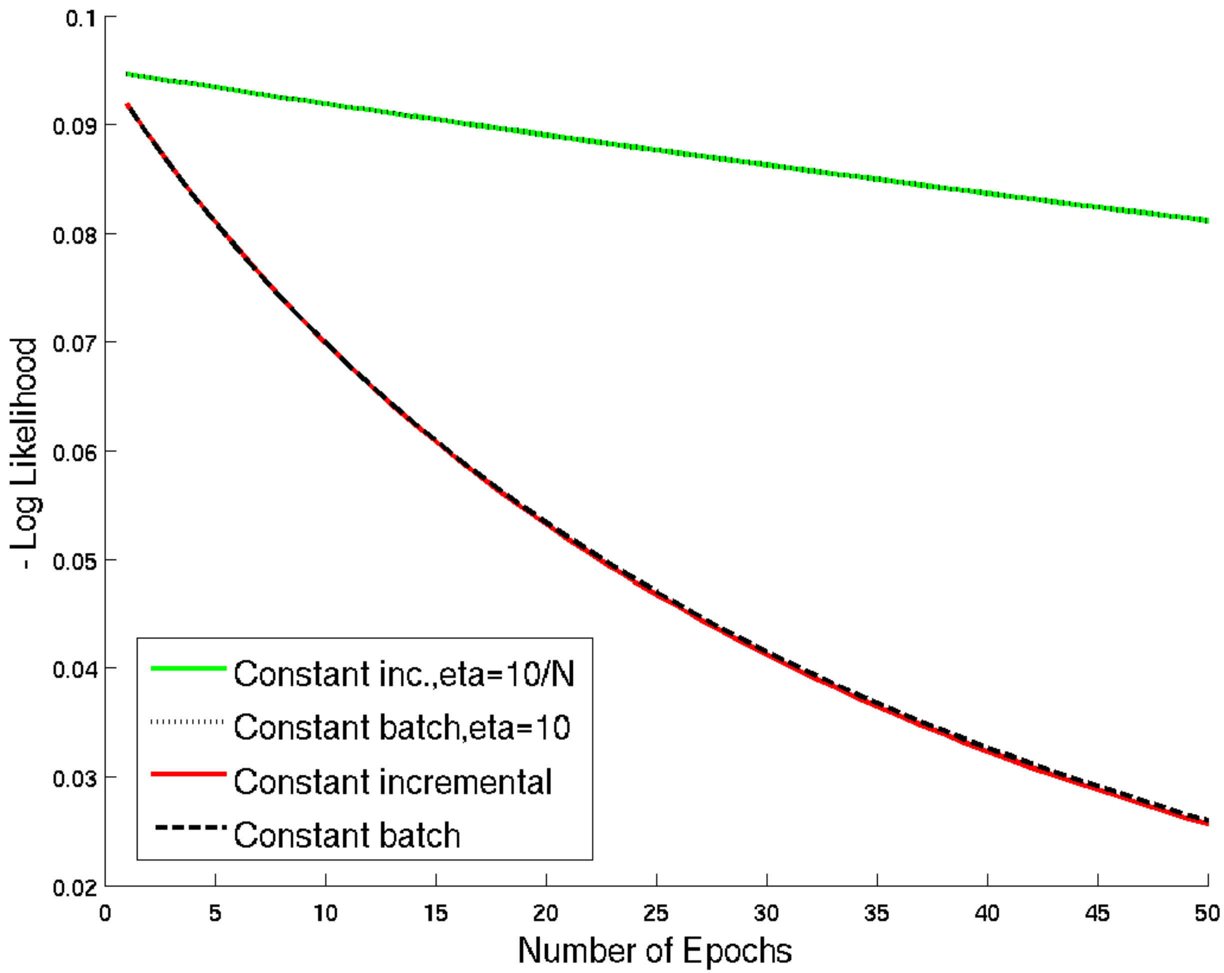
$$\beta_m^{t+1} = (1-\eta)\beta_m^t + \eta\beta_m^t \frac{1}{N}\sum_{n=1}^{N} \frac{h_{y_n m}(\mathbf{x}_n)}{c_{y_n}(\mathbf{x}_n;\beta^t)}$$

can be viewed as a gradient ascent on L(β)

- The Market update is stochastic gradient ascent

$$\beta_m^{t+1} = \left(1 - \frac{1}{N}\eta\right)\beta_m^t + \frac{1}{N}\eta\beta_m^t \frac{h_{y_n m}(\mathbf{x}_n)}{c_{y_n}(\mathbf{x}_n;\beta^t)}$$

Batch vs Incremental Market Updates



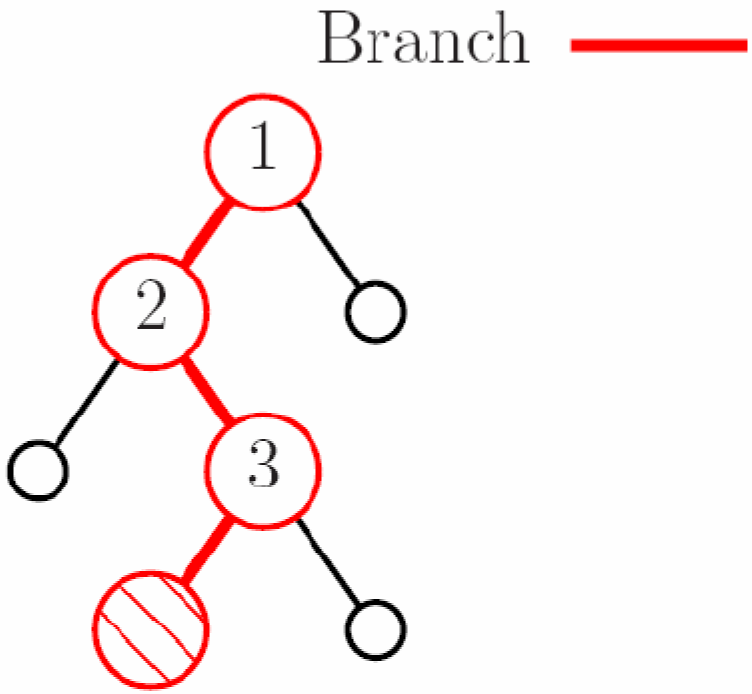
Loss functions for both the batch and Market (incremental) updates.

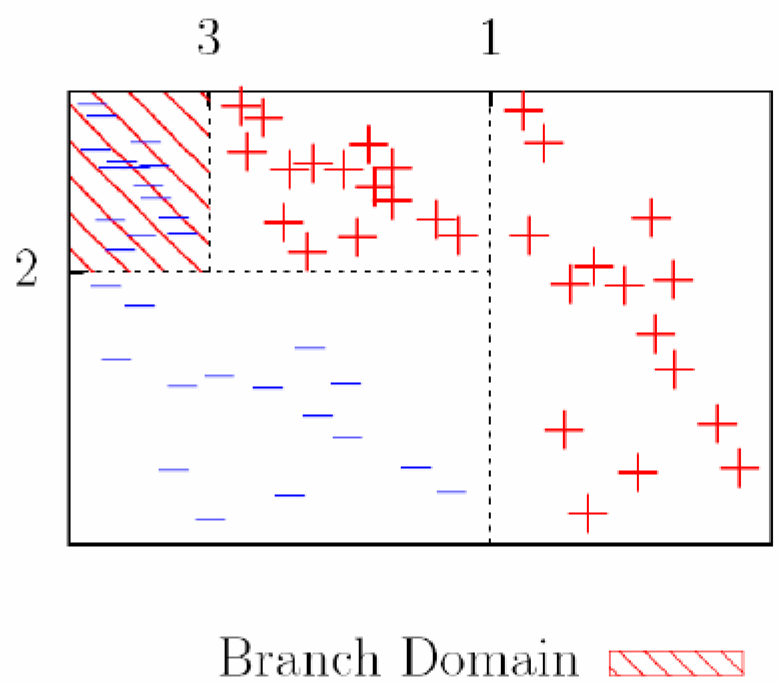
Specialization

- In Boosting and Random Forrest, all classifiers are aggregated for any observation $x \in \Omega$.
- The Market participants can be specialized
  - A participant can predict very well on a subregion of Ω.
  - It will not bet on any x outside its region.
  - For each observation, a different subset of classifiers could participate in betting
  - Example: a leaf node of a random tree

Decision Tree Rules as Specialized Classifiers

- Decision tree rules (leaves) can perfectly classify training data in their specialized domain.





Real Data Results

- 21 datasets from the UC Irvine Machine Learning repository
  - Many are small (≈ 200 examples).
  - Training and test sets are randomly subsampled, 90% for training and 10% for testing.
  - Exceptions are satimage and poker datasets with test sets of size 2000 and 10^6 respectively
- All results are averaged over 100 runs.
- Significance comparison tests ($\alpha<0.01$):
  - Mean differences from RF results from Breiman'01
  - Paired t-tests with our RF implementation

Results on UCI Data

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.

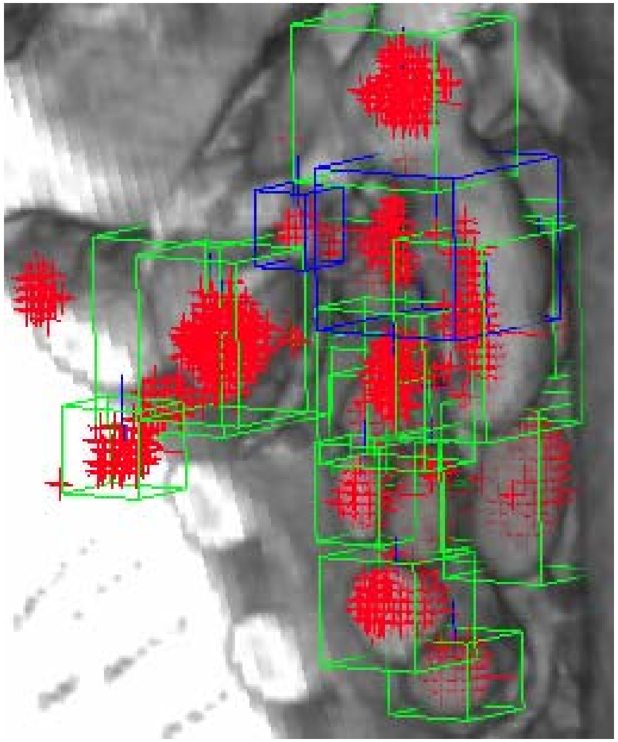
Data	Train Size	Test Size	Feat.	Cls	ADB	RFB	RF	CB	LB	AB
cancer	699	–	9	2	3.2	2.9	3.0	2.9	2.9	2.9
sonar	208	–	60	2	15.6	15.9	14.8	14.1	14.3	14.1
vowel	990	–	10	11	4.1	3.4	3.3	3.1 •	3.2	3.1 •
diabetes	768	–	8	2	26.6	24.2	23.4	23.4	23.4	23.5
ecoli	336	–	8	8	14.8	12.8	13.1	13.0	13.0	13.1
german	1000	–	20	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 •

- ADB and RFB are Adaboost and Random Forest from Breiman’01
- CB and AB perform best and significantly outperform RF in many cases
- Trained markets never performed significantly worse than RF

Application: Lymph Node Detection

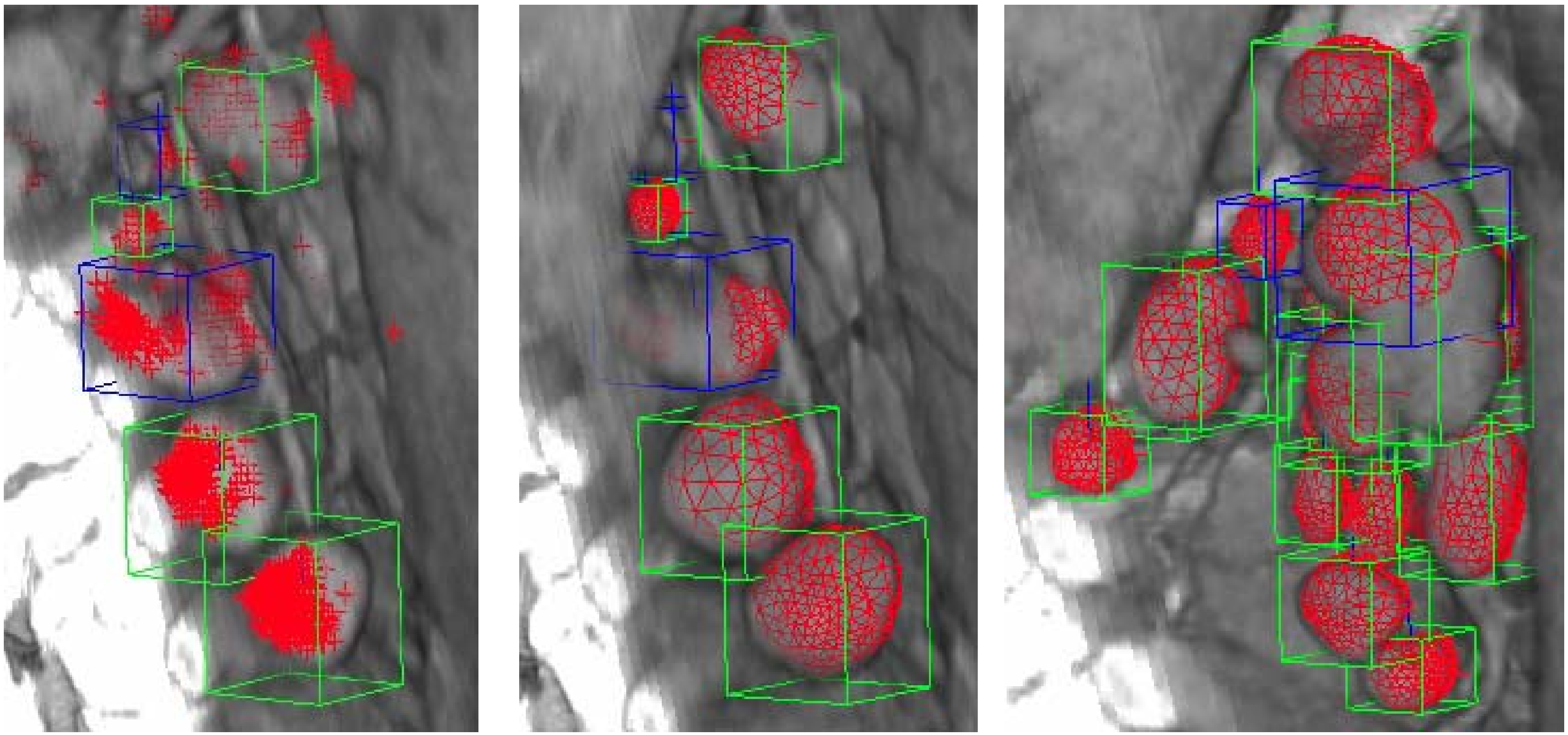
- About 2000 candidate lymph node centers are obtained with a trained detector (Barbu et al, 2012)
- At each candidate, a segmentation is obtained
- From each segmentation

  17000 features are extracted
- ~30 are selected by Adaboost



Detected lymph node candidates

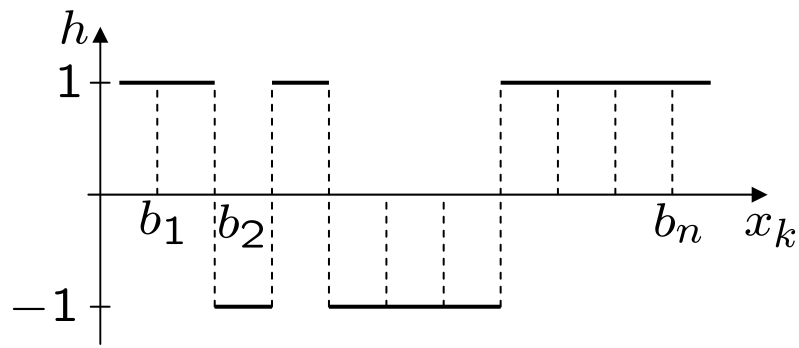
Example Axillary Region



Detected LN candidates | Detected Lymph Nodes | Detected Lymph Nodes

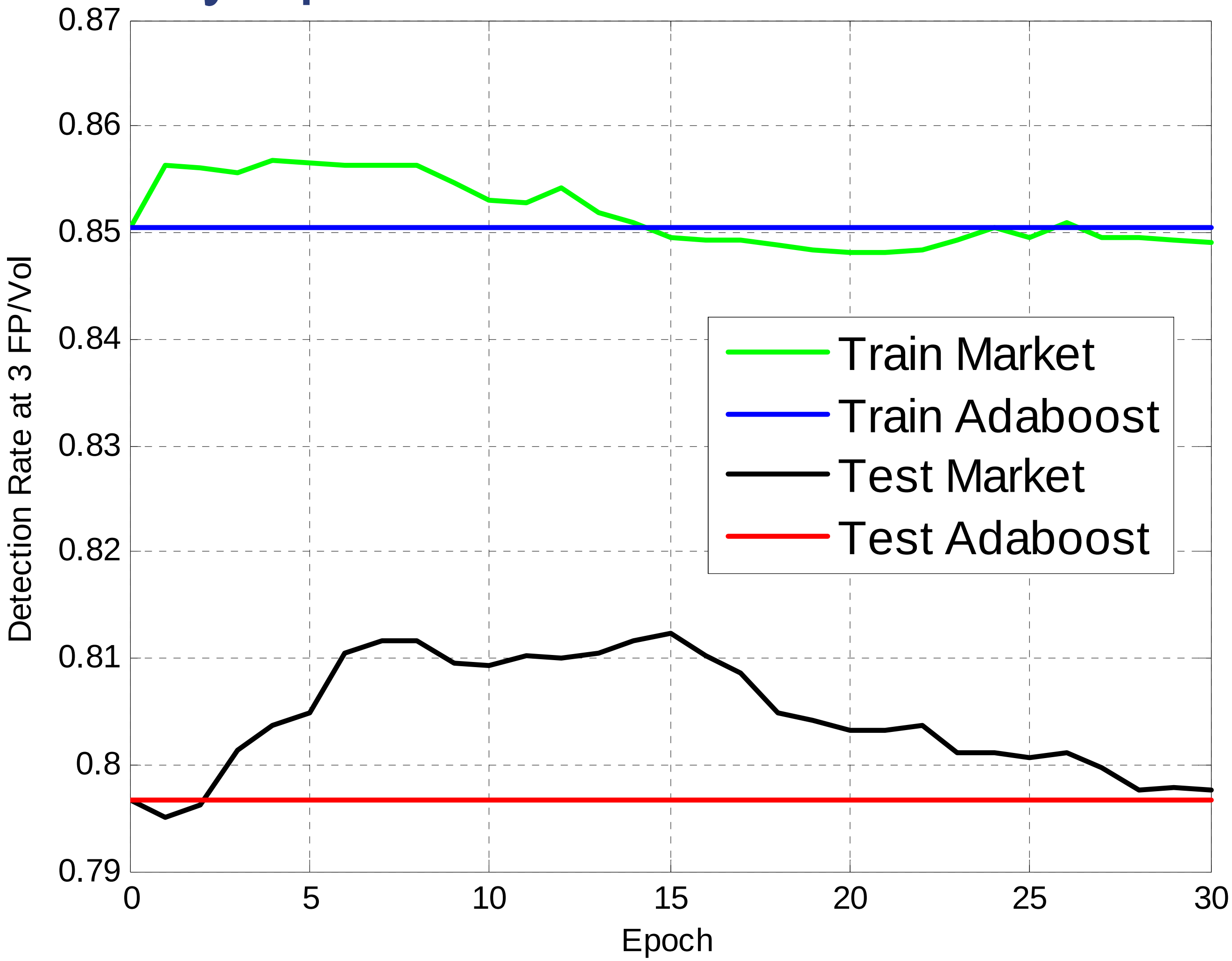
A Market of Classifier Bins

- Adaboost is based on histogram classifiers with 64 bins



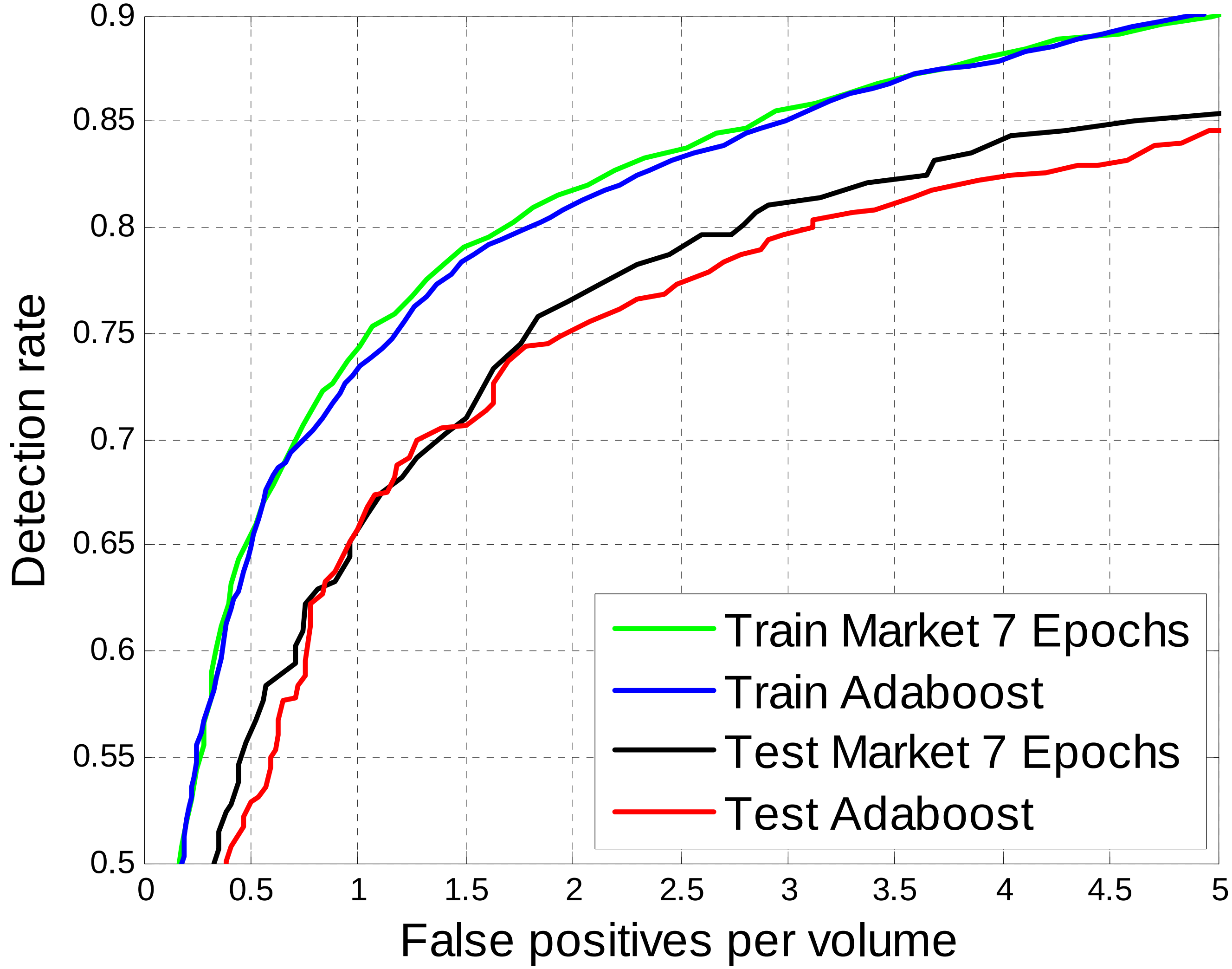
- Converted to Constant Market
  - Each bin is a specialized participant bidding for one class
  - Initial budgets are the Adaboost coefficients
  - Totally 2048 participants
  - Weighted update with $w_+=0.5/N_+$, $w_- =0.5/N_-$

Lymph Node Detection Results



- Detection rate at 3FP/vol (clinically acceptable)
- Six fold cross-validation

Lymph Node Detection Results



- Market performance at 7 epochs
- *p*-value 0.028

The Regression Market

- Extend class labels to have "uncountably many" labels
- Participants' bets and prices become conditional densities
- Equilibrium price and updates generalize
- As with Classification Market, it maximizes log likelihood and minimizes an approximation of the $E[KL(p(y|x),c(y|x;\beta)]$.

The Regression Market

- The proportion of the budget spent on contracts for “class” y at price $c(y|\mathbf{x};\boldsymbol{\beta})$ is $h(y|\mathbf{x})$
- The number of contracts purchased for y is

$$n_m(y) = \beta_m \frac{h_m(y|\mathbf{x})}{c(y|\mathbf{x};\boldsymbol{\beta})}$$

- Introduce reward kernel $K(t;y)$ that rewards for “almost” correct predictions (e.g. Gaussian, Dirac Delta).

$$\text{winnings} = \int_Y K(t;y) n_m(t) dt$$

Constant Betting Update Rule

- This gives the update rule:

$$\beta_m \leftarrow \beta_m + \eta\beta_m \left(\int_Y K(t;y) \frac{h_m(t|\mathbf{x})}{c(t|\mathbf{x};\boldsymbol{\beta})} dt - 1 \right)$$

- η caps the total proportion bet
- This prevents instantaneous bankruptcies (i.e. $\beta = 0$)
- η is also the learning rate.

Constant Betting Update Rule: Delta Update

- When $K(t;y) = \delta(t-y)$

$$\beta_m \leftarrow \beta_m + \eta\beta_m \left(\frac{h_m(y|\mathbf{x})}{c(y|\mathbf{x})} - 1 \right)$$

- Same update rule as classification market.
- Still improves aggregation but prone to overfitting.

Constant Betting Update Rule: Gaussian Update

- When

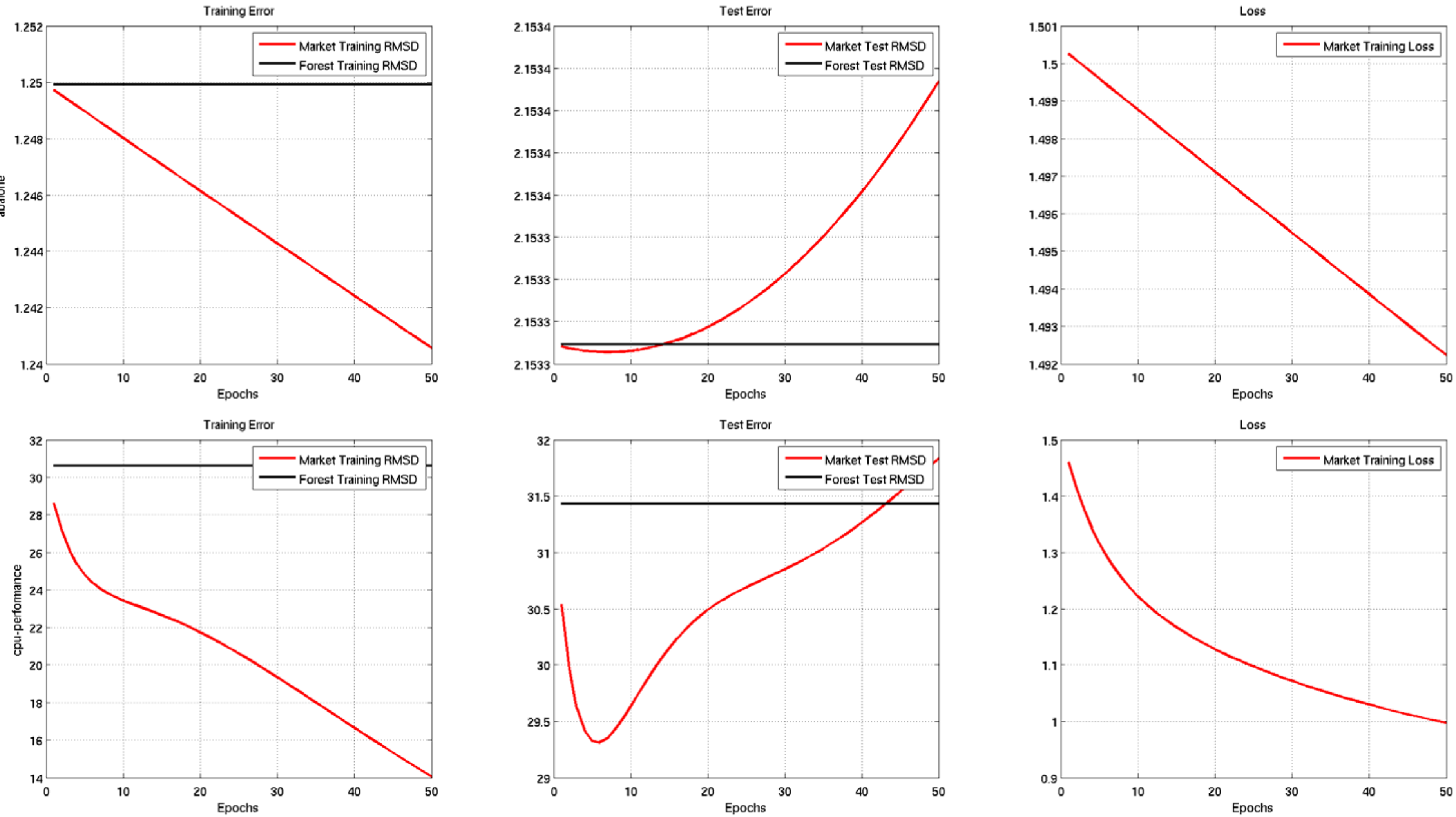
$$K(t;y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-y)^2}{2\sigma^2}}$$

- Have to evaluate an integral. Use Gaussian-Quadrature.

$$\beta_m \leftarrow \beta_m + \eta\beta_m \left(-1 + \frac{1}{\sqrt{\pi}} \sum_{i=1}^{n} \omega_i \frac{h_m(y + \sqrt{2}\sigma t_i | \mathbf{x})}{c(y + \sqrt{2}\sigma t_i | \mathbf{x})} \right)$$

- $t_i,\ \omega_i$ are the Hermite-Gauss nodal points and weights.
- σ should reflect noise level of training data.

Loss Examples



- Training, test RMSD and loss for abalone and cpu-performance data sets

Real Data Results

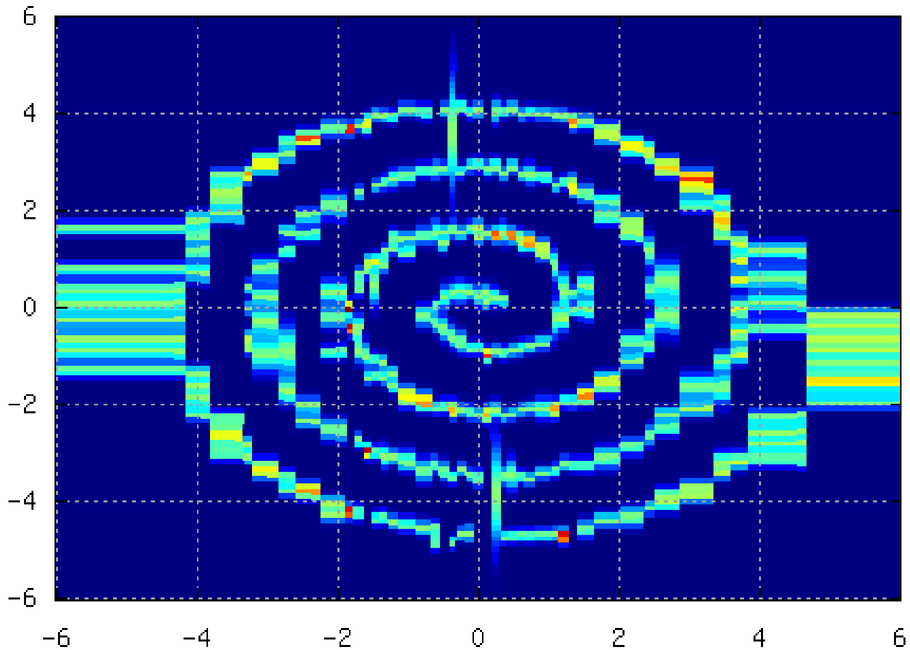
Table 1. Table of MSE for forests and markets on UCI and LIAAD data sets. The F column is the number of inputs, Y is the range of regression, RFB is Breiman's reported error, RF is our forest implementation, DM is the Market with delta updates, and GM is the Market with Gaussian updates. Bullets/daggers represent pairwise significantly better/worse than RF while +/– represent significantly better/worse than RFB.

Data	N_{train}	N_{test}	F	Y	RFB	RF	DM	GM
abalone	4177	–	8	$[1.00, 29.00]$	4.600	4.571	4.571	4.571
friedman1	200	2000	10	$[4.30, 26.03]$	5.700	4.343+	4.335•+	4.193•+
friedman2	200	2000	4	$[-167.99, 1633.87]$	19600.0	19431.852	19232.482•	18369.546•+
friedman3	200	2000	4	$[0.13, 1.73]$	0.022	0.028–	0.028•–	0.026•–
housing	506	–	13	$[5.00, 50.00]$	10.200	10.471	10.130•	10.128•
ozone	330	–	8	$[1.00, 38.00]$	16.300	16.916	16.925	16.917
servo	167	–	4	$[0.13, 7.10]$	0.246	0.336	0.295	0.322
ailerons	7154	6596	40	$[-0.00, -0.00]$	–	2.814e-008	2.814e-008•	2.814e-008•
auto-mpg	392	–	7	$[9.00, 46.60]$	–	6.469	6.444	6.405•
auto-price	159	–	15	$[5118.00, 35056.00]$	–	3823550.43	3723413.430	3815863.98
bank	4500	3693	32	$[0.00, 0.67]$	–	7.238e-003	7.212e-003•	7.210e-003•
breast cancer	194	–	32	$[1.00, 125.00]$	–	1112.270	1112.509	1108.325
cartexample	40768	–	10	$[-12.69, 12.20]$	–	1.233	1.233†	1.232•
computeractivity	8192	–	21	$[0.00, 99.00]$	–	5.414	5.398•	5.414†
diabetes	43	–	2	$[3.00, 6.60]$	–	0.415	0.426†	0.415
elevators	8752	7847	18	$[0.01, 0.08]$	–	9.319e-006	9.288e-006•	9.225e-006•
forestfires	517	–	12	$[0.00, 1090.84]$	–	5834.819	5844.493†	5680.131•
kinematics	8192	–	8	$[0.04, 1.46]$	–	0.013	0.013•	0.013•
machine	209	–	6	$[6.00, 1150.00]$	–	3154.521	2991.798•	3042.336
poletelecomm	5000	10000	48	$[0.00, 100.00]$	–	29.813	28.855•	29.863†
pumadyn	4499	3693	32	$[-0.09, 0.09]$	–	9.237e-005	8.917e-005•	8.888e-005•
pyrimidines	74	–	27	$[0.10, 0.90]$	–	0.013	0.013	0.012
triazines	186	–	60	$[0.10, 0.90]$	–	0.015	0.015	0.015

- RFB is Regression Forest from Breiman'01
- GM, DM perform best and significantly outperforms RF in most cases

Clustering Regression Tree

- Want to "regress" multimodal responses (e.g. circle).
- Generalize Regression Tree to cluster Y values
- Use Market to "weed out" poorly clustered branches of a forest.



- A single clustering regression tree on the spiral data.

Conclusion

A theory for Artificial Prediction Markets based on the Iowa Electronic Market:

- Aggregate classifiers, regressors, and densities.
- Very simple update rules.
- Logistic Regression and Kernel methods.
- Can be used for both online and offline learning.
- Significantly outperforms Random Forest in many cases, in both prediction and probability estimation.

Future Work

- Generalization error and VC dimension of the Market
- Feature (participant) selection
- Learning betting functions
- Regression Market applications in Computer Vision and Medical Imaging
- Other types of Market participants

References

- N. Lay, A. Barbu. Supervised Aggregation of Classifiers using Artificial Prediction Markets. ICML 2010
- N. Lay, A. Barbu. The Artificial Regression Market. Arxiv.
- A. Barbu, M. Suehling, X. Xu, D. Liu, S. K. Zhou, D. Comaniciu. Automatic Detection and Segmentation of Lymph Nodes from CT Data. IEEE Trans Medical Imaging, 31, No. 2, 240–250, 2012.
- A. Barbu, N. Lay. An Introduction to Artificial Prediction Markets. JMLR 2012