Supervised Aggregation Using Artificial Prediction Markets

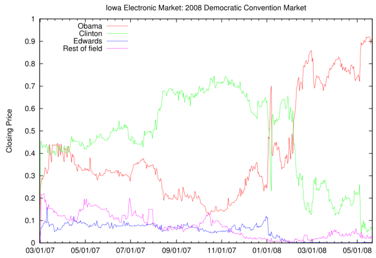
Nathan Lay

Advisor: Adrian Barbu, Department of Statistics

Co-Advisor: Anke Meyer-Baese, Department of Scientific Computing

Prediction Markets

- ➢Forum where contracts are traded on future outcomes.
- ➢Contracts pay contingent on the outcome.
- ➢Trading price of contracts reflects combined knowledge and experience of participants.
- ➢Trading price is an estimator of the probability.
- ➢Can predict outcomes of elections, sporting events, and foreign affairs.
- ➢Were demonstrated to be more accurate than polling or individual experts.

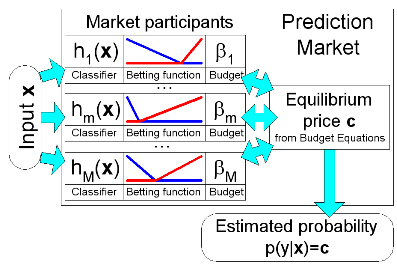


Trading prices of contracts on democratic nominees for the 2008 presidential election.

Overview

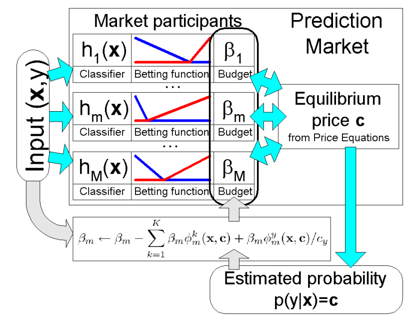
Idea

- ➢Reinterpret events as *instances*, future outcomes as instance *labels*, and participants as *classifiers*, *regressors* or *densities*.
- ➢For each instance, classifiers "purchase" contracts for each possible label.
- ➢The trading price is a probability estimate for the instance.



Learning

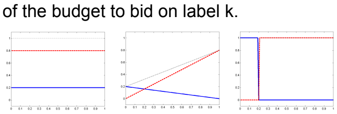
- ➢Each participant is allotted a budget.
- ➢Each participant bids for contracts and are rewarded based on *correct* prediction.
- ➢Budgets describe the prediction accuracy of each participant.
- ➢The goal is to learn the budget configuration that improves the market's prediction accuracy.



Classification

Overview

- ➢*Events* are instances **x**, and the *outcomes* are discrete labels y 2 { 1,2, … K}.
- ➢Participants are *betting functions* $\phi^k(\mathbf{x},\mathbf{c})$ and allot a proportion of the budget to bid on label k.



Three examples of betting functions: Constant, Linear, and Aggressive from left to right respectively.

Equilibrium

- ➢Equilibrium price conserves the budget sum for each update
- ➢Estimates the true conditional mass p(y|**x**)

$$c_k(\mathbf{x}) = \frac{1}{n}\sum_{m=1}^{M} \beta_m \phi_m^k(\mathbf{x},\mathbf{c}) \qquad n = \sum_{m=1}^{M} \beta_m \sum_{k=1}^{K} \phi_m^k(\mathbf{x},\mathbf{c})$$

Update Rule

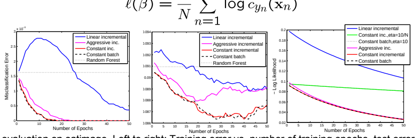
- ➢Sequential update for each instance **x** and label y.

$$\beta_m \leftarrow (1-\eta)\beta_m + \eta\beta_m \frac{\phi_m^y(\mathbf{x},\mathbf{c})}{c_y(\mathbf{x})}$$

Loss Function

- ➢The update rule maximizes the average log likelihood
- ➢Minimizes an approximation of the expected KL divergence

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



Example evaluation on satimage. Left to right: Training error vs. number of training epochs, test error vs. number of training epochs and negative log-likelihood function vs. number of training epochs.

Results

- ➢Real data sets are from UCI repository. There are 30 total.
- ➢Participants are random tree branches from a random forest.

Data	N_{train}	N_{test}	F	K	ADB	RFB	RF	CB	LB	AB
breast-cancer	683	–	9	2	3.2	2.9	2.7	2.7	2.7	2.7
sonar	208	–	60	2	15.6	15.9	18.1	17	17.4	17
vowel	990	–	10	11	4.1	3.4	4.2	3.6 •	3.9 •	3.4 •
ecoli	336	–	7	8	14.8	12.8	14.5	14.3	14.4	14.3
german	1000	–	24	2	23.5	24.4	23.7	**23.3**	23.3	**23.3**
glass	214	–	9	6	22	20.6	22	21.9	21.9	21.8
image	2310	–	19	7	1.6	2.1	2.1	**1.8** •	**1.8** •	**1.8** •
ionosphere	351	–	34	2	6.4	7.1	6.5	6.2	6.4	6.3
letter-recognition	20000	–	16	26	3.4	3.5	3.3	3.2 •	3.2 •	3.2 •
liver-disorders	345	–	6	2	30.7	25.1	26.5	26.5	26.5	26.6
pima-diabetes	768	–	8	2	26.6	24.2	24.4	24.3	24.2	24.3
satimage	4435	2000	36	6	8.8	8.6	9.1	8.8 •	8.9 •	8.8 •
vehicle	846	–	18	4	23.2	25.8	24.3	**23.6**	**24.2**	**23.6**
voting-records	232	–	16	2	4.8	4.1	4.1	4.1	4.1	4.1
zipcode	7291	2007	256	10	6.2	6.3	6.1	**6.2** †	**6.1**	**6.1**
abalone	4177	–	8	3	–	–	44.7	44.7	44.6	44.7
balance-scale	625	–	4	3	–	–	14	14.1	14.1	14.5 †
car	1728	–	6	4	–	–	2.5	0.9 •	1.2 •	0.9 •
connect-4	67557	–	42	3	–	–	19.9	16.7 •	16.9 •	16.7 •
cylinder-bands	277	–	33	2	–	–	22.5	22.7	22.5	22.5
hill-valley	606	606	100	2	–	–	45.1	44.4 •	44.8 •	44.5 •
isolet	1559	–	617	26	–	–	7.6	7.4	7.5	7.4 •
king-rk-vs-king	28056	–	6	18	–	–	21.6	11.0 •	11.8 •	11.0 •
king-rk-vs-k-pawn	3196	–	36	2	–	–	1.2	0.4 •	0.5 •	0.4 •
magic	19020	–	10	2	–	–	12.0	11.7 •	11.8 •	11.8 •
madelon	2000	–	500	2	–	–	31.2	23 •	23.1 •	23 •
musk	6598	–	166	2	–	–	2.2	1.1 •	1.2 •	1.1 •
splice-junction-gene	3190	–	59	3	–	–	4.6	4.1 •	4.2 •	4.1 •
SAheart	462	–	9	2	–	–	31.2	31.3	31.3	31.3
yeast	1484	–	8	10	–	–	37.8	37.9	37.9	37.7

The prediction market and random forest were trained and tested on 100 random samples with 90% of each data set used for training and 10% used for testing. Satimage (2000), zipcode (2007), and hill-valley (606) provide test sets. The table provides the misclassification rates for Breiman's Adaboost (ADB), Breiman's Random Forest (RFB), our Random Forest (RF), Constant Betting (CB), Linear Betting (LB), and Aggressive Betting (AB).

Regression

Overview

- ➢*Events* are instances, and the *outcomes* are real numbers
- ➢Like classification, but with uncountably many *labels*
- ➢Participants are conditional densities h(y|**x**)

Equilibrium

- ➢Equilibrium price conserves the budget sum for each update
- ➢Estimates the true conditional density p(y|**x**)

$$c(y|\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(y|\mathbf{x})$$

Update Rule

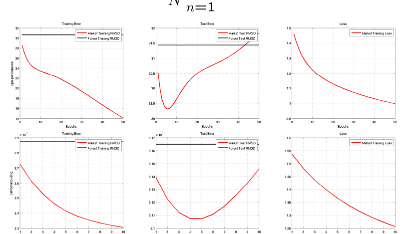
- ➢Sequential update for each instance **x** and label y.

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

Loss Function

- ➢The update rule maximizes the average log likelihood
- ➢Minimizes an approximation of the expected KL divergence

$$\ell(\beta) = \frac{1}{N}\sum_{n=1}^{N} \log c(y_n|\mathbf{x}_n)$$



(Top) Training error, test error, and negative log likelihood for the cpu-performance data set.
(Bottom) Training error, test error, and negative log likelihood for the californiahousing data set.

Results

- ➢Real data sets are from UCI and LIACC repository. There are 24 total.
- ➢Participants are regression tree branches from a regression forest.

Data	N_{train}	N_{test}	F	Y	RFB	RF	CB
abalone	4177	–	8	[1.00, 29.00]	2.14	2.15	2.15
activity	8191	–	21	[0.00, 99.00]	–	2.52	**2.50**
auto-mpg	392	–	7	[9.00, 46.60]	–	2.72	**2.72**
bodyfat	252	–	17	[0.00, 45.10]	–	1.44	**1.27**
californiahousing	20639	–	8	[14999.00, 500001.00]	–	51647.93	**51072.33**
cart	40767	–	10	[−12.69, 12.20]	–	1.05	*1.08*
concrete-slump	103	–	9	[17.19, 58.53]	–	4.10	**3.81**
concrete-strength	1030	–	8	[2.33, 82.60]	–	5.51	**5.18**
cpu-performance	209	–	7	[15.00, 1238.00]	–	31.43	**29.31**
forestfires	517	–	12	[0.00, 1090.84]	–	52.40	*53.09*
friedman	40767	–	10	[−1.23, 30.52]	–	1.38	**1.36**
gala	30	–	5	[2.00, 444.00]	–	70.36	67.96
house-price-16H	22783	–	16	[0.00, 500001.00]	–	31906.65	**31817.26**
housing	506	–	12	[5.00, 50.00]	3.19	3.24	3.24
ozone	330	–	9	[1.00, 38.00]	4.04	3.93	3.93
pima	768	–	8	[0.08, 2.42]	–	0.33	*0.33*
pole	4999	9999	48	[0.00, 100.00]	–	9.70	**6.45**
prostate	97	–	8	[−0.43, 5.58]	–	0.77	0.77
pumadyn-32nm	4498	3692	32	[−0.09, 0.09]	–	0.02	**0.02**
servo	167	–	4	[0.13, 7.10]	0.50	0.55	0.55 †
star	47	–	1	[3.94, 6.29]	–	0.33	**0.32**
uswages	2000	–	9	[50.39, 7716.05]	–	390.21	390.20
wine-red	1599	–	10	[3.00, 8.00]	–	0.58	**0.57**
wine-white	4898	–	10	[3.00, 9.00]	–	0.62	**0.60**

The prediction market and random forest were trained and tested on 100 random samples with 90% of each data set used for training and 10% used for testing. Pole (9999) and pumadyn-32nm (4498) provide test sets. The table provides RMSD errors of Breiman's regression forest (RFB), Our implementation of regression forest (RF), and constant Regression Market (CT). Bold/italic mean significantly better/worse than corresponding RF test errors. Dots/daggers mean significantly better/worse than RFB test errors.

Density Estimation

Overview

- ➢Not intuitively a prediction market
- ➢Based on regression market
- ➢Participants are densities h(**x**)

Equilibrium

- ➢Equilibrium price conserves the budget sum for each update
- ➢Estimates the true density p(**x**)

$$c(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x})$$

Update Rule

- ➢Sequential update for each instance **x**

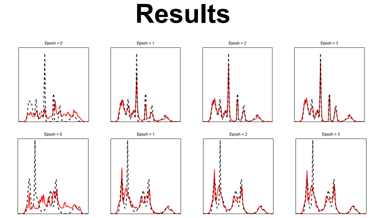
$$\beta_m \leftarrow (1-\eta)\beta_m + \eta\beta_m \frac{h_m(\mathbf{x})}{c(\mathbf{x})}$$

Loss Function

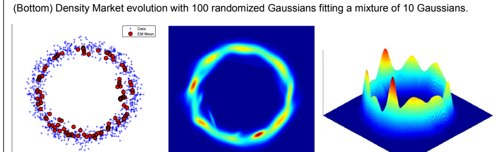
- ➢The update rule maximizes the average log likelihood
- ➢Minimizes an approximation of the KL divergence

$$\ell(\beta) = \frac{1}{N}\sum_{n=1}^{N} \log c(\mathbf{x}_n)$$

Results



(Top) Density Market evolution with 100 Gaussians with the 10 true Gaussians fitting a mixture of 10 Gaussians.
(Bottom) Density Market evolution with 100 randomized Gaussians fitting a mixture of 10 Gaussians.



Left to right: The circle data with corresponding inferred EM Gaussian means, an intensity plot of the trained Density Market viewed from above, and a 3D view of the trained Density Market.

References

[1] J. Wolfers and E. Zitzewitz. Prediction markets. Journal of Economic Perspectives, pages 107–126, 2004.
[2] K. J. Arrow, R. Forsythe, M. Gorham, R. Hahn, R. Hanson, J. O. Ledyard, S. Levmore, R. Litan, P. Milgrom, and F. D. Nelson. The promise of prediction markets. Science, 320(5878):877, 2008.
[3] L. Breiman. Random forests. Machine Learning, 45(1):5–32, 2001.
[4] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth, Belmont, California, 1984.
[5] J. Perols, K. Chari, and M. Agrawal. Information Market-Based Decision Fusion. Management Science, 55(5):827–842, 2009.
[6] C.F. Manski. Interpreting the predictions of prediction markets. Economics Letters, 91(3):425–429, 2006.
[7] C.R. Plott, J. Wit, and W.C. Yang. Parimutuel betting markets as information aggregation devices: Experimental results. Economic Theory, 22(2):311–351, 2003.