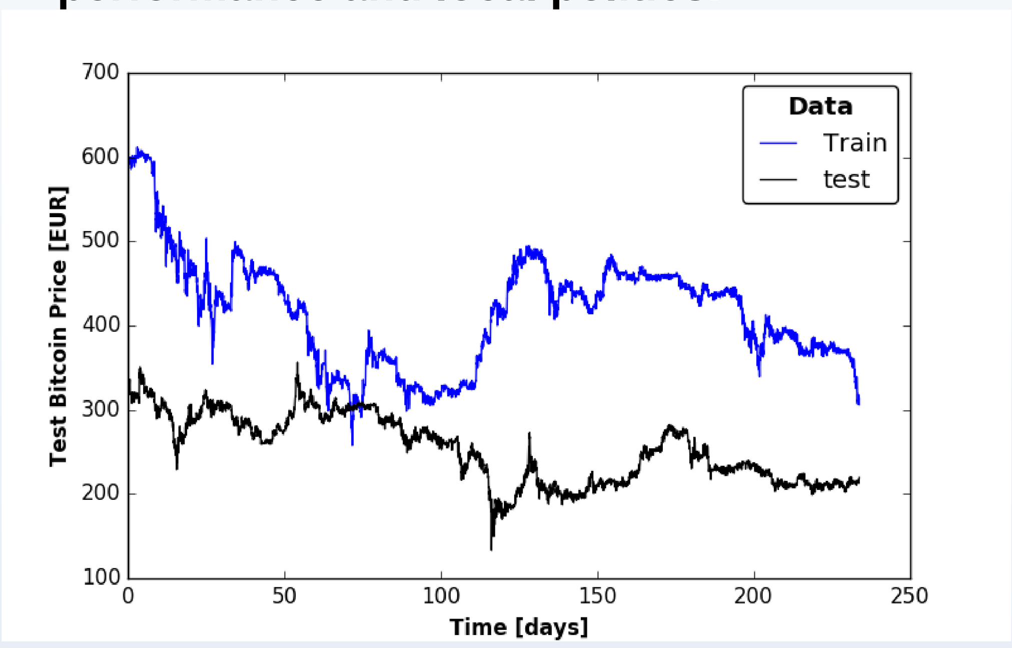
Comparative Automated Bitcoin Trading

Kareem Hegazy and Samuel Mumford
CS229: Machine Learning
Stanford University

Data: Why Trade Bitcoin?

- Bitcoin is a peer to peer traded crypto-currency.
- From market data [1] we created a record of bitcoin price for two consecutive 240 day periods in 500 second bins.
- Bitcoin is an attractive system for testing trading algorithms as it is both volatile and somewhat insulated from company performance and local politics.



Features

We used the first 5 left derivatives of the price as data features. Adding more derivatives and adding in other metrics such as the number of bitcoins sold proved to perform no better than this simple feature set.

Trading Approach

- For each model, the prediction of price change is converted to a number *c* ranging from -1 to 1.
- If *c* > 0, convert *c**(current assets not in bitcoin) into bitcoin.
- If *c* < 0, sell |c|*(currently held bitcoin).
- A fee of .25% may be applied to all assets liquidated or bought.
- RRL trades by making (1 + *c*)/2 of all assets into bitcoin at each timestep.
- The "Weighted Confidence" metric is the ratio of final asset value to initial value without a fee. It is a better analyzer of how well the algorithm is making predictions on the market than profit with fee.

Models

- **Linear Regression**: Perform locally weighted linear regression, and predict bitcoin price change based on the local slope.
- **Logistic Regression**: Fit the sign of bitcoin price change at time *i + 1* given derivatives *d* to a function of the form:

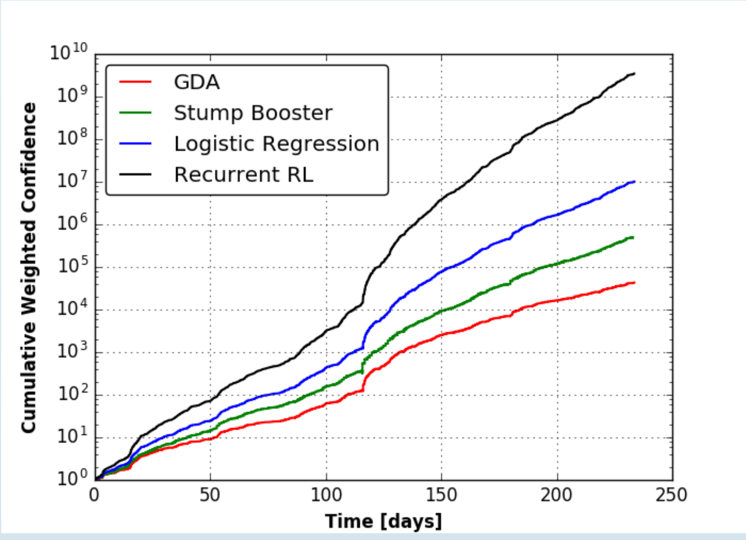
$$g(\theta^T d) = \frac{2}{1 - e^{\theta^T d}} - 1.$$

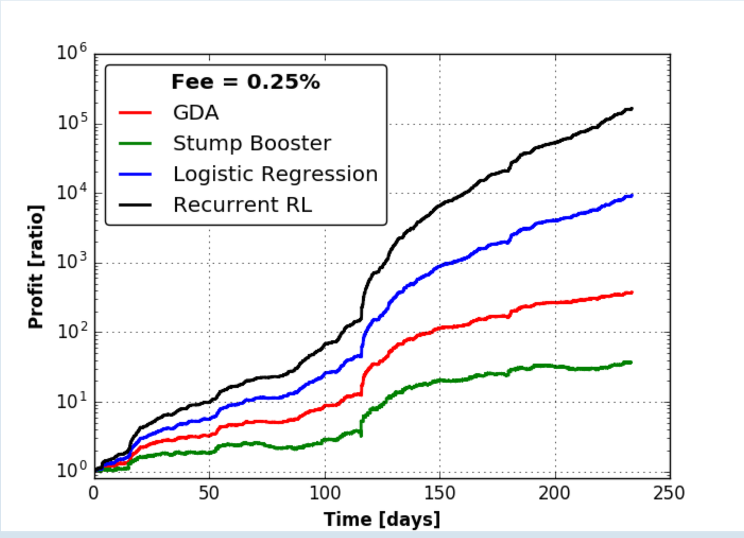
  - Fit for θ through stochastic gradient descent.
- **GDA**: Assume derivatives *d* are drawn from two independent Gaussian distributions based on if price will increase or decrease, G_1 and G_{-1}. After calculating the mean and covariance of these Gaussians we make a prediction for a test set point based on the relative probability of d_{test} belonging to G_1 and G_{-1}.
- **Stump Boosting**: Define discriminator $\phi(i, d, T)$ as returning ± 1 based on if the i^{th} derivative is > *T*. Generate many ϕ and calculate exponential-based weights to make a composite prediction function. Make final predictions with this function.
- **Decision Tree**: Define $\Phi(Is\ d, Ts)$ as returning ± 1 based on a binary decision tree. The tree branches are created based on a list of feature indexes Is and thresholds *Ts*. To make the p^{th} order branch, split based on if *d(Is[p]) > Ts[p]*.
- **Recurrent Reinforcement Learning (RRL)**: RRL directly maximizes profit ratio at time *t, P(t)*. For a ratio of total assets in bitcoin at time *t* denoted by $F_t(\theta_t, d_t)$, trading fee δ, and $r_t = \frac{Bitcoin\ Price\ at\ t}{Bitcoin\ Price\ at\ t-1} - 1$,

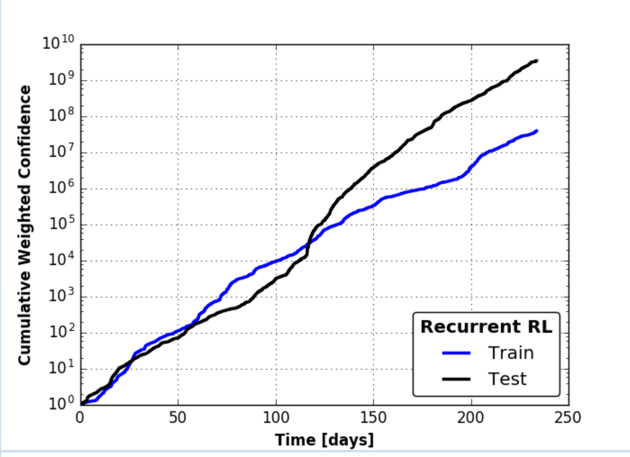
$$P(t) = \Pi_0^t (F_{t-1} r_t + 1)(1 - \delta|F_t - F_{t-1}|) = \Pi_0^t (1 + R_t).$$

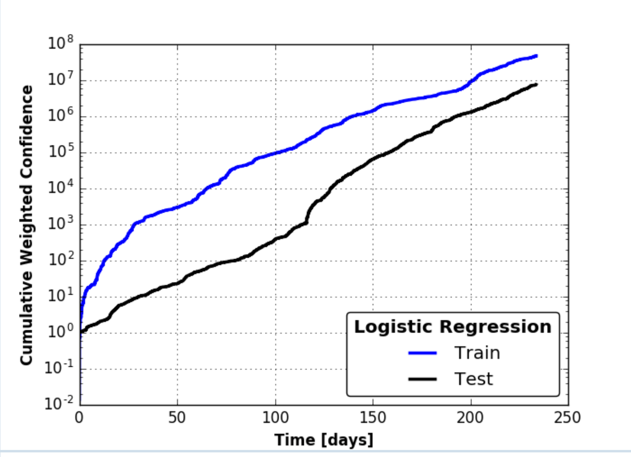
Storing values of $\frac{dF_t}{d\theta}$ for all timebins before the current time analyzed and taking *ln(P)*, *P(t)* can be maximized through online learning-style gradient ascent on θ [2][3][4].

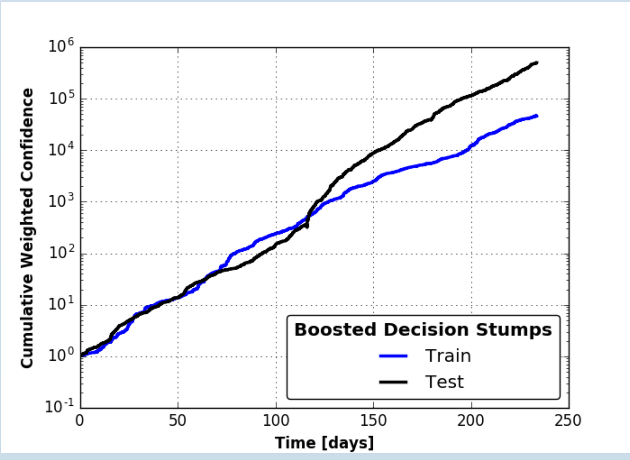
Performance Plots

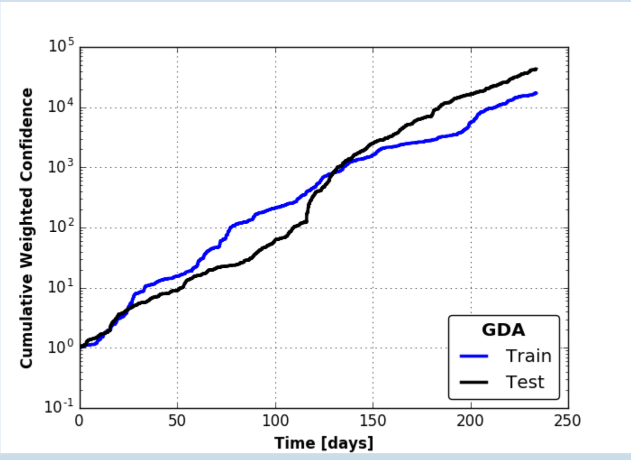












Summary of Results

Metric	Correct Slope Sign Rate		Log Weighted Confidence		Log Profit Ratio	
	Train	Test	Train	Test	Train	Test
Logistic Regression	0.8180	0.8227	7.7434	6.9907	4.6472	3.9999
GDA	0.8172	0.8216	4.6347	4.2361	2.0455	2.5740
RRL	0.7713	0.7746	7.5959	9.5394	3.7026	5.2122
Stump Booster	0.7739	0.7764	4.6657	5.7000	0.4115	1.5689
Tree Booster	0.7237	0.7171	0.7241	0.7815	0.4377	0.5075
Local Linear Regression	0.4943	0.4943	-5.9309	-5.9309	-8.8112	-8.8112
Best Alg. for Metric:	Logistic Reg	Logistic Reg	Logistic Reg	RRL	Logistic Reg	RRL

Viewing the results of the table above, it is clear that logistic regression was most successful in terms of maximizing the target metric of guessing the correct sign of bitcoin price changes. Unsurprisingly, GDA performs comparably to logistic regression, as it is a less general version of logistic regression. RRL performs poorly in terms of predicting the sign of price change, but does well for maximizing profit, as expected since its target metric is profit. Finally, the baseline local linear regression model loses money precipitously.

Conclusion and Outlook

Given the high ratio of correct predictions and extremely high profit ratios of logistic regression, GDA, and RRL, we accomplished the main project goal of making effective bitcoin price predictors. For all models other than RRL the ratio of correct predictions is strongly correlated with final profit. This demonstrates that our simple target works well as an indicator for future trading performance and maximizing profit. RRL also behaves differently than the other metrics because it is directly maximized for profit, instead of predicting the sign of bitcoin price changes.

Outlook

In light of the absurdly high profit ratios, it is clear that we are not effectively simulating market behavior. The main problematic assumption is that any trade we offer will be accepted, i.e. that we can buy any amount or sell any amount of bitcoin at any time at "market price". This is clearly not true, and to proceed we would have to model the probability of trades being accepted to make a more realistic trading simulator. Unfortunately, data allowing us to judge the market depth is not available.

It could also be illustrative to make a boosting algorithm which boosts based on the six algorithms we used thus far. Furthermore, we considered a reinforced learning algorithm to choose which algorithm to trade with.

References

[1] http://api.bitcoincharts.com/v1/csv/
[2] Moody, John, and Matthew Saffell. "Reinforcement Learning for Trading." *NIPS* (n.d.): 918-23. Web. 11 Dec. 2016.
[3] Molina, Gabriel. *Stock Trading with Recurrent Reinforcement Learning (RRL)*. Stanford U, CS229 Application Project, n.d.
[4] Du, Xin, Jinjian Zhai, and Koupin Lv. "Algorithm Trading Using Q-Learning and Recurrent Reinforcement Learning." (n.d.): n. pag.*Http://cs229.stanford.edu/proj2009/LvDuZhai.pdf*. Stanford University. Web. 11 Dec. 2016.