Network Inference
Using Gradient Boosted Decision Trees

Kevin Emmett*[1] and Sakellarios Zairis†[2]

[1]Department of Physics, Columbia University
[2]Department of Computational Biology & Bioinformatics, Columbia University

Tuesday 17th September, 2013

0 Summary

We first train gradient boosting regression trees on the time series data and then derive the network connectivity from the most frequently selected features over the boosting rounds. We also incorporate a certain degree of prior biological knowledge into the adjacency matrix.

1 Methods

1.1 Background

The general gradient boosting tree algorithm is as follows [2]:

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \cdots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}} \tag{1}$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \cdots, J_m$.
   (c) For $j = 1, 2, \cdots, J_m$ compute
   $$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma) \tag{2}$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$

*kje2109@columbia.edu
†siz2102@columbia.edu

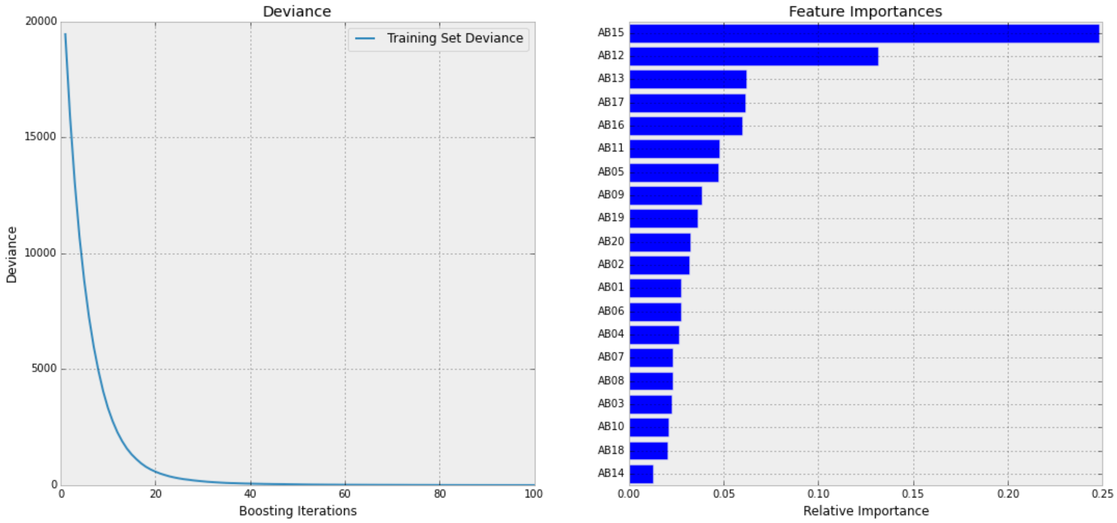

Figure 1: Training the GBR model for insilico dataset, AB15. Left: Training set deviance for GBR model as function of boosting round. Right: Feature importance rankings.

The salient features of this algorithm are (a) it builds a strong classifier from an ensemble of weaker ones, (b) it re-weights the training data at each iteration to emphasize hitherto incorrectly classified examples, and (c) the final classifier provides not only the *sign* but also its margin, or distance from zero, as a measure of confidence in the prediction.

All feature weights are rescaled to the interval [0, 1] and are used to populate the adjacency matrix. In this approach, therefore, if antibody X was repeatedly used to build the predictive function for antibody Y, then we assume that X is a parent of Y in the network.

1.2 Implementation

Our models were written in python. We used pandas to manipulate the data prior to training, and implemented our models using scikit-learn.

1.3 Parameter Tuning

Gradient Boosted Decision trees have the following parameters:

- `num_estimators`: number of rounds of boosting rounds to perform.
- `max_depth`: maximum depth of the base learners.
- `learning_rate`: the learning rate shrinks the contribution of each tree by the specified value.

We used cross validation to set the best values of the parameters for each trained model.

1.4 Inhibitor/Stimulus Modeling

Inhibitors were modeled using a perfect fixed-effects model [4]. The stimulus was not explicitly modeled. We dealt with stimulii in two ways: by grouping datasets across stimulii, and by training independent models for each stimulus. We found training separate models for each stimulus performed better.

1.5 Experimental vs inSilico

For subchallenge 1, we used the same general approach in both 1A and 1B. The only key difference is our use of a curated network prior in 1A. The prior is discussed further in the External Information section below.

1.6 Any other approaches considered?

Our first approach to modeling time series were based on symbolic regression using the genetic algorithm package Eureqa. While this approach was promising, it was computationally expensive. We also implemented a simple Dynamic Bayesian Network, following [3]. We found the DBN overfit when used to predict time courses. Before settling on the GBR model, we tried a Lasso regularized linear model, which we found performed slightly worse under cross validation.

1.7 Computational Resources

Our algorithms ran on our local machines (Intel Core i7, 8GB RAM), typically taking no more than 5 minutes per run.

2 Data Preparation

For the experimental data, we partitioned the data into independent subsets for each cell line and stimulus combination (`cell_line`, `stim`). We modeled inhibitors using a perfect fixed-effects model [4]. We used only the "Main" dataset in training our models.

For the insilico data, the dataset was partitioned into independent subsets for each stimulus. Inhibitors were again modeled using a perfect fixed-effects model.

For each dataset, we centered and scaled the columns before training the model. We did not use log-transformed data.

3 External Information

We implemented two literature models of dynamic signaling networks: a 16 node ERK pathway [5] and a 5 node yeast network [1]. We used these as gold standard networks to prototype our model. In these datasets, we had a known network structure which we used to simulate synthetic time courses.

A thorough literature review for canonical cancer signaling mechanisms was also performed. This yielded a set of network edges which the authors bias toward in the inferred adjacency matrix for 1A. This prior represents the basic understanding of RTK signaling pathways.

4 Model Validation

We used k-fold cross validation to test our model performance (k=5, k=10, LOO). We did not model randomness in our cross-validation.

5 Leader Board

We found the leader board had limited utility for subchallenge 1 because of the lack of a clearly defined scoring function. The main use we had for the leader board was in calibrating our network prior's weighting.

6 Discussion

We implemented code to use the gradient of the concentration as the output variable, but didn't get a chance to fully test it. Additionally, we plan to continue experimenting with different post-processing logic for combining adjacency matrices derived from different stimulus/inhibitor conditions. We are pleased nonetheless at our fairly successful application of a powerful method like gradient tree boosting to the domain of network inference where it appears under-utilized. We anticipate strong performance of this approach as the number of phosphoproteins reported on RPPA chips increases and the danger of overfitting in other methods increases.

7 Author Backgrounds and Statements

Kevin Emmett is a PhD student in physics with a background in discriminative modeling of emerging pathogens. Sakellarios Zairis is an MD/PhD student with a background in applying supervised learning techniques to predicting viral oncogenic potential. These authors contributed equally to the work, with an average weekly time commitment of 15 hours.

References

[1] Irene Cantone, Lucia Marucci, Francesco Iorio, Maria Aurelia Ricci, Vincenzo Belcastro, Mukesh Bansal, Stefania Santini, Mario Bernardo, Diego Bernardo, and Maria Pia Cosma. Supplemental Data A Yeast Synthetic Network for In Vivo Assessment of Reverse-Engineering and Modeling Approaches. *Cell*, 137:1–34, 2009.

[2] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 1. 2004.

[3] S M Hill, Y Lu, J Molina, L M Heiser, P T Spellman, T P Speed, J W Gray, G B Mills, and S. Mukherjee. Bayesian Inference of Signaling Network Topology in a Cancer Cell Line. *Bioinformatics*, 28(21):2804–2810, October 2012.

[4] Simon E F Spencer, Steven M Hill, and Sach Mukherjee. Dynamic Bayesian networks for interventional data. (12):1–17, 2012.

[5] Tian-Rui Xu, Vladislav Vyshemirsky, Amélie Gormand, Alex von Kriegsheim, Mark Girolami, George S Baillie, Dominic Ketley, Allan J Dunlop, Graeme Milligan, Miles D Houslay, and Walter Kolch. Inferring signaling pathway topologies from multiple perturbation measurements of specific biochemical species. *Science signaling*, 3(113):ra20, January 2010.