


Tweets and NCAA March Madness

David Freed and Samuel Green

{davidfreed,samuelgreen}@college.harvard.edu

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

Real-Time Win Probability Modeling with Twitter

This project investigates the predictive power of Twitter data in modeling the outcomes and events in NCAA March Madness basketball games. We extend previous work that used only pools of data collected in advance of games to make single-period predictions about outcomes, instead collecting live-action time series of tweets in-game. We find that the volumes of relevant tweets far outpace the tweets collected in previous work and that tweet volumes appear to coincide with in-game events. **Using basic models for online learning, we conclude that tweets can be used to improve over standard logistic regression models in making predictions about game events and outcomes.**

Data Collection

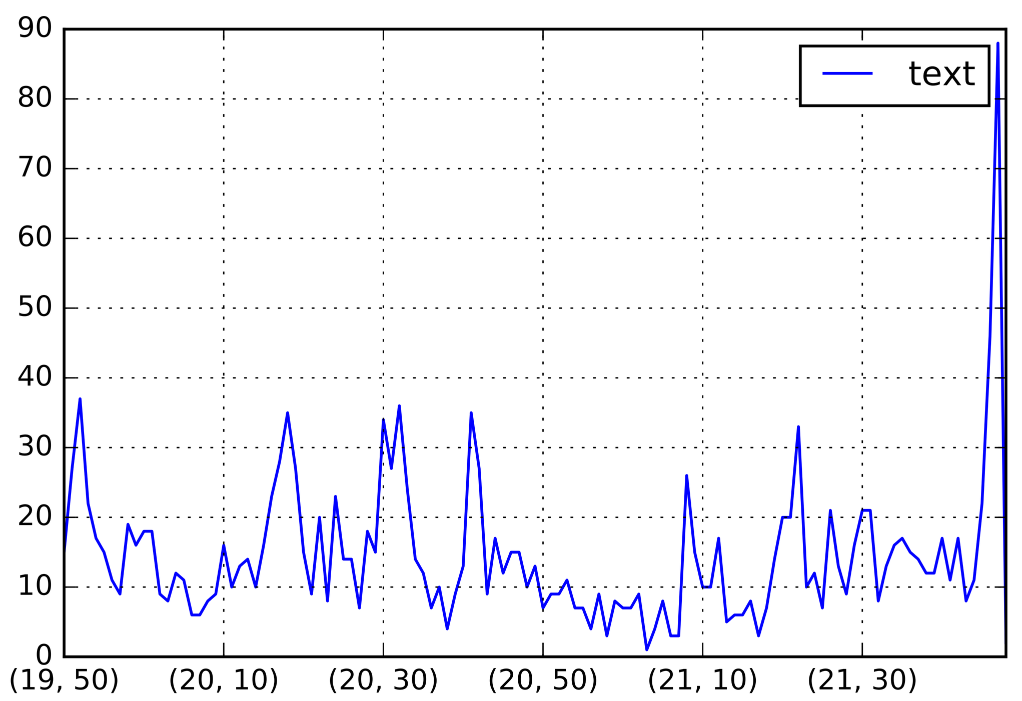
Our dataset compiles approximately 1 million tweets from 42 games during the NCAA March Madness. For each game, we compiled hashtags relevant to the game, which we then used to collect tweets via the Streaming API. An set of example hashtags is:

#MarylandvsHawaii #WeWill #Maryland
#Terrapins #RainbowWarriors #Hawaii
#HawaiiMBB #RoadWarriors #GoBows

Summary statistics for the entire dataset are:

N	Mean	St. Dev.	Min	Max
42	20,754.810	27,273.890	1,930	171,600

Volumes Track Game Events



Tweets per Minute, Oregon vs. Holy Cross

Sentiment Analysis

Sentiment classification for tweets forms a central pivot point for this project. To perform classification, we trained a simple bag-of-words random forest classifier against a labeled training corpus of 4000 tweets. While the classifier ended up treating many tweets as neutral or irrelevant because of heterogeneity between the training set and the tweets requiring classification, the simple classifier we trained provided adequate classification power to test the potential usefulness of the Twitter data.

Relevance Analysis

Sentiment in this setting is much less meaningful without relating the sentiment contained in a tweet to one of the two teams. We therefore perform some heuristic classification of the relevance of a tweet to each team.

We based on the classification off of the hashtags that were originally used to collect the tweets and off of rosters. To treat rosters uniformly, we included that listed last names for each of the 7 top players by playing time for each team, as well as the last name of the head coach, with all data from ESPN.

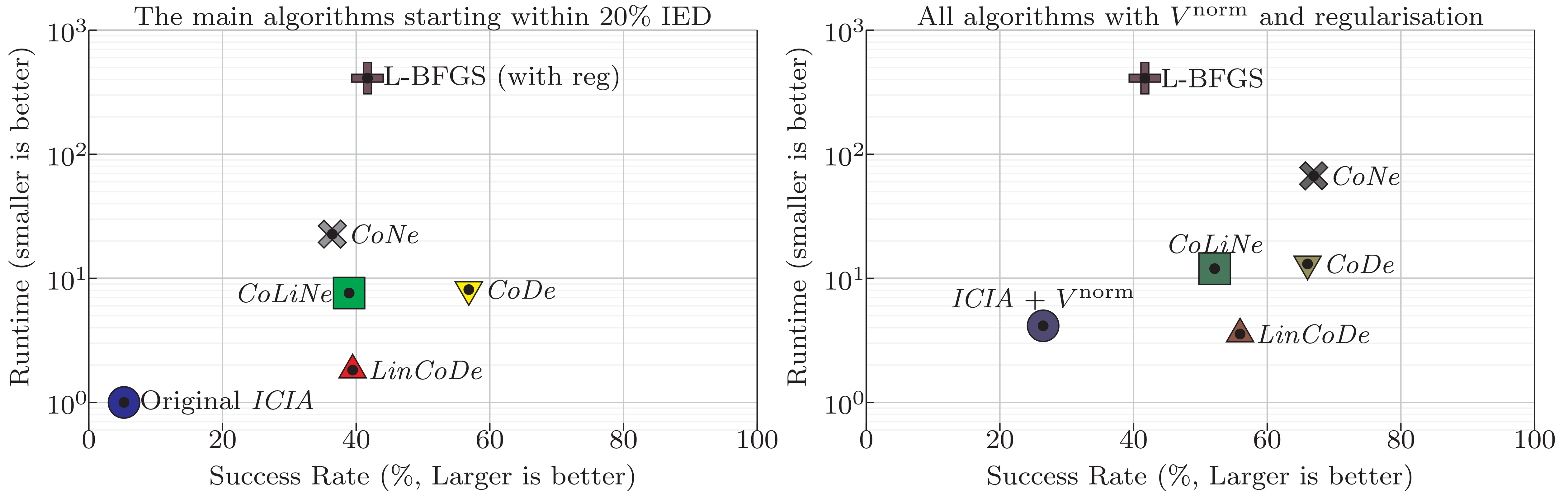
Tweets that were classified for both teams required tie-breaking.

This led to the following classification procedure:

1. Check for presence of tags and roster names.
2. If only relevent terms for 1 team, assign that team.
3. If both, choose team whose term appears first.

We also automatically classify tweets with more than 6 hashtags as irrelevant.

Our methods are at the performance/speed sweet point



Fitting a multiperson AAM. The best speed–performance tradeoffs come from the two new algorithms *CoDe* and *LinCoDe*. Note that *ICIA* is practically useless on this difficult multi-person dataset with a success rate near zero (left). It can be improved (right) by using the orthonormal incremental warp and regularisation. The *CoDe* algorithm with regularisation (right) is as accurate as the slow, approximation-free, compositional Gauss-Newton *CoNe* method but is seven times more efficient.

The experiments were performed with leave one identity out on a mixture of two databases (XM2VTS and IMM).

Tracking 5000 frames with a general model

ICIA with V^{Ortho}

Frame 10 Frame 50 Frame 450 Frame 2000 Frame 5000

ICIA with V^{Ortho} + Regularisation

Frame 10 Frame 50 Frame 450 Frame 2000 Frame 5000

Our algorithm makes fast and robust tracking possible. We compare face tracking under natural motion, using *ICIA*, *LinCoDe* and *CoDe*. The original *ICIA* fails immediately with this large model and new face data. Substituting the orthonormal incremental warp for the original *ICIA* warp, the algorithm still loses track very early, whereas *LinCoDe* and *CoDe* can track much further. Finally, adding regularisation to all algorithms, *ICIA* still loses track completely after approximately 500 frames and does not recover the local deformations accurately. In contrast *CoDe* now tracks the full 5000 frame sequence without reinitialization, and *LinCoDe* tracks for 2500 frames.

The same training dataset was used for both tracking experiments. The training data was aquired with different camera and light settings from different subjects.

References

[1] B. Amberg, A. Blake, T. Vetter On Compositional Image Alignment with an Application to Activce Appearance Models In *CVPR'09*, 2009.