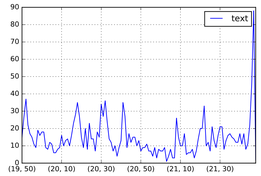


Tweets and NCAA March Madness

David Freed and Samuel Green

davidfreed@college.harvard.edu, samuelgreen@college.harvard.edu

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

Benchmarking and Old Model

...

The model was trained from 456 images from the IMM and XM2VTS datasets using 120 landmarks. Get the landmarks, model, and source code at:

`www.cs.unibas.ch/personen/amberg_brian/aam/`

References

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

Sentiment Analysis

As input to the model, we built a bag-of-words sentiment classifier using a labeled corpus of 4000 tweets. The classifer then uses a Naive-Bayes model to perform classfication after it has been trained using the labeled corpus. This training presents a problem to the analysis, since the classifier is trained on a set of tweets related to a different topic.

In order to use productively used the data to update the win probability model, we also had to associate the sentiment of the tweet with one or both teams.

Relevance Analysis

As input to the model, we built a bag-of-words sentiment classifier using a labeled corpus of 4000 tweets. The classifer then use a Naive Bayes model to perform classfication after training.

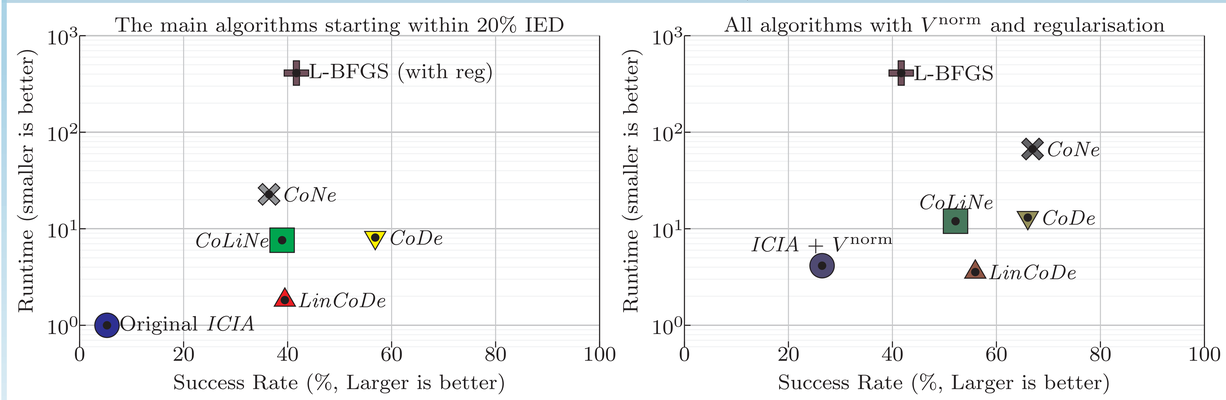
In order to use productively used the data to update the win probability model, we also had to associate the sentiment of the tweet with one or both teams.

Low Res Tracking

Tracking a low resolution video with large head motions succeeds with *CoDe*, where *ICIA* fails.

All methods used the orthonormal incremental warp, and relatively strong regularisation. *ICIA* starts to drift in the early frames, while *CoDe* tracks the full sequence. The approximate gradient method *LinCoDe* also suceeds, but looses track of the details for about 100 frames.

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

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.