UiO : Department of Informatics
University of Oslo

Using machine learning for cross-device tracking

MSc project

Elena Volkova

18.04.2017

A small disclaimer

This project turned out to have nothing to do with language technology. But due to the similarity in methodology, I think it might be interesting to see how all the tricks from a familiar toolbox are tried on a completely fresh problem.

What is cross-device matching?

- Increasing number of people use several devices;
- A lot of them use several devices for achieving the same goal;
- Cross-device matching is a way to connect several devices to the same user.

What types of cross-device matching are there?

Deterministic:

- Use a login (Google Analytics)

- Free
- Deterministic
- Not always enough

Probabilistic:

- Use machine learning

- Don’t need logins
- Invisible to the user
- Difficult

What for?

- Advertising: reducing media wastage;
- Personalization: better recommendations;
- Understanding the audience: better conversion;
- Theoretical questions: how anonymous we are on the internet.

Related research

- Not many scientific works: 7 articles from Drawbridge competition 2015.
  - Dataset: tables with cookies, mobile device identifiers and IP-addresses, and their properties;
  - Goal: for each mobile device in the test set produce a list of cookies connected to it;
  - Method: most participants used classification
    - Samples = device-cookie pairs
    - Classes = {match, non-match}
  - Evaluation: mean $F_{0.5}$-score for the positive class over all devices;
  - Number of participants: >300
- Many companies providing those services: Adbrain, Drawbridge, Tapad …

Challenges

- Technical:
  - Not always enough “labeled” data (i.e. that can be matched deterministically) for supervised ML;
  - A need for universal model: transferable to new websites;
  - Labeled samples are biased, labelling manually is impossible;
  - Scalability.
- Ethical:
  - Personalization on the whole might be harmful;
  - Probabilistic CDT is invisible to the consumer: no opt-out button
  - No regulations on who is responsible for personal data safety
  - Audio beacons

The project: goals

- Try to implement a cross-device tracking system on real-life traffic data;
- Focus on precision: can I make sense of some of this data and add a couple of certain connections to my user database?
- Try everything: supervised, semi-supervised, unsupervised ML;
- See how to adapt common traffic data to cross-device tracking;
- Look into differences and similarities between usage patterns on different websites;
- Any other insights into the problem that come up might also be useful.

The project: data

- Data was provided by Cxense
- Traffic logs from media websites. We took three sources with different geographical profiles:
  - Wall Street Journal
  - El Pais
  - Winnipeg Free Press

  Traffic logs are lists of “events” on the website in chronological order:
  - {"type":"repo","serverTime":1463360400065000,"ckp":"1378416945216339940619","site":"92223186138524869000…}

The project: supervised learning (1)

- Preprocessing: see flowchart
- Results of preprocessing: a set of labeled pairs for each source:
  - WSJ: ~ 700 000 samples, pos:neg = 0.18
  - WFP: ~ 45 000 samples, pos:neg = 0.44
  - ELP: ~ 4 000 samples, pos:neg = 4.2

  NB: don't forget that the dataset is biased.
- A "branch" of preprocessing: IP-addresses.
  - Public IP-s are filtered out (a rough heuristic) to downsize the number of possible pairs in the dataset
  - IP/device_id tables are kept for later (feature engineering)
- Environment: Apache Spark

The project: supervised learning (2)

- Feature engineering: a very important step
- Didn't find a way to apply representational learning ☹
- 25 features in total:
  - Individual features: sameLang, sameRegion, sameCity, sameDeviceType, + 1-hot-encoded device type for the two devices;
  - IP-based features: number of IP-s, common IP-s, similarity measures between sets of IP-s (Dice, Jaccard, Overlap), ratios of common IP-s to total. Some of these have 2 versions: including and excluding public IP-s.
  - IP- and event-based features: ratios of events from common IP-s to total, "micro" or "macro-averaged" for the two devices

NB: some features correlate

The project: supervised learning (3)

- Structurally: training a baseline, then trying everything to improve on it.
- Baseline: several classifiers, untuned, with default parameters, trained on 60% of the data, tested on 20% (development set)
- Environment: Python library scikit-learn.

Baseline

	Recall	Precision	F-score	Accuracy
Naive Bayes	0.83	0.47	0.60	0.79
Log Regression	0.65	0.78	0.71	0.90
SVM	0.62	0.81	0.70	0.90
Random Forest	0.77	0.78	0.78	0.91
Gradient Tree Boosting	0.80	0.76	0.78	0.91

WSJ

	Recall	Precision	F-score	Accuracy
Naive Bayes	0.84	0.51	0.64	0.71
Log Regression	0.61	0.62	0.61	0.77
SVM	0.49	0.69	0.57	0.78
Random Forest	0.60	0.67	0.63	0.79
Gradient Tree Boosting	0.58	0.67	0.65	0.80

WFP

	Recall	Precision	F-score	Accuracy
Naive Bayes	0.83	0.86	0.84	0.75
Log Regression	0.97	0.88	0.92	0.87
SVM	0.98	0.87	0.92	0.87
Random Forest	0.96	0.91	0.93	0.89
Gradient Tree Boosting	0.96	0.91	0.93	0.89

ELP

Possible ways to help

- Dealing with class imbalance: down- or up-sampling. Not a very radical imbalance, so downsampling makes recall a little better and precision – significantly worse, since the distribution in the test set is different.
- Scaling: standard scaling to unit variance and zero mean. Good for SVM, doesn't affect tree methods.
- Feature selection: by absolute threshold of usefulness or top K. We don't have too many features.
- Separate models for pairs of different types (like in Drawbridge competition): some types are better, others are worse.
- Calibration: helps with SVM and GTB (sometimes), hurts RF
- Tuning: always useful, done by default
- Moving the decision threshold: reaching a precision-recall trade-off that we like. alt's ok to make it extreme.
- Combining classifiers: voting, averaging, AND-ing. We use RF+SVM+GTB.

Evaluation on the test set

- Everything in the previous slide was optimized for development set. The final classifier is a voting ensemble of Random Forest, SVM and Gradient Tree Boosting, all of them with decision threshold 0.95. Now the proper results for the test set:

	Recall	Precision	F-score	Accuracy
WSJ	0.15	0.94	0.24	0.83
WFP	0.04	0.97	0.07	0.70
ELP	0.53	0.92	0.67	0.58

Transferring the model

- As was mentioned, not all websites have enough labeled data, so we need to make a model that works on unseen websites.
- The big disappointment: a model trained on one source does not work on another. The results are better than random guessing, but not by far. Precision for WFP drops back to 0.6.
- Possible reasons:
  - Different class distributions;
  - Different feature value distributions;
  - Specific features don’t work because of different geo-profile;
  - Model sort of overfits on one source and fails to capture more general patterns (provided they exist)

Possible ways to help

- **Different class distribution: undersampling. Helps (~10-20%) with ELP (which has very different class ratio), but not the others.**
- **Check for dataset/covariate shift: train a classifier to separate sources. F1 = 0.70 even without any geography-related features, which means the differences are deeper, in the IP-patterns.**
- **Check feature importance and feature value distributions: are the same features most important for models trained on different sources? Are their value distributions similar for negative and positive classes? (yes and yes).**
- **Overfitting: instead of a painstakingly tuned and optimized model try something simple. Surprisingly, a 10-iteration untuned (but calibrated) GTB gives much better results.**
- **Combining datasets: training on two/three sources together. Helps if tested on a source included in training. Sometimes helps if tested on a source not included in training. The most problematic source is still WFP. Possibly need more sources (although not the ideal solution).**

Overall, we couldn't reach a usable result when transferring the model.

The project: unsupervised learning (1)

- Would be great to avoid transferring issues if we could infer the connections from events on an individual website without using logins.
- This has not been tried in previous works.
- Two unsupervised strategies:
  - Cluster pairs of devices to match the split between positives and negatives – doesn't work;
  - Cluster the devices themselves

Clustering devices (1)

- Not really clustering, but rather searching for neigbors;
- Preprocessing similar to supervised learning, but without converting to pairs;
- Inspired by text processing:
  - Samples = devices
  - Features = IP-s, feature values = how many times the device has been seen on that IP
  - The result is a sparse matrix similar to distributional semantics
- A metric is introduced on devices in the IP-space: cosine similarity

Clustering devices (2)

- Two straight-forward ways to find neighbors:
  - Take all pairs with similarity > threshold (works better)
  - Take 1 nearest neighbor for every device (not all devices even have a match, which leads to many false positives)
- Results for threshold 0.95:
  - 0.85 precision for WSJ
  - 0.78 precision for WFP

Overall the results are not very high, but the approach looks viable: no transferance issues, + in cases when a match does exist, it is among 3NN in >90% cases.

The project: semi-supervised learning

- Most websites have at least some identifiable users. We could start with pairs of devices that are certain to come from the same person, and expand the set.
- Obtaining the seed set: either take a random set of positive instances or a set of “good”, prototypical matches.
  - How to find prototypical matches: we can do it with classification (the pairs that received a very high score), or possibly with hand-written rules
- Finding new pairs: either take 1NN for each seed or take all pairs that are closer than *threshold value* to some seed (better).
- Results for new pairs closer than 0.1 to any of the seeds:
  - 0.90 precision for WSJ
  - 0.82 precision for WFP
- Ideally we want to iterate this step several times: possibly better start with different seed set every time to avoid building up on mistakes. Not explored in this project.

Overall – also a viable strategy.

Conclusions

- So far only a supervised model trained and applied on the same source has reached results that can be used in practice: a voting ensemble of classifiers with a raised decision threshold;
- Supervised models do not transfer well from source to source due to slight differences in usage patterns and possibly also because some sources are less well fit for CDT;
- Unsupervised and semi-supervised learning might be the answer, and show promising results, but require further work

Further work

- Improve unsupervised and semi-supervised methods;
- Optimize all the decisions about the system parameters that were made by educated guessing;
- More features;
- More sources.

References (1)

- *Adbrain: Demystifying Cross-Device*. Whitepaper. Adbrain. URL: https://iabuk.net/resources/white-papers/adbrain-demystifying-cross-device
- R. Díaz-Morales. 'Cross-Device Tracking: Matching Devices and Cookies'. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015 IEEE International Conference on Data Mining Workshop (ICDMW)
- *ICDM 2015: Drawbridge Cross-Device Connections*. Kaggle. URL: https://www.kaggle.com/c/icdm-2015-drawbridge-cross-device-connections
- X. Cao, W. Huang and Y. Yu. 'Recovering Cross-Device Connections via Mining IP Footprints with Ensemble Learning'. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015 IEEE International Conference on Data Mining Workshop (ICDMW).

References (2)

- M. S. Kim et al. ‘Connecting Devices to Cookies via Filtering, Feature Engineering, and Boosting’. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015 IEEE International Conference on Data Mining Workshop (ICDMW)
- M. Landry, S. Rajkumar and R. Chong. ‘Multi-layer Classification: ICDM 2015 Drawbridge Cross-Device Connections Competition’. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015 IEEE International Conference on Data Mining Workshop (ICDMW)
- G. Kejela and C. Rong. ‘Cross-Device Consumer Identification’. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015 IEEE International Conference on Data Mining Workshop (ICDMW)

References (3)

- J. Walthers. ‘Learning to Rank for Cross-Device Identification’. In: *Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. ICDMW ’15. Washington, DC, USA: IEEE Computer Society, 2015
- E. Pariser. *Beware online "filter bubbles"*. URL: https://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles
- *FTC Issues Warning Letters to App Developers Using ‘Silverpush’ Code*. Press release. Federal Trade Commission. URL: https://www.ftc.gov/news-events/press-releases/2016/03/ftc-issues-warning-letters-appdevelopers-using-silverpush-code