Comparison of Feature Extraction Methods and Predictors for Income Inference

Martin Fixman*, Martin Minnoni†, Carlos Sarraute†
*Universidad de Buenos Aires, Argentina
†Grandata Labs, 550 15th Street, San Francisco, CA, USA
Email: martinfixman@gmail.com, {martin, charles}@grandata.com

***Abstract*—Patterns of mobile phone communications, coupled with the information of the social network graph and financial behavior, allow us to make inferences of users' socio-economic attributes such as their income level. We present here several methods to extract features from mobile phone usage (calls and messages), and compare different combinations of supervised machine learning techniques and sets of features used as input for the inference of users' income. Our experimental results show that the Bayesian method based on the communication graph outperforms standard machine learning algorithms using node-based features.**

I. Introduction

Mobile phone datasets present a rich view into social interactions and physical movements of large segments of the population. The voice calls and text messages exchanged between people, together with the locations of these communications, allow us to construct a rich social graph which provides insights on the users social fabric.

There is a strong homophily in the population's communications graph respect to economic variables such as the user's income [1], which results largely from social stratification between populations of different purchasing power [2] or purchasing patterns [3]. Additionally, and in part resulting from this stratification, there are different patterns of communication between users of distinct socioeconomic level [4].

Finding the best way to parse Call Detail Records (CDRs) to generate user features and construct their communications graph is still a subject of research. After describing our data sources, we present several methods of feature extraction from the raw CDR data, and describe the supervised machine learning algorithms used to predict the *socioeconomic level* of users, given a ground truth for a relatively small subset of the population. In particular we tested the Bayesian approach for income inference presented in [1]. Finally, we present our experimental results, comparing the performance obtained according to the feature set and the algorithms used.

II. Data Sources

The data used in this study contains a set P of *Call Detail Records* (CDRs), composed of voice calls, and another set S containing text messages, from a telecommunication company (*telco*) in a Latin American country for a period of 3 consecutive months. Using this data we create the social graph $G = \langle V, E \rangle$ where each $v \in V$ is a user of the telco, and E contains calls between those users. Each element $e \in E$ contains information about the *Origin* and *Destination* users, in addition to the amount of *Calls*, the total call *Time*, and the amount of *SMS* exchanged.

Additionally, we have access to information about a set B of bank accounts, for which we calculate the monthly income for each user p_s. In this paper we separate the users into two groups of equal size: *Low Income* and *High Income*.

B contains information about the users' telephone number, which is anonymized in the same way as the telco data. Therefore, we can match the data in these two datasets in order to construct the *Ground Truth* $T \subseteq V$, where each element of T contains its income category, along with the *Inner Graph* $G' = \langle V', E' \rangle$ where E' contains edges where at least one endpoint is in T, and V' is the set of endpoints of all elements in E'.

III. Accumulated Graph Features

This section presents several ways of transforming data from the graph $G = \langle V, E \rangle$ into individual features for each user $v \in V$. The aggregations are classified into levels named according to the transformation done to G, and they are merged with levels containing less information. The total amount of columns in each featureset is presented in Table II.

A. User Data — Level Ring_1

The first accumulated features consist of accumulating the three quantifiable features, *Calls*, *Time* and *SMS*, for every node, separated on whether those features are incoming or outgoing.

B. Higher Order User Data — Level $\mathrm{Ring}_{n>1}$

The *ego network* of the node v is defined as the graph consisting of v and its neighbors. A simple way to get more features about that node is to accumulate the calls and SMS information about the edges which are **not** part of the ego network, but contain one endpoint on the border of the ego network.

Similarly, we define the *user data of order* n, for any natural number n, as the accumulation of calls and SMS information for the nodes which are part of the *ego network of order* n, which is the set of nodes which are at distance at most n of v, and are not part of the *ego network of order* $n-1$, which we denote as Ring_n. The level Ring_1 contains the information of the regular *user data*, while the user data from the *ego network of order* n is assigned to Ring_n for $n > 1$.

Inner Graph

Model	Level	Accuracy	Precision	Recall	AUC	F_1-score	F_4-score	Fit Time	Predict Time
Random Selection		0.499	0.499	0.500	0.499	0.500	0.500	—	0.005 s
Majority Voting		0.681	0.640	0.826	0.681	0.721	0.712	—	0.059 s
Bayesian Algorithm		0.693	0.665	0.792	**0.746**	**0.723**	**0.783**	—	33.155 s
LR	Ring_1	0.536	0.531	0.625	0.536	0.574	0.619	0.145 s	0.002 s
	Ring_2	0.535	0.525	0.730	0.535	0.611	0.714	0.141 s	0.011 s
	Ring_3	0.568	0.578	0.525	0.569	0.550	0.528	0.119 s	0.003 s
	Cat_1	0.686	0.655	0.785	0.686	0.714	0.776	0.167 s	0.005 s
	Cat_2	0.693	0.665	0.780	0.693	0.718	0.772	1.588 s	0.011 s
	Cat_3	0.693	0.670	0.764	0.692	0.714	0.758	0.956 s	0.009 s
RF	Ring_1	0.548	0.548	0.550	0.548	0.549	0.550	5.986 s	0.588 s
	Ring_2	0.582	0.583	0.577	0.582	0.580	0.577	56.548 s	0.483 s
	Ring_3	0.576	0.577	0.580	0.576	0.579	0.580	50.197 s	0.253 s
	Cat_1	0.671	0.665	0.690	0.671	0.677	0.688	6.346 s	0.539 s
	Cat_2	**0.714**	0.713	0.716	0.714	0.714	0.716	96.005 s	0.460 s
	Cat_3	0.709	0.710	0.711	0.709	0.711	0.711	81.528 s	0.242 s

TABLE I: Resulting metrics of different methods used in Section V tested on the *Inner Graph*, which contains only nodes which have at least one neighbour with socioeconomic information. **LR** corresponds to *Logistic Regression* models, and **RF** to *Random Forest* models with the level described in Section III. Bolded items represent the highest value for the metrics Accuracy, AUC, F_1-score and F_4-score.

Level	Features
Ring_1	8
Ring_2	16
Ring_3	24
Cat_1	24
Cat_2	48
Cat_3	72

TABLE II: Amount of total features per level.

C. Categorical User Data — Level Cat_n

Another approach to building features is to do an approach similar to the *user data* presented in Section III-A, but further discriminating each feature which corresponds to each node $v \in V$ and each edge $e \in E$: when $t \in T$ is the other endpoint of the link e, we discriminate whether t corresponds to a user with high or low income. The resulting new features are of the form represented by the set in Equation (1). This way we create the datasets $\mathrm{Cat}_1 \dots \mathrm{Cat}_3$ by using the growing ego networks $\mathrm{Ring}_1 \dots \mathrm{Ring}_3$.

$$\left\{ \begin{matrix} in \\ out \end{matrix} \right\} \times \left\{ \begin{matrix} calls \\ time \\ sms \\ contacts \end{matrix} \right\} \times \left\{ \begin{matrix} low \\ high \end{matrix} \right\} \tag{1}$$

To prevent overfitting, the set T is partitioned into two disjoint sets, G and H, where G contains roughly 75% of the nodes in T is used to calculate the features, while H contains the other 25% and is used to train the models.

IV. Inference Methodology

The *Inner Graph* is defined so that a node $h \in H$ is part of it if and only if there is an edge $\langle h, x \rangle \in E$ or $\langle x, h \rangle \in E$ such that $x \in H$. This later definition becomes important when doing inferences on features using the *Categorical User Data* dataset.

The inferences were performed using *Logistic Regression* and *Random Forest* classifiers, both of which are solid classifiers commonly used for cases like this [6], and since they tend to have different variance in their results [8], noise from different sources should affect differently each predictor.

The features used were the ones presented in Section III, where each level is merged with all the previous levels with the data on G. Table II shows the amount of features in each level after merging the data. The feature sets $\mathrm{Ring}_1 \dots \mathrm{Ring}_3$ and $\mathrm{Cat}_1 \dots \mathrm{Cat}_3$ are strictly increasing.

The classifiers are trained using those features and the labels in H doing a *Grid Search* on different hyperparameters of the predictors with *5-fold cross-validation* to prevent overfitting. In addition to the *accuracy*, we computed several metrics for the comparison. In our real-life use cases, we are more interested in having high *recall* than high *precision* (that is, finding more high income users than being accurate), therefore we also measured the *F_4-score* of each prediction.

In addition, these methods based on features aggregated by node are compared against three other methods based solely on the communication graph structure:

- **Random Selection** which chooses a random category.
- **Majority Voting** which chooses the most populated category in a user's ego network (or randomly in case of a tie).
- **Bayesian Method** which uses the method presented in [1] to infer the category of each user, taking into account the uncertainty on the available information.

V. Results and Conclusion

The predictors were run in a computer with a single core of 2.00 GHz Intel Xeon CPU using `sklearn 0.18` under `Python 2.7`, and 64 GB of RAM (enough to avoid caching calculations). The results of the inference can be found in

Full Graph

Model	Level	Accuracy	Precision	Recall	AUC	F_1-score	F_4-score	Fit Time	Predict Time
Random	**Selection**	0.499	0.499	0.500	0.499	0.500	0.500	—	0.005 s
Majority	**Voting**	0.565	0.747	0.197	0.565	0.312	0.206	—	0.204 s
LR	Ring_1	0.534	0.586	0.234	0.534	0.335	0.243	0.937 s	0.016 s
	Ring_2	0.547	0.617	0.250	0.547	0.356	0.260	1.347 s	0.035 s
	Ring_3	0.563	0.586	0.430	0.563	0.496	0.437	1.055 s	0.023 s
	Cat_1	0.565	0.746	0.198	0.565	0.313	0.207	1.871 s	0.041 s
	Cat_2	0.577	0.727	0.247	0.577	0.368	0.257	9.816 s	0.077 s
	Cat_3	0.589	0.636	0.415	0.589	0.503	0.424	9.456 s	0.065 s
RF	Ring_1	0.543	0.544	0.529	0.543	0.536	0.530	25.789 s	4.878 s
	Ring_2	0.578	0.585	0.537	0.578	0.560	0.540	102.961 s	5.608 s
	Ring_3	0.583	0.590	0.541	0.583	0.564	0.543	70.447 s	3.148 s
	Cat_1	0.568	0.573	0.536	0.568	0.554	0.538	32.981 s	5.371 s
	Cat_2	0.613	0.634	0.533	0.613	0.579	0.538	44.911 s	6.002 s
	Cat_3	**0.614**	0.635	0.534	**0.614**	**0.580**	**0.539**	50.589 s	3.484 s

TABLE III: Resulting metrics of different methods used in Section V tested on both the *Full Graph*, which includes all the nodes of the graph. **LR** corresponds to *Logistic Regression* models, and **RF** to *Random Forest* ones with the level described in Section III. Bolded items represent the highest value for the metrics Accuracy, AUC, F_1-score and F_4-score.

Table I for the *Inner Graph* and in Table III for the *Full Graph*.

Both tables show various metrics which result from applying the methods described in Section IV with the hyperparameters that result in the highest *accuracy* according to the *Grid Search*. We use the *AUC* (Area under the ROC curve) to compare the different approaches.

We observe that methods based on *Random Forests* tend to perform better in this real-world scenario than the ones based on *Logistic Regression*. This may be due to the fact that *Random Forests* are more versatile with non-linear data [7], and is consistent with similar findings in [5].

Interestingly, increasing the breadth of the *Ego Network* by one level, from Ring_1 to Ring_2 improves the performance when using *Random Forest* learning, however it does not improve by going one level further to Ring_3 in the case of the *Inner Graph*, despite the fact that this dataset is a strict superset of the previous Ring_2. This is due to the fallibility of common *bagging methods* like *Random Forest*, where having some noisy or non-informative data to choose from makes it less probable that informative features will be chosen.

Adding categorical information greatly improves the prediction when using either method, particularly on *Random Forest*, and, like it was noted before, adding neighbouring data of the *ego network of distance 2* (Cat_2) also results in a better predictor. However, raising further the maximum distance to Cat_3 within the ego network in the case of the *Inner Graph* does not result in further improvements.

We can conclude that, within the machine learning methods presented, the best in terms of *AUC* is predicting the category using a *Random Forest* with the *ego network of distance 2* data (Cat_2) in the case of the *Inner Graph*. In the *Full Graph*, using Cat_3 data results in slightly better results, however the difference with Cat_2 is very small.

Finally, in the *Inner Graph* the best method is the *Bayesian Method* presented in [1], which only uses the number of *High Income* and *Low Income* users in the egonetwork, but makes a "smarter" prediction than the machine learning methods LR and RF using the models Cat_1, Cat_2, and Cat_3, which also contain this data.

Additionally, while its AUC is not higher than the best machine learning methods, the F_1-score of the *Majority Voting* predictor is higher than all the machine learning methods.

We can reach the conclusion that, in this particular case, *smaller is better*. The machine learning methods which use many features (despite these features being informative) are not better at predicting the socioeconomic level of a user than the simple *Majority Voting* or the more complex *Bayesian Method* which use only 2 simple features of the communication graph.

REFERENCES

[1] Martin Fixman, Ariel Berenstein, Jorge Brea, Martin Minnoni, Matias Travizano, and Carlos Sarraute. A bayesian approach to income inference in a communication network. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 579–582. IEEE, Aug 2016.

[2] Yannick Leo, Eric Fleury, J. Ignacio Alvarez-Hamelin, Carlos Sarraute, and Márton Karsai. Socioeconomic correlations and stratification in social-communication networks. *Journal of The Royal Society Interface*, 13(125), 2016.

[3] Yannick Leo, Marton Karsai, Carlos Sarraute, and Eric Fleury. Correlations of consumption patterns in social-economic networks. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 493–500. IEEE, Aug 2016.

[4] Shaojun Luo, Flaviano Morone, Carlos Sarraute, Matias Travizano, and Hernan A. Makse. Inferring personal economic status from social network location. *Nature Communications*, 8(15227), May 2017.

[5] David Muchlinski, David Siroky, Jingrui He, and Matthew Kocher. Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data. *Political Analysis*, 24(1):87–103, 2016.

[6] David M W Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. Technical report, School of Informatics and Engineering, Flinders University of South Australia, 2007. http://csem.flinders.edu.au/research/techreps/SIE07001.pdf.

[7] Lalit Sachan. Logistic regression vs decision trees vs svm. http://www.edvancer.in/logistic-regression-vs-decision-trees-vs-svm-part2/.

[8] Jo-Anne Ting, Franziska Meier, Sethu Vijayakumar, and Stefan Schaal. *Locally Weighted Regression for Control*, pages 1–14. Springer US, Boston, MA, 2016.