Ciencia de Redes (Humanas y Sociales)
Clase #5

Carlos Sarraute[1]

[1] Grandata Labs, Buenos Aires and San Francisco
charles@grandata.com

Abril - Junio 2019

Agenda

Presentación

Basada en el paper

June 11-15, 2018. International School and Conference on Network Science (NetSci)
"Featurization Methods and Predictors for Income Inference Based on Communication Patterns"
Carlos Sarraute, Martin Fixman, Martin Minnoni, Matias Travizano

Human Behavior

There's definitely, definitely, definitely no logic
to human behavior

Björk — Debut

Restate my assumptions:
1. Mathematics is the language of nature.
2. Everything around us can be represented and understood through numbers.
3. If you graph the numbers of any system, patterns emerge.

Darren Aronofsky — Pi

The Scientific Connection

Scientific Collaborations

- Hernan Makse (CCNY)
- Aline Viana (Inria, Paris)
- Eric Fleury, Marton Karsai (ENS, Lyon)
- Sandy Pentland and the Human Dynamics team (MIT)
- Marta Gonzalez and the Human Mobility team (MIT)
- Alejo Salles and Pablo Groisman (UBA)
- Fundación Mundo Sano

The Scientific Connection

Scientific Collaborations

- Hernan Makse (CCNY)
- Aline Viana (Inria, Paris)
- Eric Fleury, Marton Karsai (ENS, Lyon)
- Sandy Pentland and the Human Dynamics team (MIT)
- Marta Gonzalez and the Human Mobility team (MIT)
- Alejo Salles and Pablo Groisman (UBA)
- Fundación Mundo Sano

Publications

- 56 papers published!
- Conferences: NetMob, ASONAM, KDD, AGRANDA, ...
- Journals: Nature Communications, AI Communications, ...

Summary

Objective

Compare methods for the inference of socioeconomic status in the communication graph.

Summary

Objective

Compare methods for the inference of socioeconomic status in the communication graph.

- Use 2 data sources:
  - Call Detail Records (CDRs) from the operator allow us to construct a social graph.
  - Banking reported income for a subset of clients obtained from a large bank.
- We construct an inference algorithm that allows us to predict the socioeconomic status of users.
- We compare it with standard machine learning techniques using growing set of features from nodes and their network.

Datasets

Mobile Phone Data Source

Each CDR $\boldsymbol{p} \in \mathcal{P}$ contains:

- phone numbers of origin and destination $\langle \boldsymbol{p_o}, \boldsymbol{p_d} \rangle$ anonymized using a cryptographic hash function
- starting time $\boldsymbol{p_t}$, call duration $\boldsymbol{p_s}$
- latitude and longitude of antenna used $\langle \boldsymbol{p_y}, \boldsymbol{p_x} \rangle$ for subset of data.

Datasets

Mobile Phone Data Source

Each CDR $p \in \mathcal{P}$ contains:

- phone numbers of origin and destination $\langle p_o, p_d \rangle$ anonymized using a cryptographic hash function
- starting time p_t, call duration p_s
- latitude and longitude of antenna used $\langle p_y, p_x \rangle$ for subset of data.

Banking Information

- Account balances for over 10 million clients of a bank for a period of 6 months, denoted $\mathcal{B}$.
- For each client $b \in \mathcal{B}$ we have his phone number b_p, anonymized with the same hash function used in $\mathcal{P}$.
- The average income of 6 months b_s.

Bank and Telco Matching

- Phone numbers in each call p_o and p_d are anonymized with the same hash function as the phone number in the bank data, b_p.
- We can match users to their unique phone to create the social graph:

$$G = \mathcal{P} \bowtie_{p_o = b_p} \mathcal{B} \bowtie_{p_d = b_p} \mathcal{B}$$

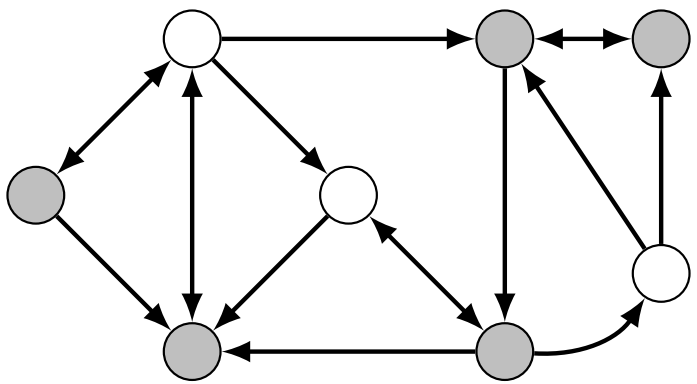
- $\forall g \in G$ we have its phone number g_p, its average income over 6 months g_s, and its age g_a.
- This graph has a total of 2,027,554 nodes with 5,044,976 edges, which represent 29,599,762 calls and 5,476,783 text messages.

Fuente de Datos

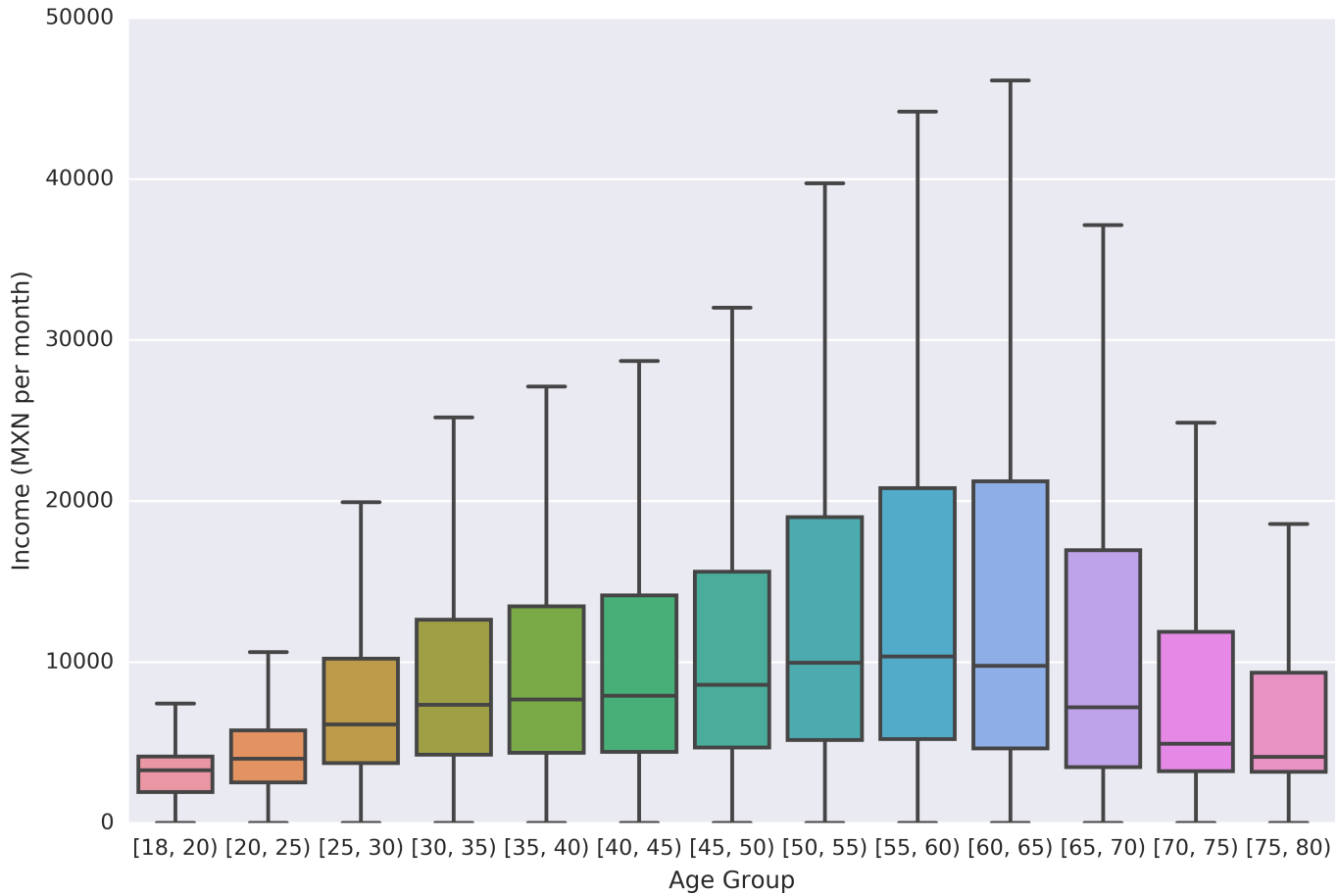
Con estos datos se calcula el *Grafo Social*.

$$\boldsymbol{G} = \langle \mathrm{V}, \mathrm{E} \rangle$$

Donde $\boldsymbol{V}$ contiene datos de usuarios y su nivel de ingreso (si se conoce), y $\boldsymbol{E}$ contiene sus conexiones con otros usuarios. Se puede usar el grafo social para entender el comportamiento de los usuarios (Gonzalez et al. (2008), Ponieman et al. (2013), Sarraute et al. (2015)).

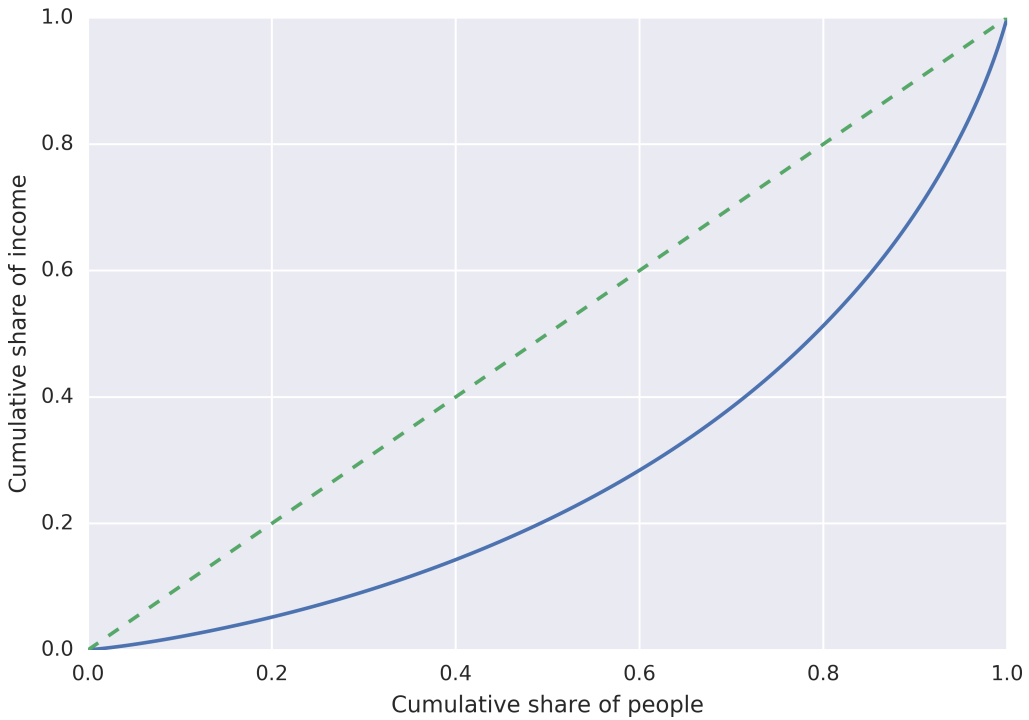


Distribución de Ingresos por Edad



Distribución de ingresos por grupo de edad.

Distribución de Ingresos Totales



Proporción acumulada de ingresos por proporción acumulada de la población.

Dentro de los usuarios de este banco:

- 20% de la población tiene 50% de los activos.
- Gini = 45%.

Agenda

¿Que usamos?

Features individuales ?

o

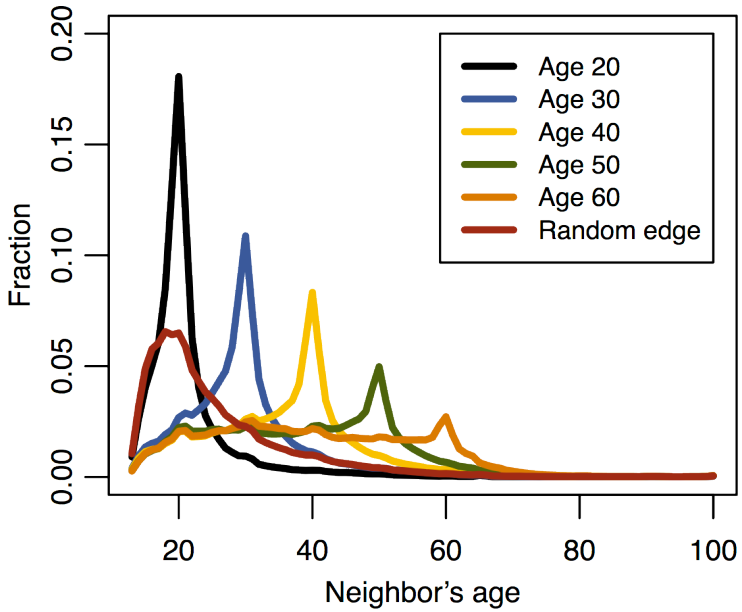
Topología de la red ??

Homofilia Social

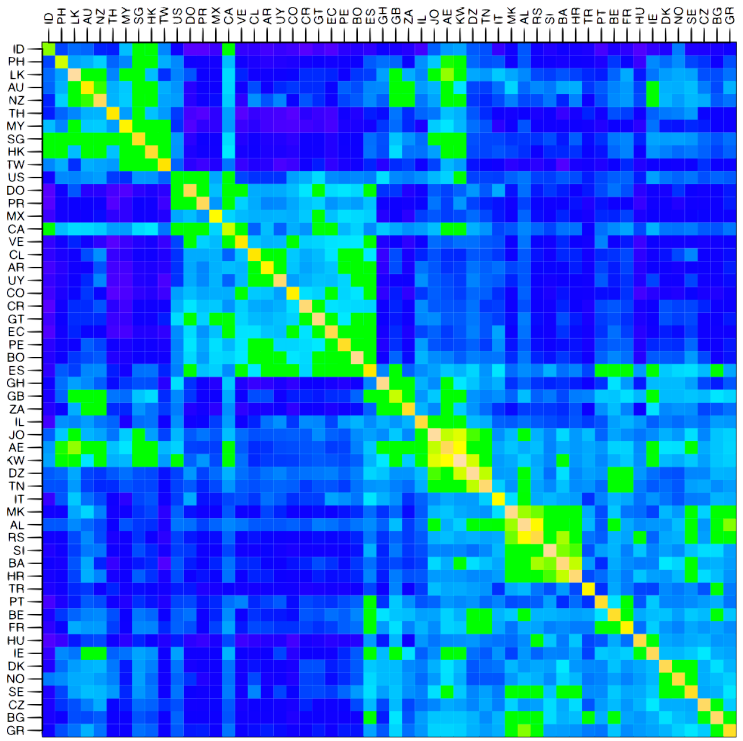
"La gente ama a los que son como sí mismos."

Aristoteles
Retórica

Homofilia Social



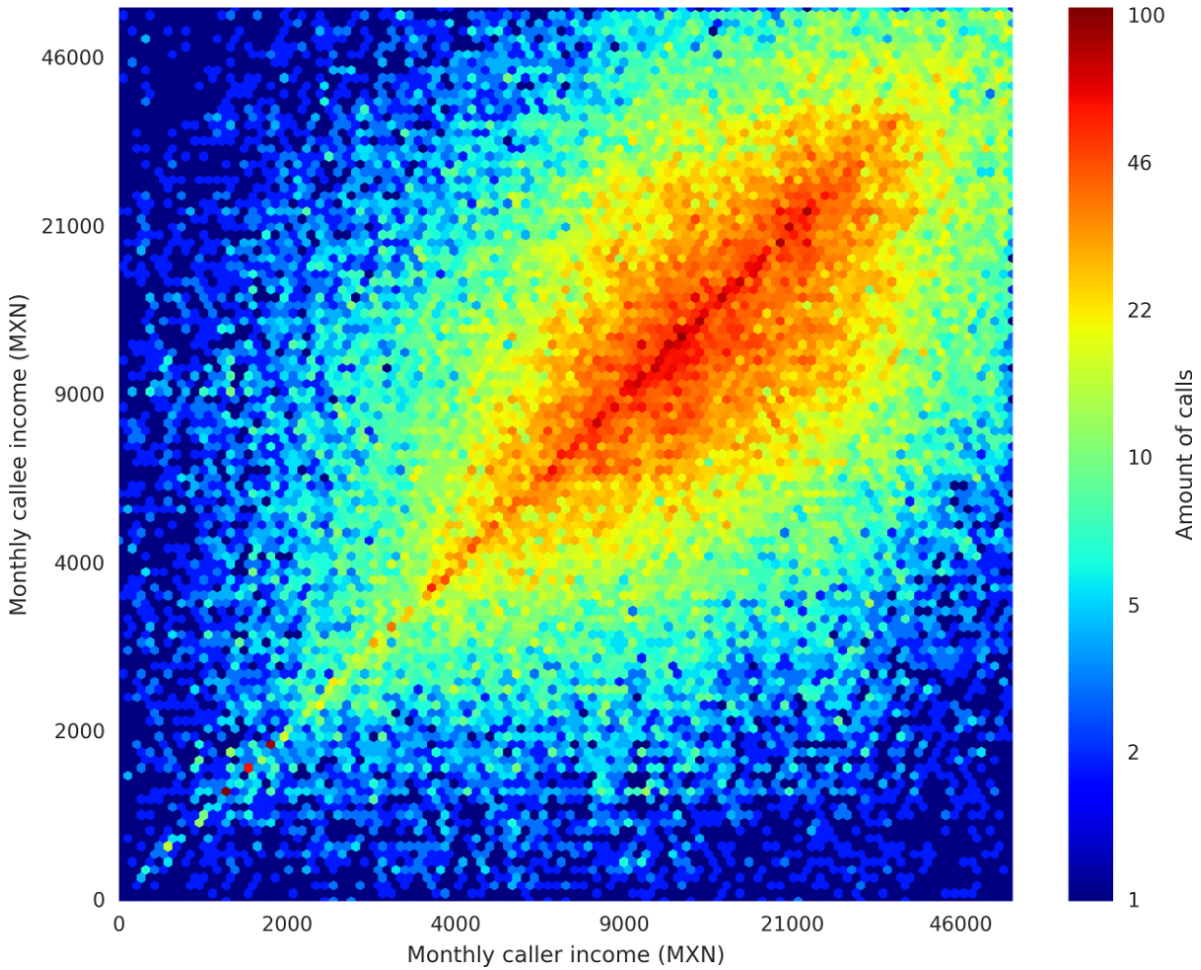
a



b

Ejemplos varios de homofilia en un cierto grafo social Ugander et al. (2011). a: Distribución de edades para contactos de usuarios de cada edad. b: Mapa de calor marcando la cantidad normalizada de contactos entre cada par de países.

Income Homophily



Number of calls between users, according to their monthly income

Similar to homophily with respect to age in Brea et al. (2014).

What do we predict?

Instead of predicting the exact value of a user's income, our strategy is to distinguish between 2 categories:

- $\boldsymbol{R_1 = [1000, 6300)}$ i.e. low income
- $\boldsymbol{R_2 = [6300, \infty)}$ i.e. high income

We place users into two distinct groups $\boldsymbol{H_1, H_2 \subseteq G}$:

$$\boldsymbol{g \in H_i \iff g_s \in R_i}$$

Features, features, features

$$\mathbf{calls}_v^{\mathbf{low}} = \sum_{\substack{e \in E \\ e_d = v \\ e_o \in H_1}} e_c + \sum_{\substack{e \in E \\ e_o = v \\ e_d \in H_1}} e_c \qquad \mathbf{calls}_v^{\mathbf{high}} = \sum_{\substack{e \in E \\ e_d = v \\ e_o \in H_2}} e_c + \sum_{\substack{e \in E \\ e_o = v \\ e_d \in H_2}} e_c$$

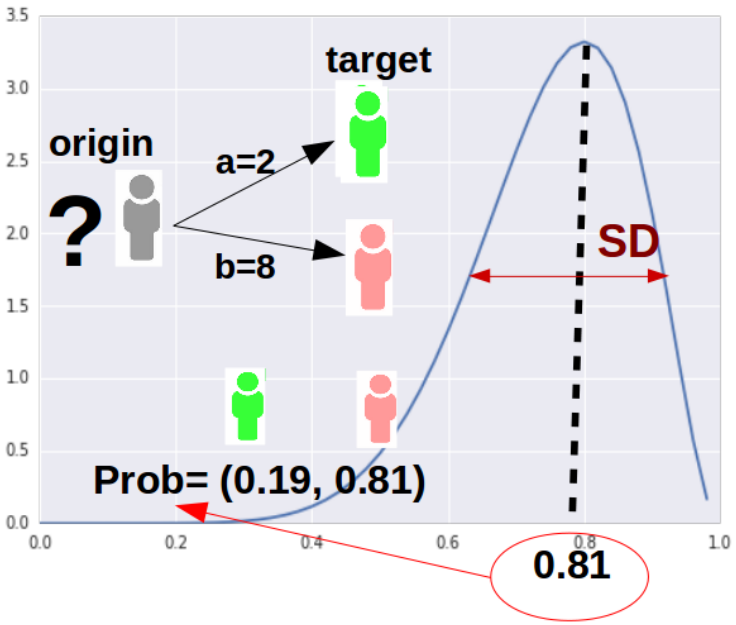
$$\mathbf{time}_v^{\mathbf{low}} = \sum_{\substack{e \in E \\ e_d = v \\ e_o \in H_1}} e_t + \sum_{\substack{e \in E \\ e_o = v \\ e_d \in H_1}} e_t \qquad \mathbf{time}_v^{\mathbf{high}} = \sum_{\substack{e \in E \\ e_d = v \\ e_o \in H_2}} e_t + \sum_{\substack{e \in E \\ e_o = v \\ e_d \in H_2}} e_t$$

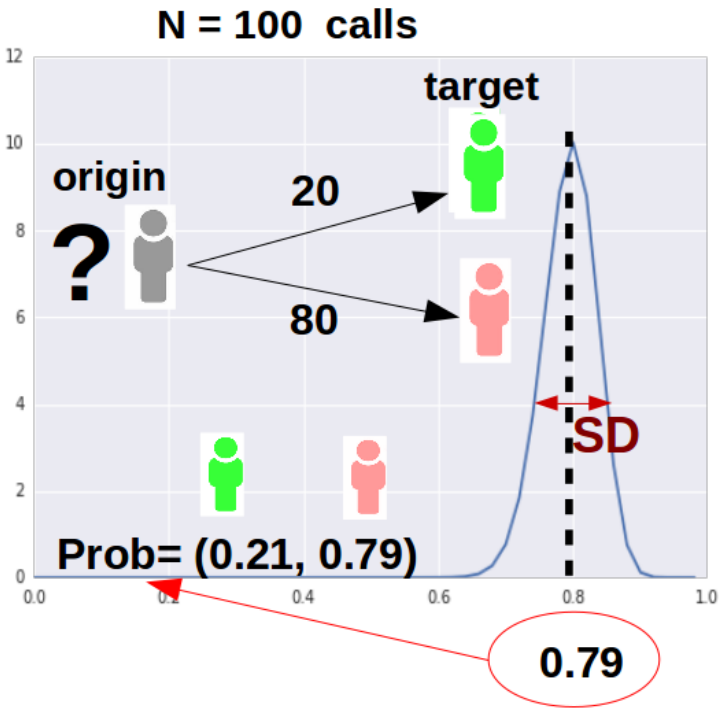
$$\mathbf{sms}_v^{\mathbf{low}} = \sum_{\substack{e \in E \\ e_d = v \\ e_o \in H_1}} e_s + \sum_{\substack{e \in E \\ e_o = v \\ e_d \in H_1}} e_s \qquad \mathbf{sms}_v^{\mathbf{high}} = \sum_{\substack{e \in E \\ e_d = v \\ e_o \in H_2}} e_s + \sum_{\substack{e \in E \\ e_o = v \\ e_d \in H_2}} e_s$$

$$\mathbf{contacts}_v^{\mathbf{low}} = |\{e \in E \mid e_o = v \wedge e_d \in H_1\} \cup \{e \in E \mid e_d = v \wedge e_o \in H_1\}|$$

$$\mathbf{contacts}_v^{\mathbf{high}} = |\{e \in E \mid e_o = v \wedge e_d \in H_2\} \cup \{e \in E \mid e_d = v \wedge e_o \in H_2\}|$$

Motivation





The frequency of calls (to category 1 and 2) loses information.
We want to compare distributions.

Beta Distribution

We define B^j as the Beta probability distribution function for each user:

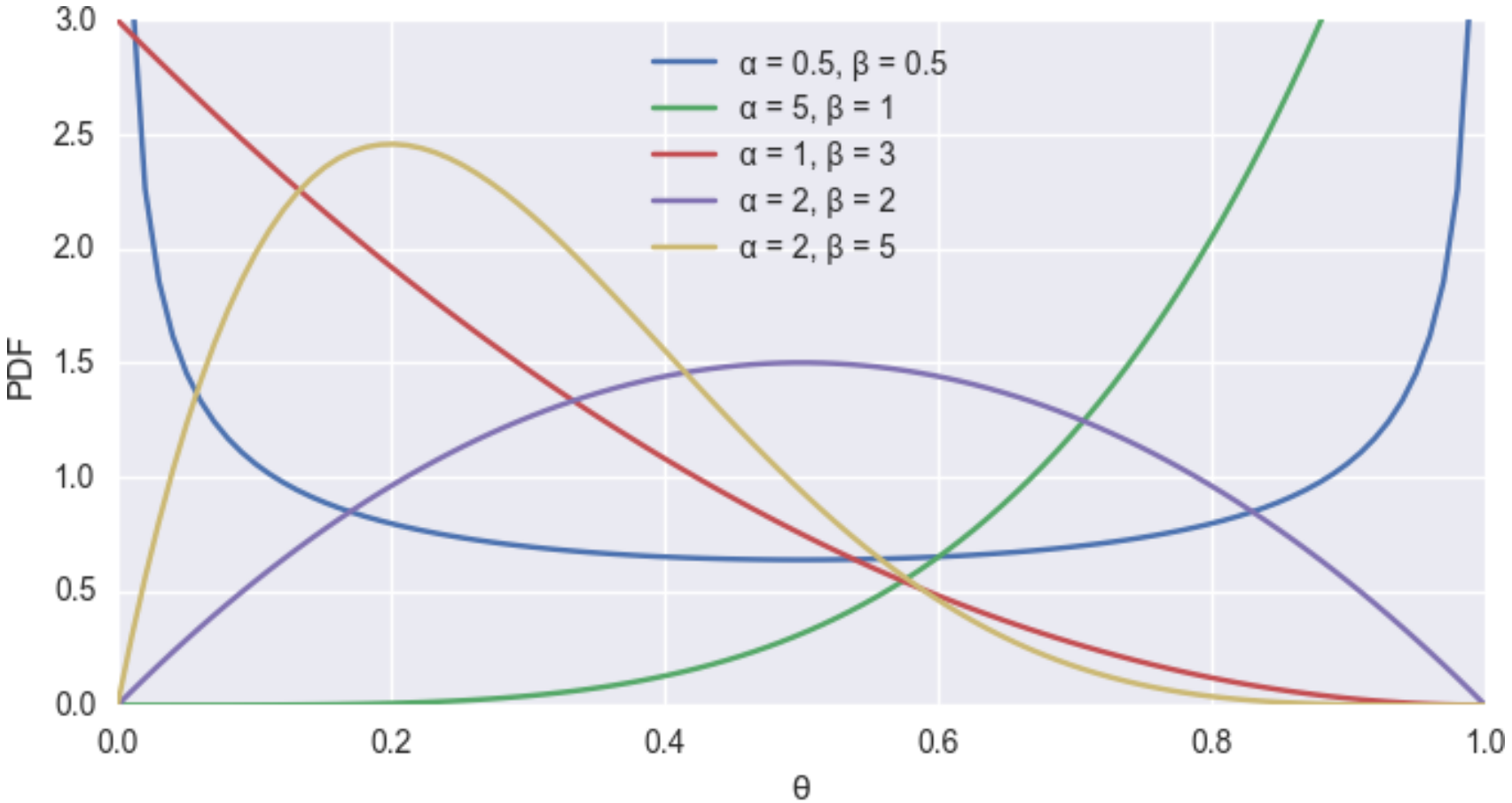
$$B^j\left(x;\alpha^j,\beta^j\right)=\frac{1}{B\left(\alpha^j,\beta^j\right)}x^{\alpha^j-1}\cdot(1-x)^{\beta^j-1} \quad (1)$$

where $\alpha^j=a_1^j+1$ and $\beta^j=a_2^j+1$ are the parameters of the Beta distribution, and B is the beta function, defined as:

$$B\left(\alpha,\beta\right)=\frac{\Gamma(\alpha)\cdot\Gamma(\beta)}{\Gamma(\alpha+\beta)} \quad (2)$$

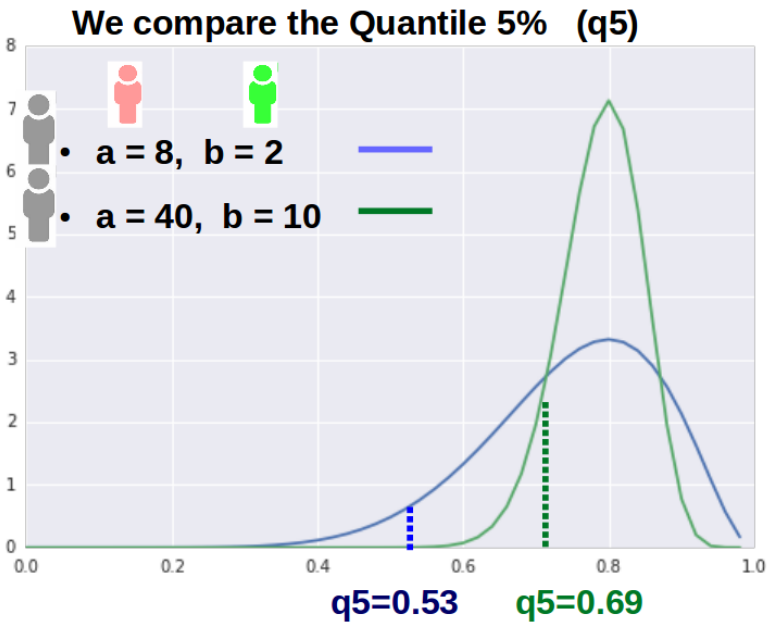
We obtain a Beta distribution for the probability of belonging to high income category (for each user).

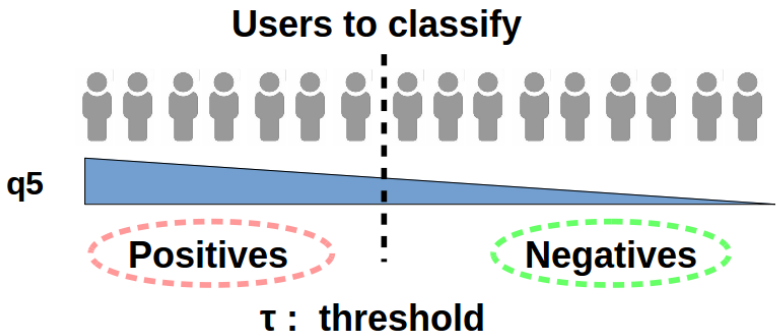
Distribución Beta



Distribución Beta para diferentes valores de α y β.

Determining the category





- Find the lowest 5 percentile $q5$ for this probability.
- If $q5$ is above threshold τ, we assign user to H_2.
- Take into account both the mean and the broadness (uncertainty) of the distribution.
- Category assigned to a user depends on its Beta distribution and on our choice of τ.

Agenda

Confusion matrix

		True condition		
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$

Evaluation of Performance

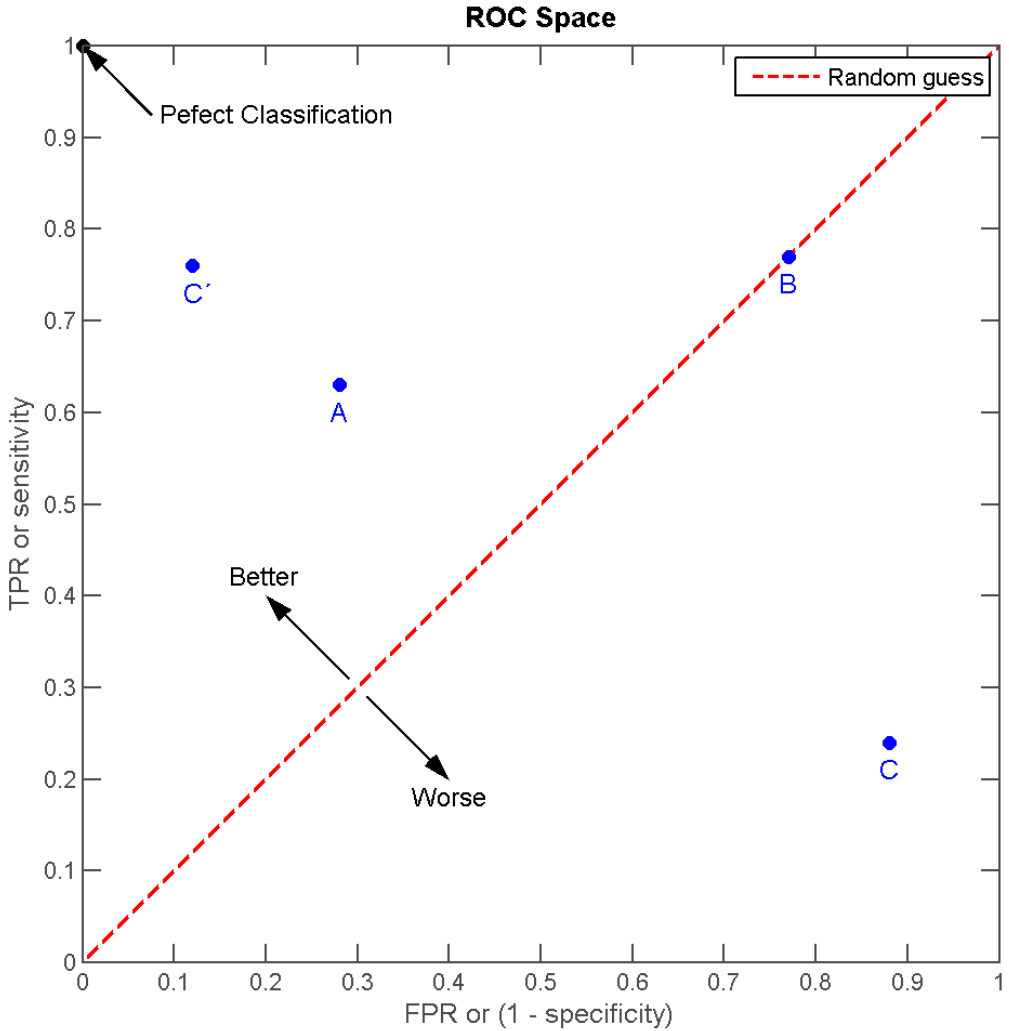
We have:

- $\mathbf{TP}$ is the number of correctly predicted users with high income,
- $\mathbf{P}$ is the total number of users with high income,
- $\mathbf{FP}$ is the number of users incorrectly classified as having high income,
- $\mathbf{N}$ is the total number of users with low income.

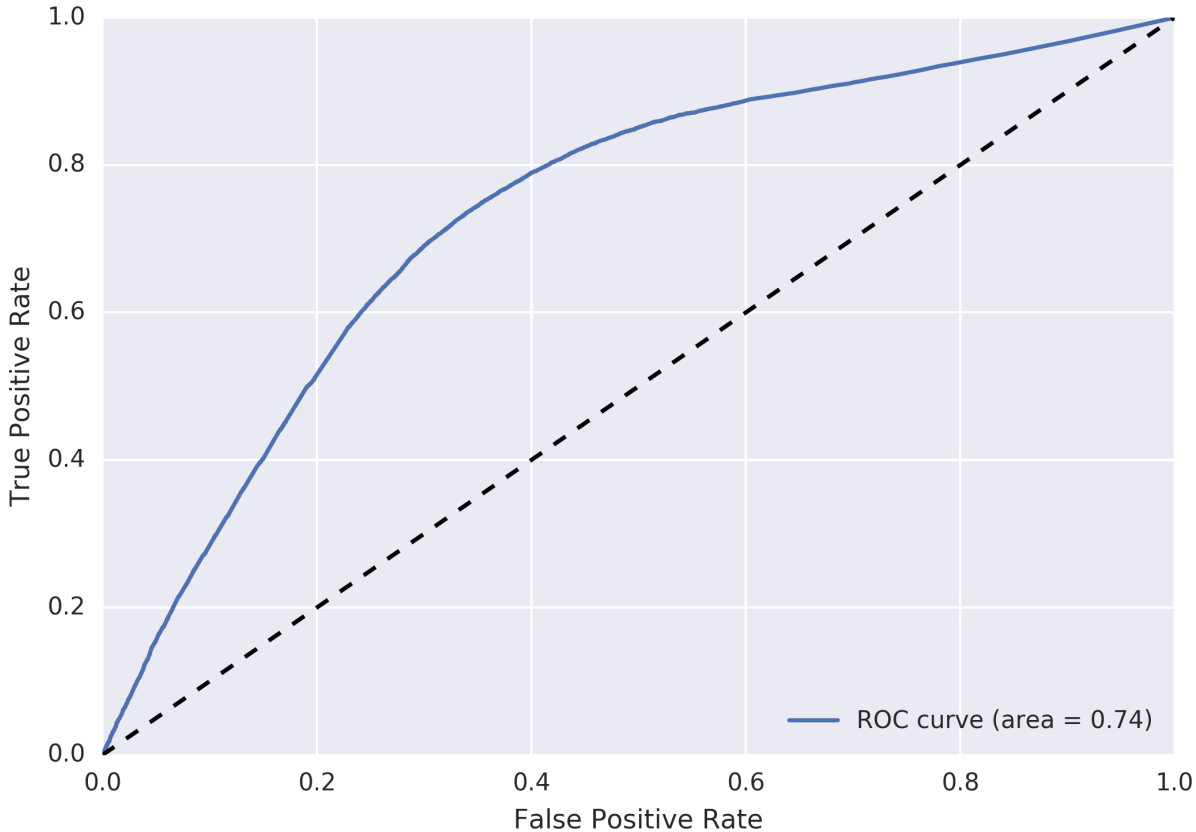
We examine:

- accuracy $= (\mathbf{TP} + \mathbf{TN})/(\mathbf{P} + \mathbf{N})$
- precision $= \mathbf{TP}\,/(\mathbf{TP} + \mathbf{FP})$
- recall $=$ true positive rate $\mathbf{TPR} = \mathbf{TP}\,/\,\mathbf{P}$
- false positive rate $\mathbf{FPR} = \mathbf{FP}\,/\,\mathbf{N}$
- $\boldsymbol{F_1}$ score $=$ harmonic mean between precision and recall

ROC Curve


ROC Space
Random guess
Pefect Classification
C´
B
A
Better
Worse
C
TPR or sensitivity
FPR or (1 - specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

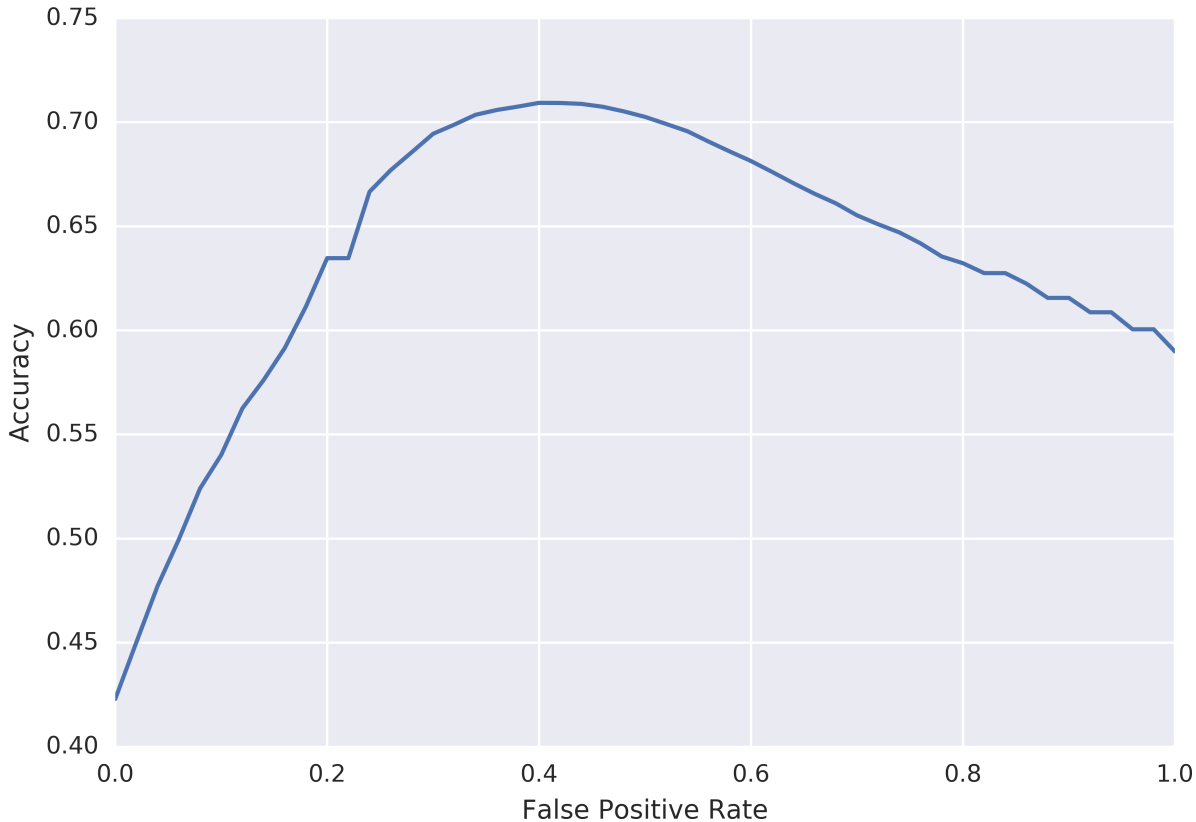
ROC Curve



ROC curve for prediction procedure

We observed an $\mathbf{AUC = 0.74}$ indicating that our predictor is better than a random predictor ($\mathbf{AUC \simeq 0.50}$).

Accuracy



Accuracy as a function of FPR

The best accuracy obtained is 0.71 for $\boldsymbol{\tau = 0.51}$.

Agenda

Otros Modelos Basados en Machine Learning

Presentamos otros métodos basados en prácticas más comunes del aprendizaje automático. El problema a resolver sigue siendo el mismo.

Dado un *grafo social* $\boldsymbol{G} = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$, buscar cuáles usuarios $\boldsymbol{v} \in \boldsymbol{V}$ tienen bajos ingresos $[\boldsymbol{v} \in \boldsymbol{H_1}]$ y cuáles tienen altos ingresos $[\boldsymbol{v} \in \boldsymbol{H_2}]$.

Selección Aleatoria

El método de selección aleatoria simplemente elige una categoría al azar.

$$P(v \in H_1) = 1/2$$
$$P(v \in H_2) = 1/2$$

Votación Mayoritaria

El método de votación mayoritaria elige la categoría de cada usuario como la categoría a la que pertenecen la mayoría de sus contactos. En caso de empate, se elige una categoría al azar.

$$\boldsymbol{P}\left(\boldsymbol{v} \in \boldsymbol{H_1}\right) = \begin{cases} \mathbf{0} & \text{si } \mathbf{contacts}_{\boldsymbol{v}}^{\mathbf{low}} < \mathbf{contacts}_{\boldsymbol{v}}^{\mathbf{high}} \\ \mathbf{1/2} & \text{si } \mathbf{contacts}_{\boldsymbol{v}}^{\mathbf{low}} = \mathbf{contacts}_{\boldsymbol{v}}^{\mathbf{high}} \\ \mathbf{1} & \text{si } \mathbf{contacts}_{\boldsymbol{v}}^{\mathbf{low}} > \mathbf{contacts}_{\boldsymbol{v}}^{\mathbf{high}} \end{cases}$$

Generation of Graph Features

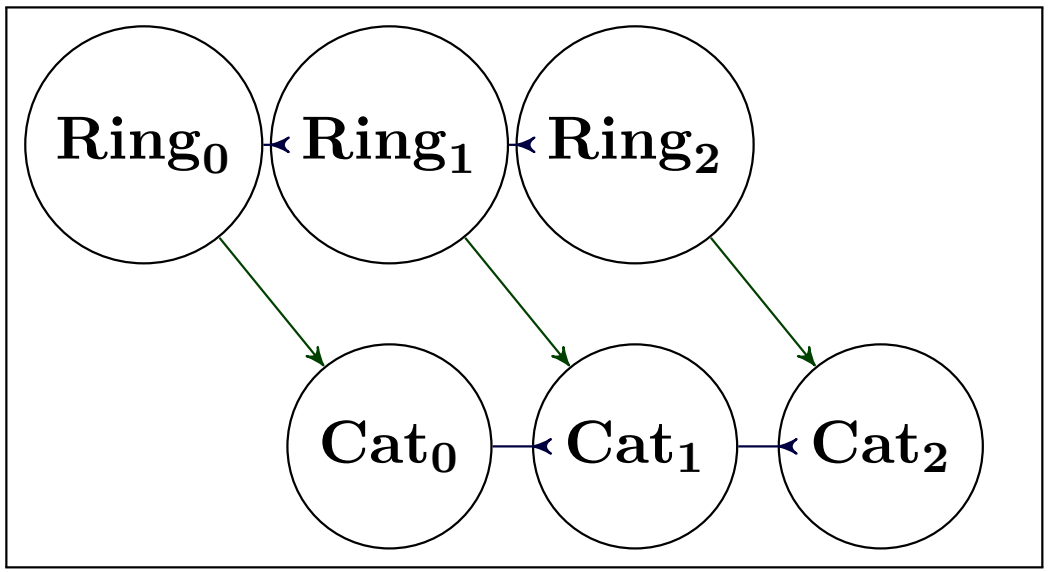
For each link $e \in E$ in the graph we have:

- **Origin** of the calls and SMS
- **Destination** of the calls and SMS
- **Calls**: total number of calls
- **Time**: total time (in seconds) of all the calls
- **SMS**: total amount of messages

Métodos de Extracción de Features en un Grafo

A continuación se presentan 6 métodos de extracción de features para el *grafo social* G.

- Los métodos $\mathbf{Ring}_{\{0,1,2\}}$, que usan datos sobre las aristas adyacentes a n niveles del *ego network* de cada nodo.
- Los métodos $\mathbf{Cat}_{\{0,1,2\}}$, que separan estos datos en diferentes categorías dependiendo del nivel socioeconómico de cada vecino.



Relaciones entre los métodos de extracción de features.

User Data — Método $\mathbf{Ring_0}$

Este método acumula diferentes features de las aristas de cada usuario.

$$\mathbf{incalls}_v = \sum_{\substack{e \in E \\ e_d = v}} \mathbf{calls}_e \quad \mathbf{outcalls}_v = \sum_{\substack{e \in E \\ e_o = v}} \mathbf{calls}_e$$

$$\mathbf{intime}_v = \sum_{\substack{e \in E \\ e_d = v}} \mathbf{time}_e \quad \mathbf{outtime}_v = \sum_{\substack{e \in E \\ e_o = v}} \mathbf{time}_e$$

$$\mathbf{insms}_v = \sum_{\substack{e \in E \\ e_d = v}} \mathbf{sms}_e \quad \mathbf{outsms}_v = \sum_{\substack{e \in E \\ e_o = v}} \mathbf{sms}_e$$

$$\mathbf{incontacts}_v = |\{e \in E \mid e_d = v\}|$$

$$\mathbf{outcontacts}_v = |\{e \in E \mid e_o = v\}|$$

Categorical User Data — Método $\mathbf{Cat_0}$

Los nodos $\boldsymbol{\Upsilon} \subseteq \boldsymbol{V}$ en los que se va a evaluar este métodos contiene información bancaria de los usuarios.
Esto permite crear features con los siguientes nombres.

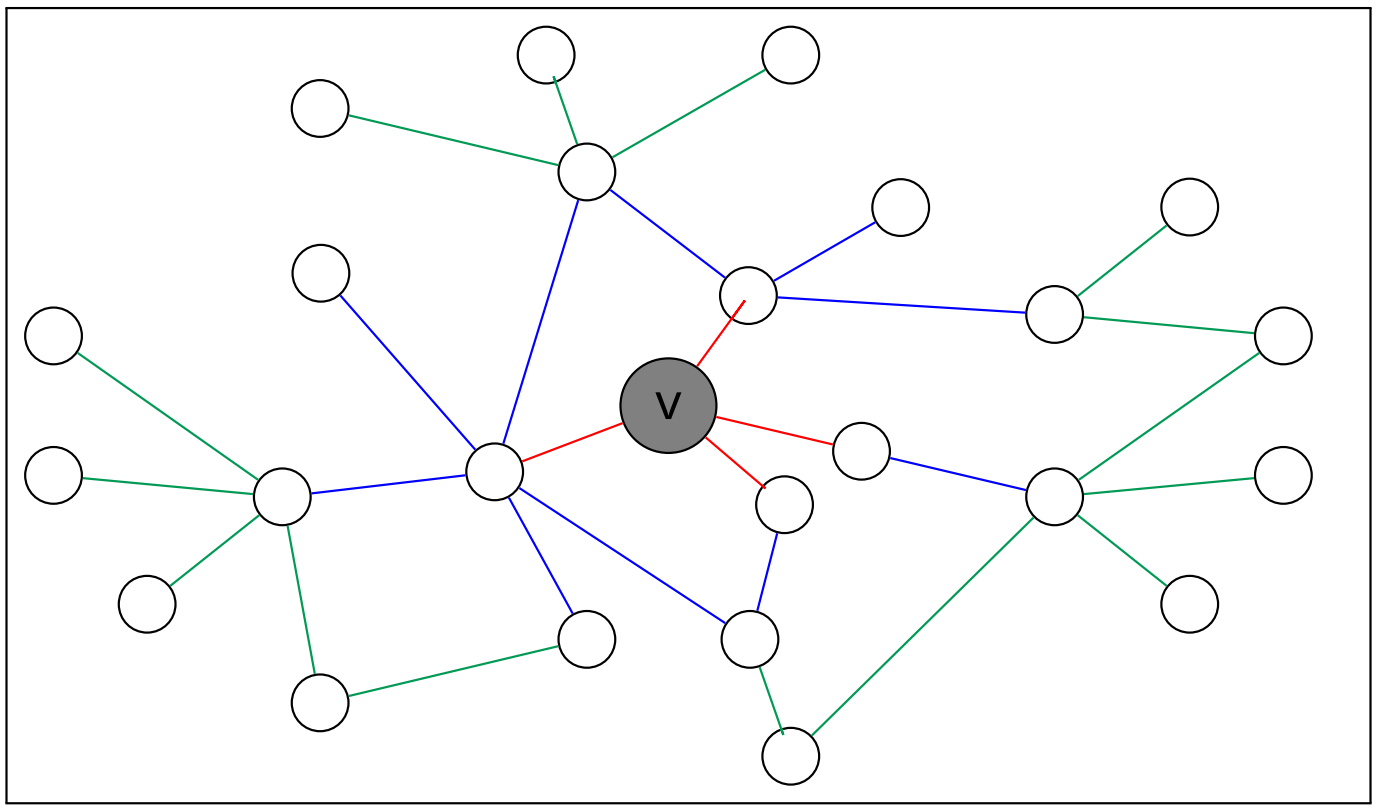
$$\left\{ \begin{matrix} \text{in} \\ \text{out} \end{matrix} \right\} \times \left\{ \begin{matrix} \text{calls} \\ \text{time} \\ \text{sms} \\ \text{contacts} \end{matrix} \right\} \times \left\{ \begin{matrix} \text{low} \\ \text{high} \end{matrix} \right\}$$

Todos estos features se general de una manera similar a la siguiente ecuación.

$$\mathbf{outcallslow}_{\boldsymbol{v}} = \sum_{\substack{\boldsymbol{e} \in \boldsymbol{E} \\ \boldsymbol{e_d} \in \boldsymbol{H_1} \\ \boldsymbol{e_o} = \boldsymbol{v}}} \mathbf{calls}_{\boldsymbol{e}} \quad \mathbf{outcallshigh}_{\boldsymbol{v}} = \sum_{\substack{\boldsymbol{e} \in \boldsymbol{E} \\ \boldsymbol{e_d} \in \boldsymbol{H_2} \\ \boldsymbol{e_o} = \boldsymbol{v}}} \mathbf{calls}_{\boldsymbol{e}}$$

Higher Order User Data — Método $\mathbf{Ring}_n$

El *Ego Network de Orden* n de un nodo v contiene el nodo v, y todos los nodos y las aristas a los que tienen distancia $\leq n$ a v.



Los ejes que se usan al calcular el método $\mathbf{Ring_2}$ de un nodo v. Las aristas rojas son las aristas usadas en $\mathbf{Ring}_{n\geq 0}$, las azules las usadas en $\mathbf{Ring}_{n\geq 1}$, y las verdes los que se usan en $\mathbf{Ring}_{n\geq 2}$.

Higher Order User Data — Método $\mathbf{Ring}_n$

Se extienden los features del método $\mathbf{Ring_0}$ con datos del *Ego Network de Orden* $\boldsymbol{n}$ de $\boldsymbol{v}$.

$$\mathbf{incalls}_{\boldsymbol{v}}^{\boldsymbol{n}} = \sum_{\substack{\boldsymbol{e}\in\boldsymbol{E} \\ \boldsymbol{d}(\boldsymbol{e_o},\boldsymbol{v})=\boldsymbol{n} \\ \boldsymbol{d}(\boldsymbol{e_d},\boldsymbol{v})=\boldsymbol{n}+\mathbf{1}}} \mathbf{calls}_{\boldsymbol{e}} \qquad \mathbf{outcalls}_{\boldsymbol{v}}^{\boldsymbol{n}} = \sum_{\substack{\boldsymbol{e}\in\boldsymbol{E} \\ \boldsymbol{d}(\boldsymbol{e_d},\boldsymbol{v})=\boldsymbol{n} \\ \boldsymbol{d}(\boldsymbol{e_o},\boldsymbol{v})=\boldsymbol{n}+\mathbf{1}}} \mathbf{calls}_{\boldsymbol{e}}$$

Categorical Higher Order User Data — Método $\mathbf{Cat}_n$

Se extienden los features del método $\mathbf{Cat_0}$ con datos del *Ego Network de Orden* n de v, donde cada arista agrega diferentes valores para los vecinos de bajo y alto nivel socioeconómico.

$$\mathbf{incallslow}_v^n = \sum_{\substack{e \in E \\ e_d \in H_1 \\ d(e_o,v)=n \\ d(e_d,v)=n+1}} \mathbf{calls}_e \qquad \mathbf{incallshigh}_v^n = \sum_{\substack{e \in E \\ e_d \in H_2 \\ d(e_o,v)=n \\ d(e_d,v)=n+1}} \mathbf{calls}_e$$

$$\mathbf{outcallslow}_v^n = \sum_{\substack{e \in E \\ e_o \in H_1 \\ d(e_d,v)=n \\ d(e_o,v)=n+1}} \mathbf{calls}_e \qquad \mathbf{outcallshigh}_v^n = \sum_{\substack{e \in E \\ e_o \in H_2 \\ d(e_d,v)=n \\ d(e_o,v)=n+1}} \mathbf{calls}_e$$

Métodos de Machine Learning

Cada uno de estos conjuntos de features es entrenado usando uno de estos métodos de aprendizaje automático y grid search, y luego evaluado el resultado en Υ haciendo 5-fold cross validation.

- *Regresión Logística*, eligiendo el coeficiente regulador C en incrementos exponenciales.

$$C \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$$

- *Random Forest*, con alguno de los siguientes hiperparámetros: $\mathbf{Criterion}, \mathbf{Features}, \mathbf{Replacement}$.

Agenda

Resultados

El método bayesiano, los 2 métodos triviales, y los 2 métodos de aprendizaje automático aplicados a los 6 métodos de extracción de features se entrenaron con el mismo conjunto de datos y fueron evaluados en un server con las siguientes propiedades.

- Intel Xeon D-1540 con 2GHZ y 128GByte de RAM.
- Numpy 1.12.1
- Scipy 0.18.1
- Pandas 0.19.2
- Scikit-learn 0.18

Resultados – Inner graph

Model	Level	AUC	F_1-score	F_4-score
Random	**Selection**	0.499	0.500	0.500
Majority	**Voting**	0.681	0.721	0.712
Bayesian	**Algorithm**	**0.746**	**0.723**	**0.783**
LR	$\mathbf{Ring_1}$	0.536	0.574	0.619
	$\mathbf{Ring_2}$	0.535	0.611	0.714
	$\mathbf{Ring_3}$	0.569	0.550	0.528
	$\mathbf{Cat_1}$	0.686	0.714	0.776
	$\mathbf{Cat_2}$	0.693	0.718	0.772
	$\mathbf{Cat_3}$	0.692	0.714	0.758
RF	$\mathbf{Ring_1}$	0.548	0.549	0.550
	$\mathbf{Ring_2}$	0.582	0.580	0.577
	$\mathbf{Ring_3}$	0.576	0.579	0.580
	$\mathbf{Cat_1}$	0.671	0.677	0.688
	$\mathbf{Cat_2}$	0.714	0.714	0.716
	$\mathbf{Cat_3}$	0.709	0.711	0.711

Resultados

Modelo	Features	Acc.	Prec.	Rec.	AUC	F_1	F_4	t_{fit}	t_{pred}
Bayesiano		0.693	0.665	0.792	**0.746**	**0.723**	**0.783**	—	33.155 s
Aleatorio		0.499	0.499	0.500	0.499	0.500	0.500	—	0.005 s
Mayoría		0.681	0.640	0.826	0.681	0.721	0.712	—	0.059 s
LR	$\mathbf{Ring_0}$	0.536	0.531	0.625	0.536	0.574	0.619	0.145 s	0.002 s
	$\mathbf{Ring_1}$	0.535	0.525	0.730	0.535	0.611	0.714	0.141 s	0.011 s
	$\mathbf{Ring_2}$	0.568	0.578	0.525	0.569	0.550	0.528	0.119 s	0.003 s
	$\mathbf{Cat_0}$	0.686	0.655	0.785	0.686	0.714	0.776	0.167 s	0.005 s
	$\mathbf{Cat_1}$	0.693	0.665	0.780	0.693	0.718	0.772	1.588 s	0.011 s
	$\mathbf{Cat_2}$	0.693	0.670	0.764	0.692	0.714	0.758	0.956 s	0.009 s
RF	$\mathbf{Ring_0}$	0.548	0.548	0.550	0.548	0.549	0.550	5.986 s	0.588 s
	$\mathbf{Ring_1}$	0.582	0.583	0.577	0.582	0.580	0.577	56.548 s	0.483 s
	$\mathbf{Ring_2}$	0.576	0.577	0.580	0.576	0.579	0.580	50.197 s	0.253 s
	$\mathbf{Cat_0}$	0.671	0.665	0.690	0.671	0.677	0.688	6.346 s	0.539 s
	$\mathbf{Cat_1}$	0.714	0.713	0.716	0.714	0.714	0.716	96.005 s	0.460 s
	$\mathbf{Cat_2}$	0.709	0.710	0.711	0.709	0.711	0.711	81.528 s	0.242 s

References

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. ISSN 1742-5689. doi: 10.1098/rsif.2016.0598.

Carlos Sarraute, Carolina Lang, Nicolas B Ponieman, and Sebastian Anapolsky. The city pulse of Buenos Aires. In *Workshop Big Data & Environment*, 2015.

Carlos Sarraute, Pablo Blanc, and Javier Burroni. A study of age and gender seen through mobile phone usage patterns in Mexico. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 836–843. IEEE, 2014.

Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

Nicolas Ponieman, Alejo Salles, and Carlos Sarraute. Human mobility and predictability enriched by social phenomena information. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1331–1336. ACM, 2013.

Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the Facebook social graph. *Structure*, 5:6, 2011.

Jorge Brea, Javier Burroni, Minnoni Martin, and Carlos Sarraute. Harnessing mobile phone social network topology to infer users demographic attributes. In *ACM SIGKDD*. ACM, 2014.