EPJ B
Condensed Matter and Complex Systems

EPJ.org
your physics journal

Eur. Phys. J. B (2015) 88: 194 DOI: 10.1140/epjb/e2015-60357-1

Statistical analysis of Nomao customer votes for spots of France

Róbert Pálovics, Bálint Daróczy, András Benczúr, Julia Pap, Leonardo Ermann, Samuel Phan, Alexei D. Chepelianskii and Dima L. Shepelyansky

edp sciences
Società Italiana di Fisica
Springer

Eur. Phys. J. B (2015) 88: 194
DOI: 10.1140/epjb/e2015-60357-1

THE EUROPEAN
PHYSICAL JOURNAL B

Regular Article

Statistical analysis of Nomao customer votes for spots of France

Róbert Pálovics[1,2], Bálint Daróczy[1,2], András Benczúr[1,3], Julia Pap[1], Leonardo Ermann[4], Samuel Phan[5], Alexei D. Chepelianskii[6], and Dima L. Shepelyansky[7,a]

[1] Informatics Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Pf. 63, 1518 Budapest, Hungary
[2] Technical University Budapest, 1111 Budapest, Hungary
[3] Eötvös University Budapest, 1053 Budapest, Hungary
[4] Departamento de Física Teórica, GIyA, CNEA, Av. Libertador 8250, C1429BNP Buenos Aires, Argentina
[5] Nomao.com, 1 av Jean Rieux, 31500 Toulouse, France
[6] LPS, Université Paris-Sud, CNRS, UMR 8502, 91405, Orsay, France
[7] Laboratoire de Physique Théorique du CNRS, IRSAMC, Université de Toulouse, UPS, 31062 Toulouse, France

Received 5 May 2015
Published online 3 August 2015 – © EDP Sciences, Società Italiana di Fisica, Springer-Verlag 2015

Abstract. We investigate the statistical properties of votes of customers for spots of France collected by the startup company Nomao. The frequencies of votes per spot and per customer are characterized by a power law distribution which remains stable on a time scale of a decade when the number of votes is varied by almost two orders of magnitude. Using the computer science methods we explore the spectrum and the eigenvalues of a matrix containing user ratings to geolocalized items. Eigenvalues nicely map to large towns and regions but show certain level of instability as we modify the interpretation of the underlying matrix. We evaluate imputation strategies that provide improved prediction performance by reaching geographically smooth eigenvectors. We point on possible links between distribution of votes and the phenomenon of self-organized criticality.

1 Introduction

The young startup company Nomao [1] collected a large database about customer (or user) votes for spots (or Points of Interest POIs or items) in France. The spots represent mainly restaurants and hotels with known geolocation coordinates. In this paper we investigate the statistical properties of these Nomao votes and ratings of geolocalized items in a mix of geographic information and recommendations systems. The geographical distributions of votes are shown in Figure 1 for the whole France and more specifically for Paris. The frequency distributions of votes per spot and votes per user are shown in Figure 2 for France at different time intervals. It shows that these frequency distributions are stabilized in time and thus we are dealing with an unusual statistical system been at a certain steady-state. We note that at present a variety of real systems and networks are found to possess power law distribution (see e.g. Ref. [2]) and thus here we investigate a new type of such a case with algebraic statistical properties.

To analyze the statistical properties of this real system we use the methods of recommender systems [3] which gained a broad recognition in computer science after the Netflix Prize competition [4]. In our research, distance, region and location become a side information over a multi-objective classification or regression problem. We concentrate on predicting user preferences by a spectral analysis based collaborative filtering that uses geo-location in addition to the ratings matrix.

We investigate how user taste, as described by latent factors, is reflected in the geographic information system. We compare the latent factors obtained by a full spectral analysis and by the stochastic gradient method, the standard recommendation technique [3] applicable for matrices with a very large fraction of missed values.

The key difficulty in the spectral analysis lies in the abundance of missing values in the rating matrix: our matrix consists of 99.5% missing values while the Netflix matrix for example is 99% unknown. Several early results describe expectation maximization based Singular Value Decomposition (SVD) algorithms, dating back to the seventies [5–8], describe the method for a recommender application.

A successful implementation of spectral analysis in recommender matrices with only a few known elements is described by Funk in reference [9]. His method is a variant of Stochastic Gradient Descent (SGD) reminiscent of gradient boosting [10]. SGD computes no eigenvalues and does

[a] e-mail: dima@irsamc.ups-tlse.fr

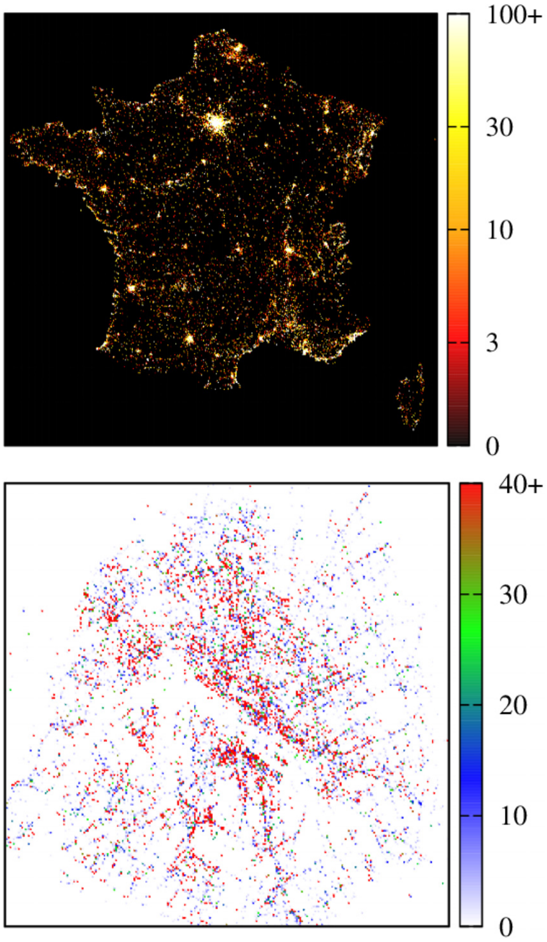

Fig. 1. Geographical distribution of votes for spots (POIs) in the original datasets. Top panel: case of France; (each square pixel represents 7.8 km^2); bottom panel: case of Paris (each square pixel represents 1370 m^2); color bars give a number of votes per pixel (cell), a limitation in number of votes is introduced for a better color representation.

not guarantee the orthogonality of the matrix factors. On the other hand, regularization is easily incorporated in SGD, which enables a better handling of the very large amount of missing values in the matrix and in particular, prevents overfit to training elements and provide better quality predictions of the unknown ratings.

In this paper, after describing methods and related results (Sect. 2) and the Nomao data sets (Sect. 3), we compare and visualize the geo-localization of the matrix factors defined by SVD and SGD under various parameter settings in Section 4. We show that by imputing ratings to nearby locations we may form factors that yield a better description of the ratings matrix in Section 5.

2 Methods and related results

Recommenders based on the rank k approximation of the rating matrix with the first k singular vectors are probably first described in references [11–14] and many others near year 2000.

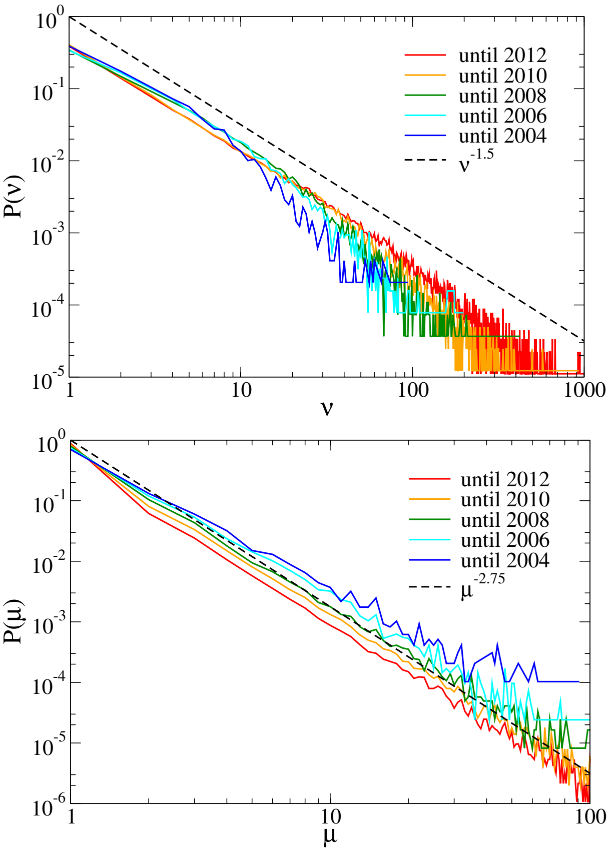

Fig. 2. Differential frequency distributions of votes for the case of France for different time intervals until year 2004, 2006, 2008, 2010, 2012. Top panel: differential probability $P(\nu)$ to have ν votes for spots (POIs); bottom panel: differential probability $P(\mu)$ to have μ votes per user (customer). Here the dashed lines show an average algebraic decay with exponents $a = 1.5$ (top), $b = 2.75$ (bottom).

The SVD of a rank ρ matrix R is given by $R = U^T \Sigma V$ with U an $m \times \rho$, Σ a $\rho \times \rho$ and V an $n \times \rho$ matrix such that U and V are orthogonal. By the Eckart-Young theorem [15] the best rank-k approximation of R with respect to the Frobenius norm is:

$$||R - U_k^T \Sigma_k V_k||_F^2 = \sum_{ij} \left(r_{ij} - \sum_k \sigma_k u_{ki} v_{kj} \right)^2, \tag{1}$$

where U_k is an $m \times k$ and V_k is an $n \times k$ matrix containing the first k columns of U and V and the diagonal Σ_k containing first k entries of Σ.

The RMSE differs from the above equation only in that summation is over known ratings

$$\text{RMSE}^2 = \sum_{ij \in \text{known}} \text{err}_{ij}^2 \text{ where } \text{err}_{ij} = r_{ij} - \sum_k \sigma_k u_{ki} v_{kj}. \tag{2}$$

Early works [11] used SVD for recommenders by defining various strategies for handling the missing values in the rating matrix R [16]. The most natural idea is to impute the missing elements by zeroes, averages, or even repeatedly re-fill by predictions. It has turned out that all above missing value imputation methods overfit to the imputed values [16]. More recent results emphasize the importance of regularization to avoid overfitting [17,18]. For this reason, the recommender systems community turned away from SVD and use other optimization methods for rating matrices with missing values, most notably stochastic gradient descent [19] and alternating least squares [20].

In our problem, locality is an additional information that can be exploited for analyzing the recommender matrix. Surveys on recommendations in location-based social networks [21,22] combine spatial ratings for non-spatial items, nonspatial ratings for spatial items, and spatial ratings for spatial items [23]. Flickr geotags are used for travel route recommendation, concentrating on routes and not individual places in reference [24]. User similarity based methods may combine friendship information with the distance of the user home locations [25,26].

Most similar to our method is the probabilistic matrix factorization approach that fuses geographic information [27] and observes that "users tend to check in around several centers, where the check-in locations follow a Gaussian distribution at each center [... and] the probability of visiting a place is inversely proportional to the distance from its nearest center; if a place is too far away from the location a user lives, although he/she may like that place, he/she would probably not go there".

3 The Nomao datasets

Nomao is a startup company located in France [1]. It performs the analysis of point of interest (POI) rating and reservation services and collects POI information including user ratings from France with a special accent on Paris and Toulouse regions where the company headquarters are located. The Nomao dataset used in our experiments contain user-POI ratings, and GPS information of the rated POIs. We investigate two separated datasets. The first one contains information on POIs in France, while the second has ratings only on POIs located in Paris. We analyze the datasets collected during the time period up to year 2012.

Table 1 (top) shows the basic attributes of the original datasets. The average number of ratings per item is relatively large, the average number of ratings per user is very low. Moreover, only a very few percent of all user-item scores is known.

The distribution $P(\nu)$ of frequency of votes per spot ν (or item i) for France dataset is shown in the top panel of Figure 2. This distribution is stable in time and is well described by the power law $P(\nu) \propto 1/\nu^a$ with $a \approx 1.5$. Also, the distribution $P(\mu)$ of frequency of votes per customer μ (or user u) remains stable in time with the power law dependence $P(\mu) \propto 1/\mu^b$ with $b \approx 2.75$. It is important to note that these distributions remain stable from year 2004 up to year 2012 even if the number of votes increases almost by two orders of magnitud during this period. At the moment we cannot provide theoretical reasons for the values of these exponents.

Table 1. Attributes of the original (top), and cleaned (bottom) datasets.

Original	Paris	France
Number of ratings	1 539 964	1 432 601
Number of users	998 127	1 077 568
Number of items	20 576	99 976
Average ratings per user	1.543	1.329
Average ratings per item	74.84	14.32
Ratio of known ratings	0.0075%	0.0013%
Cleaned	Paris	France
Number of ratings	114 352	97 452
Number of users	5756	9471
Number of items	2952	7605
Average ratings per user	19.87	10.29
Average ratings per item	38.74	12.81
Ratio of known ratings	0.672%	0.135%
Average rating	3.714	3.747

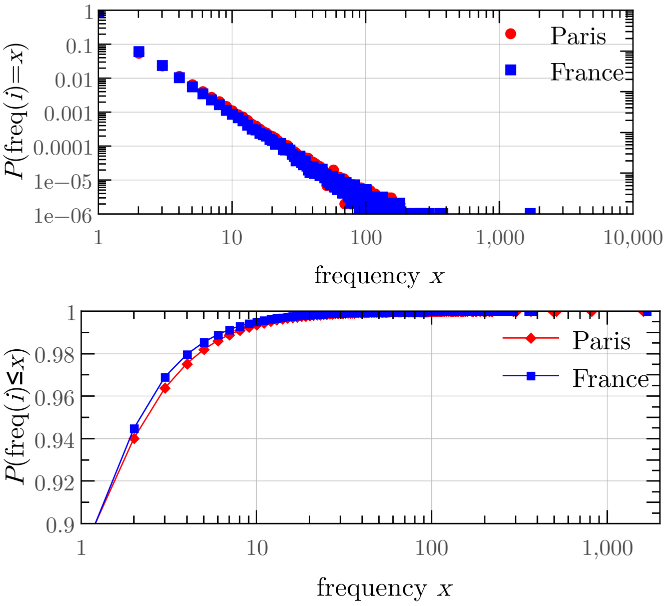

Fig. 3. Probability density function (top), and cumulative density function (bottom) of user activities in the original datasets.

We call user activity how many times a user scored different items. We define item activity similarly. Figure 3 shows the probability density function (PDF), an the cumulative density function (CDF) of user activities. Figure 4 shows the same distributions for items. Both user and item activities follow power-law distributions with the exponent values being very similar for France and Paris datasets. As in Figure 2 we find that the exponent for probability of votes for POIs is $a \approx 1.5$ while the exponent for the exponent for probability of votes of users is $b \approx 2.75$.

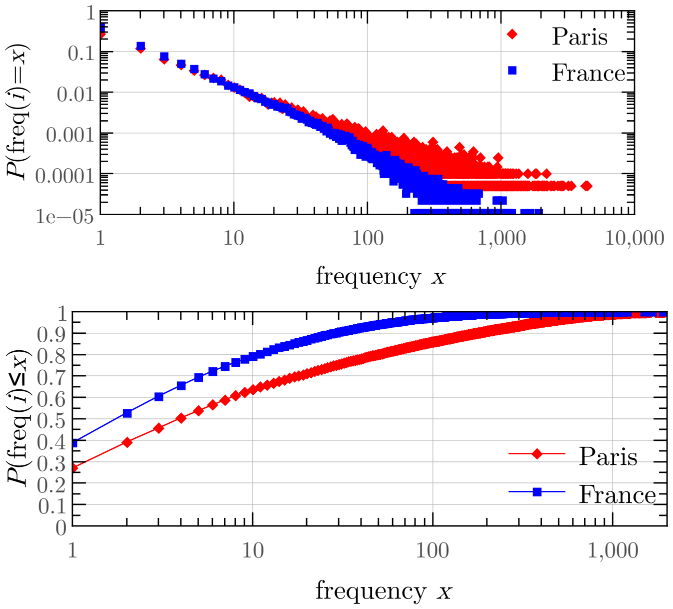

Fig. 4. Probability density function (top), and cumulative density function (bottom) of item activities in the original datasets.

To handle the extreme sparsity of the user-item matrices, we selected a smaller subset of the user-item rating datasets by the following selection criteria:

- We only used ratings between 0–5. Part of the ratings, probably originating from a different system, were out of this range.
- We filtered out users and items that have less than A ratings. In other words, we selected the subgraph of the user-item rating bipartite graph with users and items that have degree at least A. For Paris we set $A = 10$, for France we set $A = 5$.

Table 1 (bottom) shows the attributes of the selected subsets. In what follows we use these datasets in our experiments.

In Figure 5 we show the score distributions: the top (bottom) panel shows the distributions for the original (cleaned) datasets. We see that the original and cleaned datasets have similar distributions of scores.

Figure 6 shows the geographical density of POIs in France (top) and Paris (bottom) for the original datasets. The geolocation data of POIs are used in the following sections for spectral analysis.

In the following we perform all computations with the cleaned datasets since the analysis of multiple votes of the same user provides more reliable statistical data.

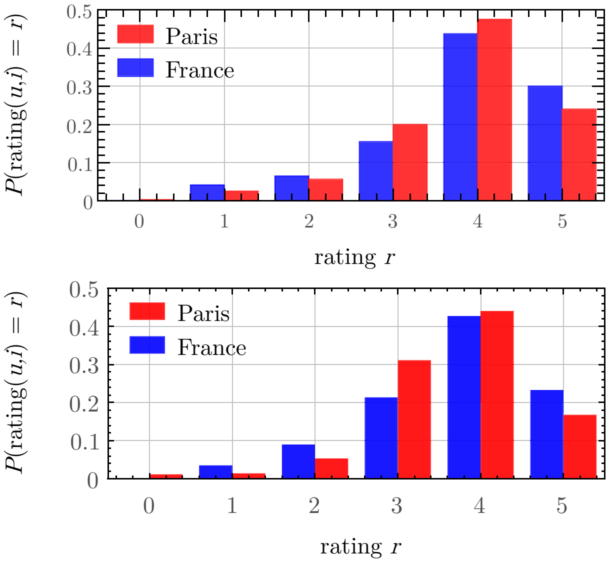

Fig. 5. Score distributions with integer binning. Top: original dataset, bottom: cleaned dataset.

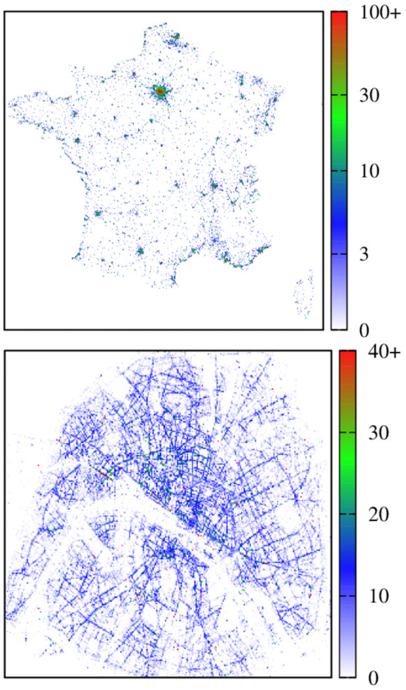

Fig. 6. Geographical distribution of POIs in the original datasets. Top panel: case of France (each square pixel represents 7.8 km^2); bottom panel: case of Paris (each square pixel represents 1370 m^2); color bars give number of POIs per pixel (cell); a limitation in number of POIs is introduced for a better color representation.

4 Spectra of recommender matrices

4.1 Singular value decomposition

The recommender matrix R consists of the preference values $r(u, i)$ of users u for items i. The values may denote *explicit* rating values, e.g. 1–5 stars for Netflix movies [17]. We may also consider the so-called *implicit* ratings problem, where the value is 1 if the user visited POI i and 0 otherwise. The value of the explicit matrix is missing whenever the user gave no rating yet. In most of the cases, this matrix is very sparse with only 1% or less known values. The implicit matrix is always a full 0–1 matrix, however the 0 values are uncertain: the user may not know about the item or had no time yet to visit it.

The so-called *Latent Factor Model* is an approximation $\hat{R}$ of the original rating matrix R,

$$\hat{r}(u,a) = \sum_{f=1}^{k} p_{uf} q_{af}, \tag{3}$$

where $P = [p_{uf}]$ and $Q = [p_{uf}]$ are the user and item factor models, respectively.

For a fixed number of factors k, $\hat{r}$ approximates r with the smallest root mean squared error if it is defined by the singular vectors corresponding to the k largest singular values,

$$\hat{r}(u,a) = \sum_{f=1}^{k} p_{uf} q_{af}, \tag{4}$$

where the singular value decomposition (SVD) of R is $U\Sigma V^T$.

Since

$$RR^T = U\Sigma^2 U^T \text{ and } R^T R = V\Sigma^2 V^T, \tag{5}$$

the spectrum of the recommender matrix R is defined identically by the square root of the eigenvalues of RR^T or $R^T R$. These latter matrices are symmetric positive semidefinite, the spectrum is non-negative real.

If R contains missing values such as in the case of an explicit rating matrix, SVD is undefined. We may still define the best root mean square approximation by summing the error for the known ratings only as in equation (1).

4.2 Stochastic gradient descent: latent factor modeling with missing values

We use the regularized matrix factorization method of [18] and optimize the minimum squared error of the k-factor model

$$\hat{r}(u,i) = \sum_{l=1}^{k} p_{ul} q_{il}, \tag{6}$$

where p and q contain the user and item models, respectively. By adding regularization with weight λ, we optimize the quantity

$$\sum_{u,i}\left(r(u,i) - \sum_{l=1}^{k} p_{ul} q_{il}\right)^2 + \lambda \sum_{u}\sum_{l=1}^{k} p_{ul}^2 + \lambda \sum_{i}\sum_{l=1}^{k} q_{il}^2. \tag{7}$$

For a single event (u,i) we optimize the coefficients p_{ul} and q_{il} for $l = 1, \ldots, k$ by gradient descent with learning rate lrate as:

$$p_{ul} \leftarrow p_{ul} + \text{lrate} \cdot \left(r(u,i) - \sum_{l=1}^{k} p_{ul} q_{il}\right) q_{il} - \text{lrate} \cdot \lambda p_{ul}; \tag{8}$$

$$q_{il} \leftarrow q_{il} + \text{lrate} \cdot \left(r(u,i) - \sum_{l=1}^{k} p_{ul} q_{il}\right) p_{ul} - \text{lrate} \cdot \lambda q_{il}. \tag{9}$$

Unlike SVD where eigenvalues are sorted, the SGD factors are not ordered by the above equations. In order to produce the eigenvector maps, we built ranked factors by an iterative SGD that optimize only on a single factor at a time [9].

4.3 Mapping SVD and SGD latent factors

First we set each unknown value of R to zero and computed the SVD decomposition. The first, second, and fourth singular vectors are plotted over the map of France (Fig. 7, left) by assigning the value in the vector to the location of the POI. More specifically, we averaged these values on a grid to create the final heatmaps. The smoothing algorithm weighted the value of each POI to the closest grid point inversely proportional to their euclidean distance.

The heatmaps in Figure 7, left, indicate that the singular vectors are strongly geolocation related. The first few dimensions correspond to the largest cities in France.

Similarly, we investigated the latent vectors of R computed with the SGD algorithm. The first, second and fourth latent vectors are plotted over the map of France in Figure 7, right, similar to the SVD eigenvectors. While the SVD singular vectors were centralized one-by-one on a large city, the SGD latent factors are the linear combination of them. The latent factors are also geolocation related, but not separated among the main cities like the SVD singular vectors.

In Figure 8 we mapped the first three singular vectors of the Paris dataset. The different vectors may focus on different districts. However, they are not as clearly separated as the singular vectors of the France dataset.

In Section 5 we use these key observations to improve the recommendation quality of the SGD.

5 Prediction for ratings and visits to locations

5.1 Recommender evaluation

Recommender systems serve to find *new* products for the users that are relevant for them. More specifically, for a given user u an item i a recommender system may retrieve the predicted relevancy $\hat{r}_{ui}$. This is called the rating prediction task. The Netflix Prize competition [28] was a challenge in rating prediction. While in the Netflix Prize competition, contestants were optimizing to predict all ratings to the users, a recommender system in practice selects the top rated items for a given user. In this top-K prediction task [29–31], a recommender system should retrieve for a given user a top list of items with length K. The top list should contain the most relevant items for the given user. This problem is more application related than the rating prediction task. In our experiments we examine both problems on the Nomao datasets.

In addition to RMSE defined by equations (1) for full and (2) for partial matrices, we use two measures that evaluate the accuracy of the top-K recommendation task.

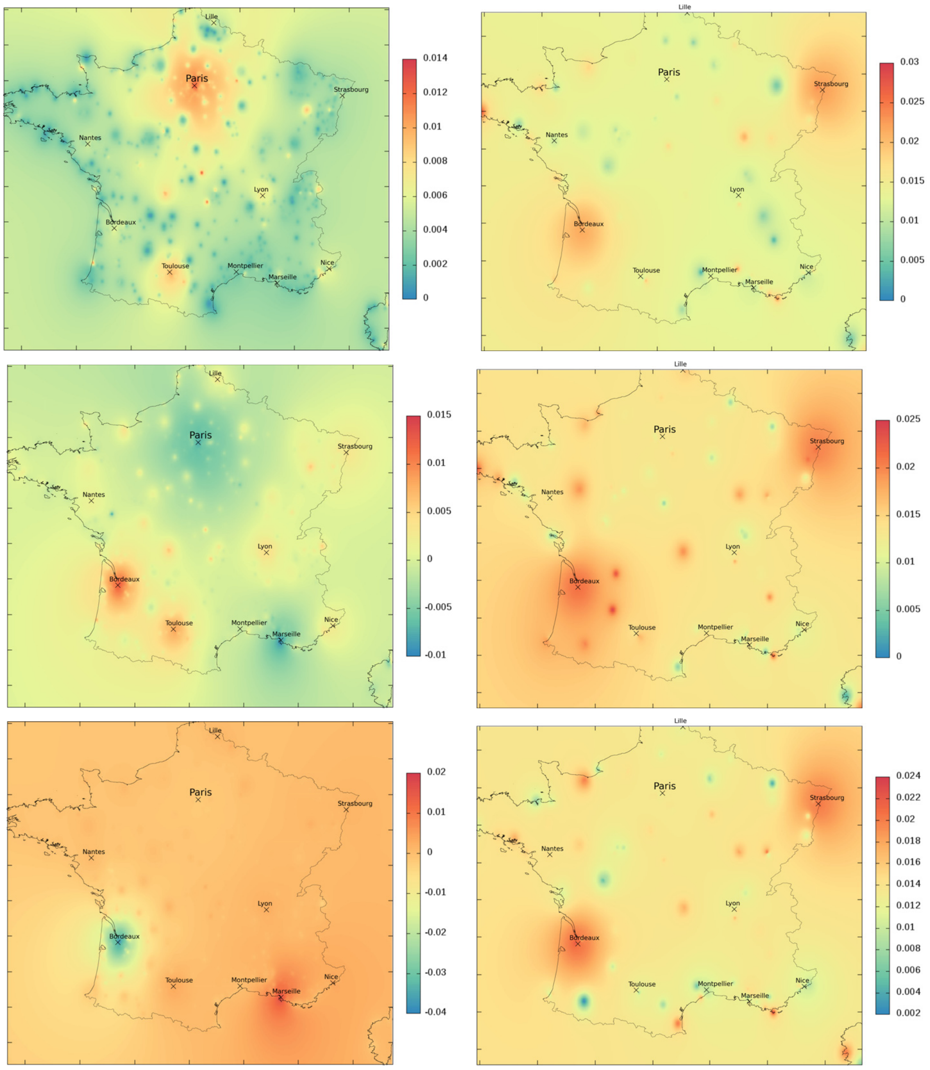

Fig. 7. The first, second and fourth singular vectors of the Nomao France rating matrix by SVD (left) and SGD (right). Here, the SVD eigenvectors correspond to Paris and Toulouse; Bordeaux, Toulouse and Marseille; Bordeaux and Marseille respectively, while the SGD plots for respective vectors are scattered around several cities.

Recall at k is defined as the number of relevant POIs among the highest k values of row u in the matrix approximation,

$$\mathrm{Recall}_u(k) = \frac{1}{|R_u|}\sum_{i=1}^{k} \mathrm{rel}_{u,i}, \tag{10}$$

where $\mathrm{rel}_{u,i}$ is the actual relevance of POI i for user u in the evaluation data, and R_u is the number of relevant items for user u in the dataset. We may average for all users to obtain

$$\mathrm{Recall}(k) = \frac{1}{|U|}\sum_{u} \mathrm{Recall}_u(k). \tag{11}$$

Normalized discounted cumulative gain at k weights the relevance by the order of the predicted values as:

$$\mathrm{NDCG}_u(k) = \frac{\mathrm{DCG}_u(k)}{\mathrm{iDCG}_u(k)}, \tag{12}$$

where

$$\mathrm{DCG}_u(k) = \sum_{i=2}^{k} \frac{\mathrm{rel}_{u,i}}{\log_2(i+1)} \tag{13}$$

and

$$\mathrm{NDCG}(k) = \frac{1}{|U|}\sum_{u} \mathrm{NDCG}_u(k). \tag{14}$$

In our experiments, we randomly cut the data to training and test sets. We only use records in the training set to set the parameters of our model. The lower MSE, and the higher NDCG and recall we measure on the test set, the better is our model.

5.2 The rating prediction task

As indicated in Table 1, bottom, and in Figure 5, the scores have a peaked distribution. This indicates first that the rating prediction task makes less sense with these

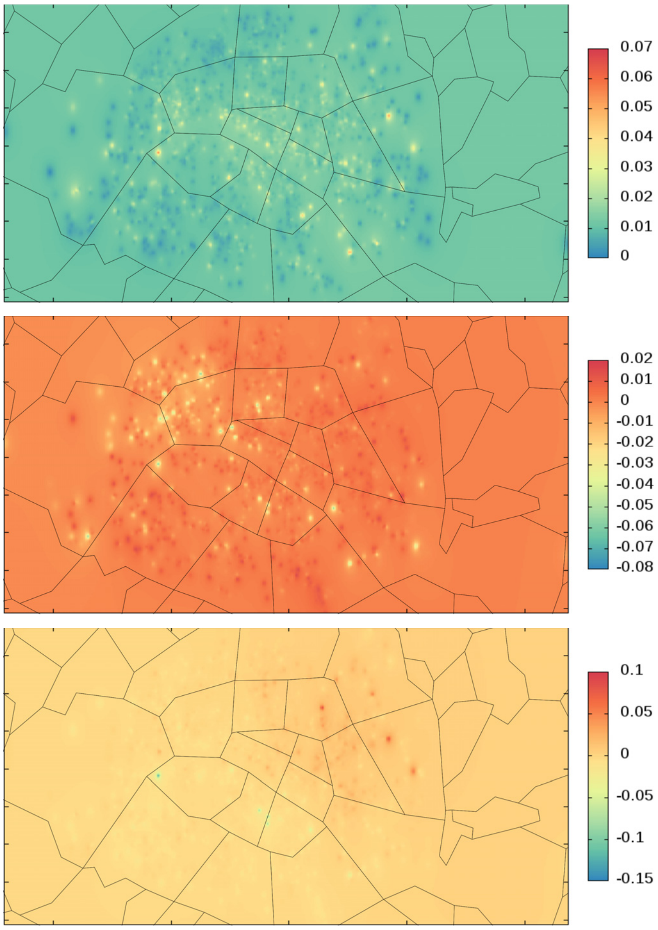

Fig. 8. The first, third and fourth singular vectors of the Nomao Paris rating matrix obtained by SVD.

datasets. We trained up an SGD recommender by using 50% of the datasets and computed NDCG(k) for $K = 1, \ldots, 20$. To understand the performance of the model, we also measured the performance of a random recommender that predicts ratings uniform randomly. We repeated our experiments 10 times with 10 different random training and test sets. Figure 9 shows the computed ten performance curves for the SGD and the baseline random recommendation. Both for SGD and the random prediction the ten curves are similar. This indicates the stability of our algorithms and evaluation metrics. We achieved significantly better result with the SGD recommender. However for the random algorithm, the baseline NDCG is around 0.85. This is due to the fact that most of the ratings are around the mean as the score distribution is peaked.

5.3 Improving top recommendation with rating imputation

Instead of simply recommending locations near to already visited places, we expand the training set by relying on the locality of the ratings. We compare our results by using SVD or SGD both for the rating matrix and for simply predicting the visits, i.e. the existence of a rating regardless of its value. When considering locality, we may identify the nearest neighbors by taking the absolute distance and possibly correcting by density: in an area densely served by POIs, customers may reach more locations, on the other hand, the speed of travel is likely lower than in rural areas.

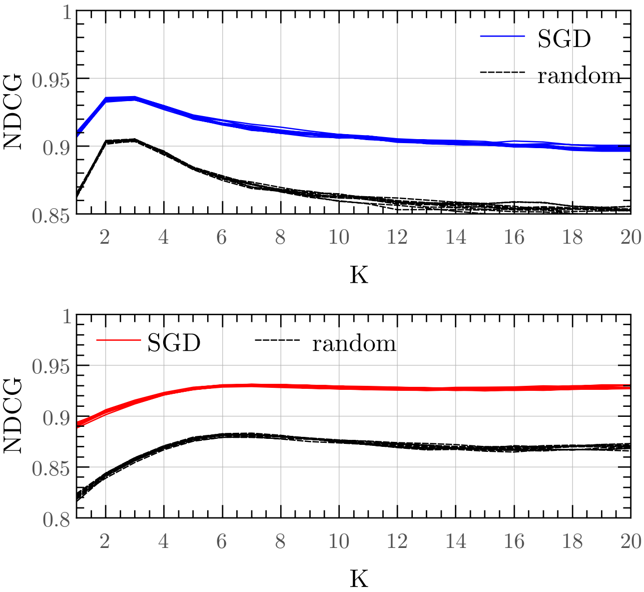

Fig. 9. Performance of score prediction on the France (top) and Paris (bottom) datasets.

For our imputation methods, let E be the set of known ratings and N_j the neighbors of location j. We modify the training set as follows. For all (u, i),

$$\hat{r}_{u,i} = \begin{cases} r_{u,i} & \text{if } (u,i) \in E \\ f(R_u, N_{u,i}) & \text{if } (u,i) \notin E \text{ and for some } j, \\ & (u,j) \in E \text{ and } i \in N_j \\ \text{missing} & \text{otherwise,} \end{cases} \qquad (15)$$

where f is function of R_u, the set of known ratings by user u, and $N_{u,i}$, the set locations visited by u in the neighborhood of i.

In our model, we expand the list of locations per user with the neighbors of visited places by the two strategies:

Constant:

$$f(R_u, N_{u,i}) = c \qquad (16)$$

Ratings Average:

$$f(R_u, N_{u,i}) = \frac{1}{|N_{u,i}|} \sum_{j \in N_{u,i}} r_{u,j}. \qquad (17)$$

The performance for expansion with the original ratings (see Eq. (17)) on the France dataset is seen in Figure 10 where we observe that expansion by the 30–40 nearest

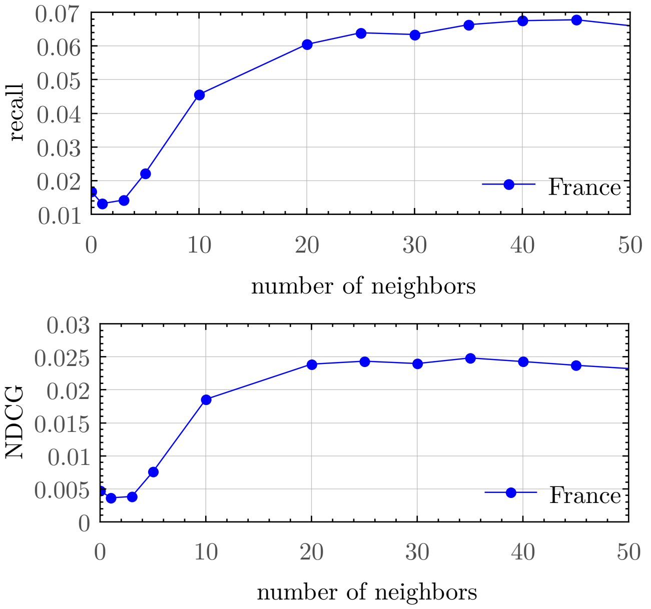

Fig. 10. Recall@100 and NDCG@100 for expansion with the original ratings.

POIs improves significantly the matrix approximation by the first few eigenvectors.

We may also consider the task of predicting which POIs the user will visit, regardless of the actual rating given by the user. In this so-called implicit recommendation task, we consider a 0–1 matrix. Although the matrix is fully known, the meaning of a "1" is certain while a "0" may simply mean that the user has not yet had a chance to visit the POI or does not even know about it. Based on (16), the performance of the implicit task with expansion for the France dataset is seen in Figure 11 showing an improvement compared to Figure 10. However, for the Paris dataset, both in case of the ratings and implicit expansion experiments, we could not improve further the original SVD. This can be due to the fact that the Paris dataset is more dense geographically.

5.4 Improving recommendation with fixed factors

Results of Figure 7 indicate that while SGD finds the most important cities in France, it cannot separate them precisely. Furthermore, not recommending to a user POIs, that he/she have not visited, can be easily implemented without using SGD. Indeed, SGD should learn the taste of the different users like in case of the movie prediction task of Netflix. To fix this issue in the France dataset, we selected the top t cities in France. For a given item, we fixed the first ith factor to 1, if the item is located in the ith city, and 0 otherwise. We set the user factors similarly according to the places visited by the user in the test set. We then trained a k dimensional latent factor model where we updated only the remaining $k-t$ dimensions. We compared this recommender with a traditional k dimensional SGD recommender.

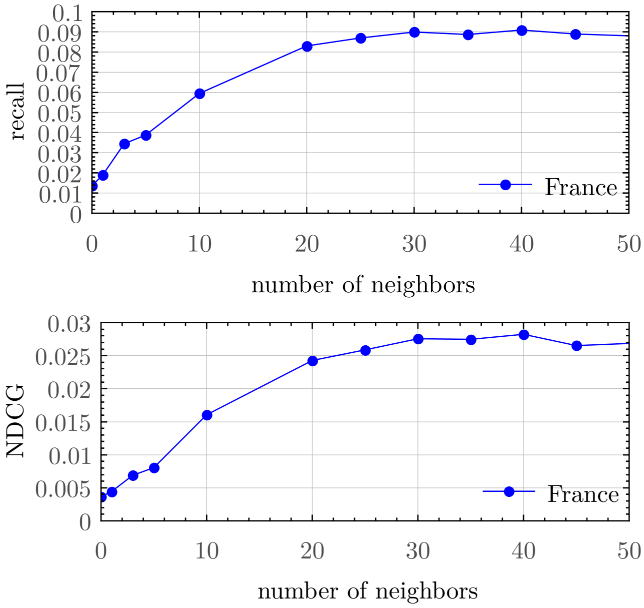

Fig. 11. Recall@100 and NDCG@100 for expansion nearby visited locations.

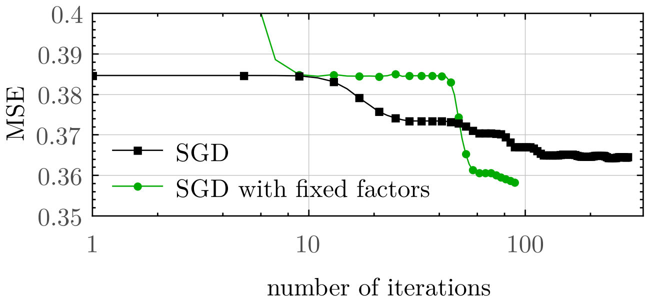

Fig. 12. Improved MSE results with fixed factors on the France dataset.

In our experiments we included the top $t = 10$ cities, in order Paris, Marseille, Lyon, Toulouse, Nice, Nantes, Strasbourg, Montpellier, Bordeaux, and Lille. Figure 12 shows the MSE on the test set as the function of the number of iterations on the training set. With the fixed factor model we can achieve significantly better MSE. Furthermore, our best result is achieved with half less iterations compared to the number needed to train the original latent factor model.

6 Discussion

Our statistical analysis of Nomao votes of customers for spots of France shows that it is described by a power law frequency distributions with exponents $a \approx 1.5$ (for spots) and $b \approx 2.75$ (for customers) which remain stable in time even is the number of votes is increased almost by two orders of magnitude during this time period. Further studies are required to establish the physical origins of such laws and to clarify for universal they are. It is possible that the physical reasons for emergence of such type distributions have certain similarities with the phenomenon of self-organized criticality broadly discussed in physical systems (see e.g. [32–34]). It is interesting to note that the exponent of cluster distribution in self-organized critical models in 3D has an exponent close to 1.4 [32] being not so far from the exponent $a = 1.5$ we find for spots.

We explored the spectrum and the singular vectors of a POI ratings matrix of customer votes for spots of France. The fact that the matrix consists of 99.5% missing values makes the spectrum highly dependent on how we handle the missing values. We computed the SVD of the full 0–1 "implicit" matrix of the visits without considering the rating. For the ratings matrix, we used SGD, a popular approach that uses only the known values to compute the factors. We observed that SGD and SVD factors are similar but SVD has stronger geo-localization. SVD singular vectors with highest eigenvalues are mostly correlated with a particular place. As key practical observations, we found that imputing the missing ratings for the neighbors of visited places could increase the performance, and that defining fixed Geographic factors could improve SGD recommendation quality.

We expect that a broader analysis of a larger number of similar type datasets of votes will allow to gain better understanding of underlying physical process and provide better recommendations for specific customers and spots.

We thank the representatives of Nomao [1] and especially Estelle Delpech (Nomao) for providing us with the friendly access to the Nomao datasets. This research is supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No. 288956).

References

1. Nomao company, http://fr.nomao.com/. Web. (2015)
2. S. Dorogovtsev, *Lectures on Complex Networks* (Oxford University Press, Oxford, 2010)
3. Wikipedia contributors, Recommender system, Wikipedia (2015)
4. Wikipedia contributors, Netflix Prize (2015)
5. J. Canny, Collaborative filtering with privacy via factor analysis, in *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (ACM Press, New York, 2002), p. 238
6. K.R. Gabriel, S. Zamir, Technometrics **21**, 489 (1979)
7. N. Srebro, T. Jaakkola, in *Weighted low-rank approximations*, edited by T. Fawcett, N. Mishra (ICML, AAAI Press, 2003), p. 720
8. S. Zhang, W. Wang, J. Ford, F. Makedon, J. Pearlman, Using singular value decomposition approximation for collaborative filtering, in *CEC '05: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology* (*CEC'05*) (IEEE Computer Society, Washington, 2005), p. 257
9. S. Funk, Netflix update: Try this at home, http://sifter.org/~simon/journal/20061211.html (2006)
10. J.H. Friedman, Ann. Stat. **29**, 1189 (2001)
11. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of dimensionality reduction in recommender systems – a case study, ACM WebKDD Workshop (2000)
12. M.H. Pryor, *The effects of singular value decomposition on collaborative filtering*, Technical report, Dartmouth College, Hanover, 1998
13. D. Billsus, M.J. Pazzani, Learning collaborative information filters, in *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning* (Morgan Kaufmann Publishers Inc., San Francisco, 1998), pp. 46−54
14. D. Gupta, M. Digiovanni, H. Narita, K. Goldberg, Jester 2.0 (poster abstract): evaluation of an new linear time collaborative filtering algorithm, in *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (ACM Press, New York, 1999), p. 291
15. G.H. Golub, C.F.V. Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 1983)
16. M. Kurucz, A.A. Benczúr, K. Csalogány, L. Lukács, Spectral clustering in telephone call graphs, WebKDD/SNAKDD Workshop 2007 in conjunction with KDD 2007 (2007)
17. R.M. Bell, Y. Koren, ACM SIGKDD Explorations Newsletter **9**, 75 (2007)
18. G. Takács, I. Pilászy, B. Németh, D. Tikk, Investigation of various matrix factorization methods for large recommender systems, in *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition* (ACM, 2008), p. 1
19. G. Takács, I. Pilászy, B. Németh, D. Tikk, A unified approach of factor models and neighbor based methods for large recommender systems, in *Applications of Digital Information and Web Technologies, 2008, ICADIWT 2008, First International Conference* (IEEE, 2008), p. 186
20. Y. Koren, R. Bell, C. Volinsky, Computer **42**, 30 (2009)
21. J. Bao, Y. Zheng, D. Wilkie, M.F. Mokbel, ACM Transaction on Intelligent Systems and Technology (to be published) (2013)
22. P. Symeonidis, D. Ntempos, Y. Manolopoulos, in *Location-based social networks* (Springer, Berlin, 2014), p. 35
23. J.J. Levandoski, M. Sarwat, A. Eldawy, M.F. Mokbel, Lars: A location-aware recommender system, in *28th International Conference on Data Engineering (ICDE)* (IEEE, 2012), p. 450
24. T. Kurashima, T. Iwata, G. Irie, K. Fujimura, Travel route recommendation using geotags in photo sharing sites, in *Proceedings of the 19th ACM international conference on Information and knowledge management* (ACM, 2010), p. 579
25. M. Ye, P. Yin, W.-C. Lee, Location recommendation for location-based social networks, *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (ACM, 2010), p. 458
26. M. Ye, P. Yin, W.-C. Lee, D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (ACM, 2011), p. 325
27. C. Cheng, H. Yang, I. King, M.R. Lyu, in *Proccedings of 26th AAAI Conference on Artificial Intelligence* (2012), pp. 17–23

28. J. Bennett, S. Lanning, The Netflix prize, KDD Cup and Workshop in conjunction with KDD 2007 (2007)
29. P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in *Proceedings of the fourth ACM conference on Recommender systems* (ACM, 2010), p. 39
30. M. Deshpande, G. Karypis, ACM Trans. Inf. Syst. (TOIS) **22**, 143 (2004)
31. Q. Yuan, L. Chen, S. Zhao, Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation, in *Proceedings of the fifth ACM conference on Recommender systems* (ACM, 2011), p. 245
32. P. Bak, C. Tang, K. Wiesenfeld, Phys. Rev. Lett. **59**, 381 (1987)
33. Wikipedia contributors, Abelian sandpile model (2015)
34. Wikipedia contributors, Self-organized criticality (2015)