Towards Dynamic Classification Completeness in Twitter

Dimitris Milioris
Massachusetts Institute of Technology
77 Massachusetts Avenue, MA 02139, USA
Email: milioris@mit.edu

***Abstract*—In this paper we study the application of Matrix Completion in topic detection and classification in Twitter. The proposed method first employs Joint Complexity to perform topic detection based on score matrices. Based on the spatial correlation of tweets and the spatial characteristics of the score matrices, we apply a novel framework which extends the Matrix Completion to build dynamically complete matrices from a small number of random sample Joint Complexity scores. The experimental evaluation with real data from Twitter presents the topic detection accuracy based on complete reconstructed matrices, and thus reducing the exhaustive computation of Joint Complexity scores.**

I. Introduction

During the last decade social networks have changed the way of communication and as a result the information exchanged between users has increased dramatically. In this work we want to address the major challenges of topic detection and classification in Twitter by using less information, and thus reducing the complexity.

The evaluation of the proposed method is based on the detection of topics like the categories of a mainstream news portal. First we perform topic detection based on Joint Complexity (JC) and then we introduce the theory of dynamic Matrix Completion (DynMC) to reduce the computational complexity of JC scores based on the simple Matrix Completion method (MC). The method is context-free and does not use grammar, dictionaries, stemming processes or semantics. Moreover, since it relies directly to the alphabet it is language-agnostic.

The sequence of text is decomposed in linear time into a memory efficient structure called Suffix Tree (ST) [1] and by overlapping two trees, in linear or sublinear average time, we obtain the JC defined as the cardinality of subsequences that are common in both trees. The method has been extensively tested for Markov sources of any order for a finite alphabet and gave good approximation for text generation, language discrimination and author identification. Due to space limitation a more complete prior study of ours can be found in [2].

While there are recent works that propose solutions to the exhaustive computations [3] they are not taking into account the dynamics of the users and use synthetic data instead. We extend these methods by proposing a dynamic framework in Twitter that takes advantage of the spatial correlation of tweets, and thus reduces the computational complexity.

The paper is organized as follows: Section II introduces the JC method, while Section III gives the motivation and describes the application of MC. Section IV describes the framework of DynMC, while Section V evaluates the performance with real data obtained from Twitter. Finally, Section VI summarises our main results and provides directions for future work.

II. Sequence Complexity

Several studies [4], [5] have attempted to define mathematically the *complexity of a sequence* based on the concept of its randomness. Lets assume that X is our sequence and $I(X)$ its set of factors (sub-sequences), then $|I(X)|$ is defined as the complexity of the sequence X. For example if $X = apple$ then $I(X) = \{$a, p, l, e, ap, pp, pl, le, app, ppl, ple, appl, pple, apple, $\nu\}$ and $|I(X)| = 15$ (ν denotes the empty string). Figs. 1 and 2 show the STs for the word *apple* and *maple* respectively.

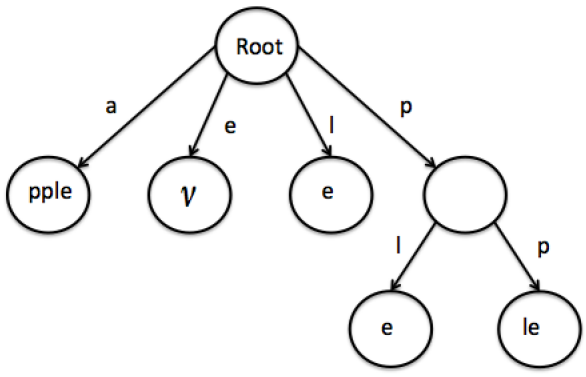

Fig. 1. Suffix Tree for the sequence *apple*, where ν is the empty string.

In order to measure the similarity of two sequences, we have to study the common subsequences between them. In [6] the concept of JC of two sequences was introduced as the number of common distinct factors between them, *i.e.* the JC of sequence X and Y is equal to $J(X,Y) = |I(X) \cap I(Y)|$. Fig. 3 shows the ST superposition for the word *apple* and

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

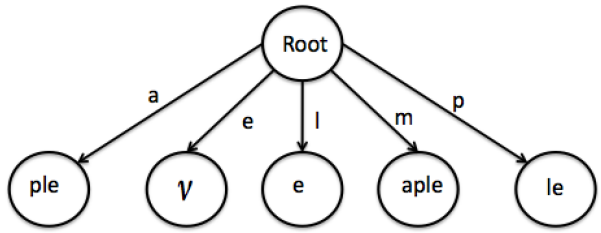

Fig. 2. Suffix Tree for the sequence *maple*, where ν is the empty string.

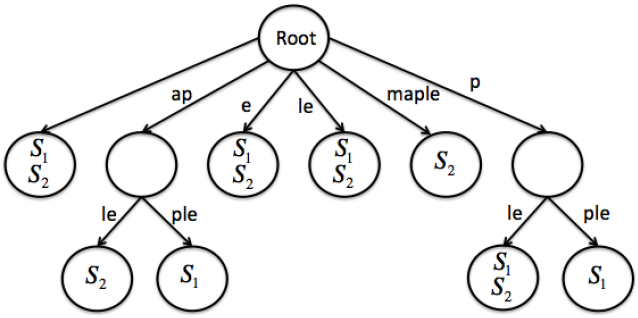

Fig. 3. Suffix Tree superposition for the sequences S_1 = *apple* and S_2 = *maple*, with $J(S_1, S_2) = 9$.

maple respectively, which have nine common factors, *i.e.* $J(X, Y) = 9$.

Our previous experiments [2] showed that the mentioned complexity estimate for memoryless sources converges very slowly. Furthermore, memoryless sources are not appropriate for modelling text generation; we extended JC estimate to Markov sources of any order on a finite alphabet. Markov models are more realistic and have a better approximation for text generation than memoryless sources [2], [7].

Although we do not present deep theoretical analysis in this paper, it worths to mention that the second order asymptotics for JC has the following form:

$$\gamma \frac{m^{\kappa}}{\sqrt{\alpha \log m + \beta}} \tag{1}$$

for some $\beta > 0$, $\kappa < 1$ and $\gamma, \alpha > 0$ which depend on the parameters of the two sources. This new estimate has a faster convergence, and is preferred for texts of order $m \approx 10^2$; For some Markov sources our analysis indicates that JC oscillates with m. This is outlined by the introduction of a periodic function $Q(\log m)$ in the leading term of our asymptotics. This additional term even further improves the convergence for small values of m and therefore JC is an efficient method to capture the similarity degree of short texts. JC has a variety of applications, such as detection of the similarity degree of two sequences, for example the use of plagiarism in texts or documents or the identification of authors and era of texts. In this paper we use it to analyze online social networks, such as Twitter, which has small posts ($\leq$ 140 characters) and perform topic detection.

We have two phases, the *Training*, where we build our databases and the *Runtime*, where we run and test the algorithm.

A. Training Phase

We construct the training databases (DBs) by using Twitter's Streaming API which delivers tweets in the basic *.json* format, while filtering for specific keywords for each category. The main categories we use are politics, economics, technology, lifestyle and sports. For example, we build a class about politics by sending a request to Twitter API for tweets that contain the word "politics". Using these requests we build C classes, with $C = 5$, and each class has N tweets, with $N = 15,000$. We allocate a number of keywords to each class, which are necessary to construct the class.

B. Runtime Phase

Assume that we have a dataset of S timeslots with $s = 1 \dots S$, and each timeslot is a 15 minutes request in Twitter API. For every tweet x_i, where $i = 1 \dots N$, with N being the total number of tweets, in the s-th timeslot, *i.e* x_i^s, we build a suffix tree, $ST(x_i^s)$. Building a ST costs $O(m \log m)$ and takes $O(m)$ space in memory, where m is the length of the tweet, in comparison of m^3 needed by algorithms based on semantic analysis.

Then we compute the JC metric, $JC(x_i^s, x_j^s)$ of the tweet x_i^s with every other tweet x_j^s of the s-th timeslot, where $j = 1 \dots N$, and $j \neq i$ (by convention we choose $JC(x_i^s, x_j^s) = 0$ when $i = j$). The JC between two tweets can be computed efficiently in $O(m)$ operations (sublinear on average) by ST superposition. For the S timeslots we store the JC scores in the matrices $s_1, s_2, \dots, s_S$ of $N \times N$ dimensions.

The representation of timeslots is given by fully connected edge weighted graphs, where each tweet is a node and S_n holds the weight of each edge [8]. The score for each node is calculated by summing weights of all the edges that are connected to that node. The node that gives the highest score is the most representative and central tweet of the timeslot. As mentioned previously, most of the timeslots have $15,000$ tweets, so matrices $s_1, s_2, \dots, s_S$ have approximately 225×10^6 entries for each timeslot. Since the score matrices are symmetric, only half of these entries could be used, *i.e* upper/lower triangular, which reduces the complexity to $\frac{N^2}{2}$.

We then assign a new tweet to the class that maximises the JC metric within that class. In order to limit the size of each reference class we delete the oldest tweets or the least significant ones (e.g. the ones which obtained the lowest JC score).

III. Motivation

The previously described procedure requires the extensive calculation of JC scores, i.e. if there are N_s tweets in the s-th timeslot of the set of tweets to be classified, there are $N_s \times (N_s - 1)/2$ such scores to be calculated.

In order to address the exhaustive computation of JC scores, we perform random sampling on the score matrices. Random sampling reduces the time needed to build the score matrices and as a result the computation needed to check every tweet with every other tweet. In order to succeed on this, we need the existence of correlation between the matrix scores, which depend directly on the sequences (tweets) we have in our database's timeslots.

Tweets have spatial correlation since the ones closer to the meaning show similar measurement score matrices. Since the tweets are correlated, the degrees of freedom of the score matrices are much lower than their dimension. If a matrix has a low rank property, then it presents a limited number of degrees of freedom.

While the recovery of the $i \cdot j$ entries of matrices $s_1, s_2, \ldots, s_S$ is impossible from a number of measurements m (where $m \ll i \cdot j$), MC [9] shows that such a recovery is possible when the rank of a matrix s is small enough compared to its dimensions. In fact, the recovery of the unknown matrix is feasible from $m \geq cj^{6/5} r \log j$ random measurements, where $j > i$ and rank$(s) = r$. The original matrix s can be recovered by solving the following optimization problem:

$$\min ||s||_* \; s.t. \; A(s) = A(M) \tag{2}$$

where $||s||_* = \sum_{k=1}^{\min\{i,j\}} \sigma_k(s)$ with $\sigma_k(s)$ being the k-th largest singular value of s. M is the matrix s after subsampling, while A is a linear map from $R^{i,j} \rightarrow R^m$, that has uniform samples in rows and columns and satisfies the Restricted Isometry Property (RIP).

Sampling on the measurements vectors of s will provide an incomplete scores matrix. The topic that has to be revealed uses a subset Ω of the measurements of S, which randomly chooses by sensing a random number of the $k < h$ tweets of the timeslot. Finally the topic detection get $\Omega \subseteq |i| \times |j|$ measurements, with

$$|\Omega| = \frac{k(i \times j)}{h} \tag{3}$$

while the sampling map $A_\Omega(M)$ has zero entries at the j-th position of the i-th timeslot if $s(i,j) \notin \Omega$.

During the runtime phase we need to recover the unobserved measurements of matrix s, denoted by s^-, by solving the following minimization problem

$$\min ||s^-||_* \; s.t. \; ||A_\Omega(s^-) - A_\Omega(M)||_F^2 < \epsilon \tag{4}$$

where F denotes the Euclidean norm, and ϵ is the noise parameter. The convex optimization problem in (4) can be solved by an interior point solver, *e.g.* CVX [10], or via singular value thresholding, *e.g.* FPC and SVT [11], which applies a singular value decomposition algorithm and then projection on the already known measurements in each step.

IV. Proposed Framework

In this Section, we describe our proposed framework, DynMC, led by the intuition of the spatio-temporal correlations between JC scores among the several representative tweets. During the training phase we collect tweets and compute the JC scores at each time t.

Assume that $C \in R^{i \times i}$ defines the temporal correlation of the tweet in specific classes, while ϵ indicates the noise. The relationship of the tweets between the JC scores and the representative tweets over time can be expressed as:

$$[A(M)]^t = C \; [A(M)]^{t-1} + \epsilon \tag{5}$$

where $[A(M)]^t$ and $[A(M)]^{t-1} \in R^{i \times 1}$ represent the JC scores at time t and $t-1$, respectively, received at a specific class.

As it was mentioned earlier, tweets have a spatial correlation, since closer tweets or classes show similar measurement vectors. In this paper we try to address this problem by introducing a dynamic Matrix Completion technique. The proposed technique is able to recover the unknown matrix at time t by following a random sampling process and reduce the exhaustive computation of JC scores.

As it was mentioned in Section III subsampling gives matrix M_t at each time t of the sampling period and we receive a subset $\Omega_t \subseteq |i| \times |j|$ of the entries of M_t, where $|\Omega|_t = k \times i$. The sampling operator A_t (as defined in Section III) gives

$$[A_t(M)]_{j,i} = \begin{cases} P_{j,i}, (j,i) \in \Omega_t \\ 0, otherwise. \end{cases} \tag{6}$$

where $P_{j,i}$ is the JC score received at j, i and Ω_t is a subset of the complete set of entries $|i| \times |j|$, where $\Omega_t \cup \Omega_t^C = |i| \times |j|$.

While the sampling operator $A_t^C(M_t)$ collects the unobserved measurements at time t, we also define the sampling operator $A_I = A_{t-1} \cap A_t^C$ as the intersection of the training measurements of the classes by time $t-1$.

We need to recover the fingerprint map M_t that will be used during the runtime phase by taking into account the JC scores received on previous time windows. The proposed technique reconstructs matrix M_t that has the minimum nuclear norm, subject to the values of $M_t \in \Omega_t$ and the sampled values at time $t-1$. There is a clear correlation with measurements at time t via C according to the model in Eq. 5. Matrix C

and the original matrix M_t can be recovered by solving the following optimization problem

$$\min_{\tilde{M}_t, C} ||\tilde{M}_t||_* \; s.t. \tag{7}$$

$$||A_t(\tilde{M}_t) - A_t(M_t)||_F^2 \leq \epsilon_1 \tag{8}$$

$$||A_I(C \cdot M_{t-1}) - A_I(M_t)||_F^2 \leq \epsilon_2 \tag{9}$$

where $\tilde{M}_t$ is the recovered JC scores matrix at time t. Variables $\epsilon_1, \epsilon_2 \geq 0$ represent the tolerance in approximation error, while $||\cdot||_F$ denotes the Frobenious norm as mentioned in Section III. Matrix C expresses the relationship between the values of $M_t \in \Omega_t^C \cap \Omega_{t-1}$ and it is adjusted to the number of common tweets at time $t-1$ and t. CVX [10] can be used to solve the general convex optimization problem in Eq. 8 and 9.

V. Experimental Results

Considering the topic detection and classification evaluation, the accuracy of the tested methods was measured with the standard *F-score* metric, using a ground truth over the database of more than 1,5M tweets.

The Document-Pivot (DP) method was selected to compare with our method, since it outperformed the other state-of-the-art techniques in a Twitter context as shown in [12]. The tweets are collected by using specific queries and hashtags and then a bag-of-words is defined, which uses weights with term frequency-inverse document frequency (*tf-idf*). Tweets are ranked and merged by considering similarity context between existing classified and incoming tweets. The similarity is computed by using Locality Sensitive Hashing (LSH) [12], with its main disadvantage being the manual observation of training and test tweets [13].

The classification performance is compared for: (a) Document Pivot (DP), (b) Joint Complexity with Compressive Sensing (JC+CS) [14], [15], (c) Document Pivot with URL (DPurl), (d) Joint Complexity and Compressive Sensing with URL (JCurl+CS) [14], [15], where (c) and (d) include the information of the compressed URL of a tweet concatenated with the original tweet's text; extracted from the *.json* file.

Fig. 4 shows the recovery error of the score matrices s based on FPC algorithm. We can recover the s_i matrices by using approximately 77% of the original symmetric part with the error of *completeness* $\rightarrow 0$, while addressing the problem of exhaustive computations.

Several widely-used norm-based techniques and Bayesian CS algorithms are employed to treat the classification problem.[1]: 1) ℓ_1-norm minimization using the primal-dual interior point method (L1EQ-PD), 2) Orthogonal Matching Pursuit (OMP), 3) Stagewise Orthogonal Matching Pursuit (StOMP), 4) LASSO, 5) BCS [16], and 6) BCS-GSM [17], [18].

[1]For the implementation of methods 1)-5) the MATLAB codes can be found in: http://sparselab.stanford.edu/, http://www.acm.caltech.edu/l1magic, http://people.ee.duke.edu/~lcarin/BCS.html

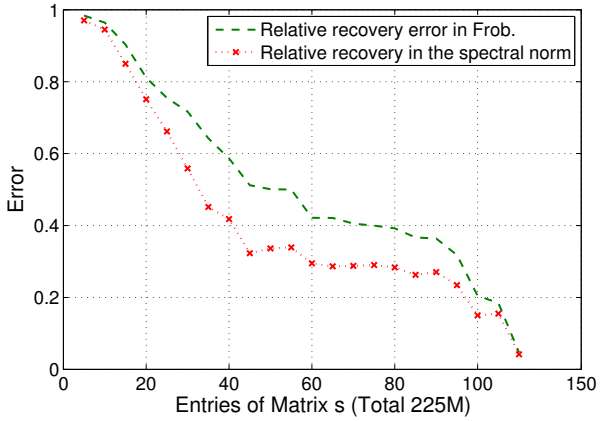

Fig. 4. Reconstruction error of s_i by using the FPC algorithm. Approximately 77% of the symmetric part needed while the error of *completeness* $\rightarrow 0$.

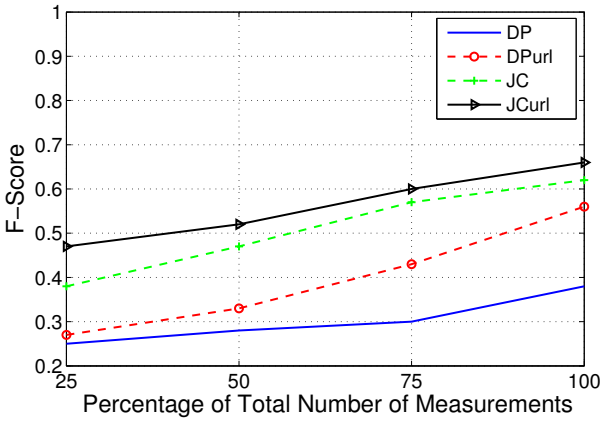

Fig. 5. Topic detection accuracy measured by F-Score for the DP, DPurl and JC, JCurl method as a function of the number of measurements (%) on the recovered matrix s^- by using the ℓ_1-norm min. on 67% of the measurements.

Fig. 5 compares the topic detection accuracy decreased by 10% for the DP and DPurl and 5% for the JC and JCurl method as a function of the number of measurements of the reconstructed matrix s^- (67% of **s**) by using the ℓ_1-norm min. JC and JCurl uses the Bayesian framework which makes them more robust to the noisy process of scores matrices computation while exploring the spatial correlation of the measurements.

Finally, Fig. 6 shows the performance of the proposed DynMC method versus the classic MC as a function of the total number of measurements of the reconstructed symmetric matrix. We can observe a faster convergence of the DynMC method as the number of measurements is increased. More specifically, after the critical point of 40% of the symmetric part, the DynMC performs better and achieves completeness at 78% of the original matrix.

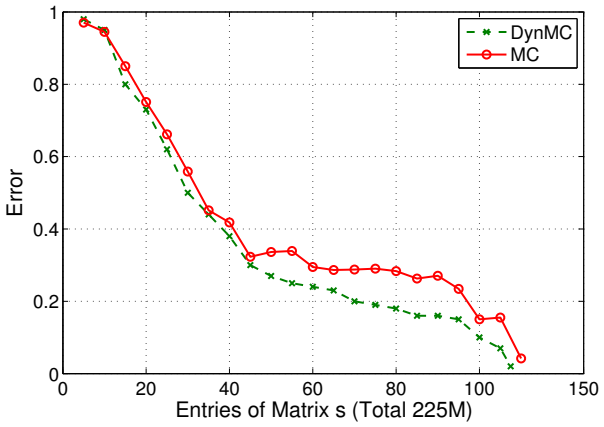

Fig. 6. Reconstruction error of s_i by using the FPC algorithm. DynMC has a faster convergence compared to MC as the error of *completeness* $\rightarrow 0$.

VI. Conclusions

In this paper, we applied a novel framework based on dynamic Matrix Completion to reduce the exhaustive computation of the score matrices during the process of topic detection based on Joint Complexity. The experimental evaluation revealed great performance with low computational complexity. As a future work, we intend to exploit the joint sparsity structure of the score matrices among the several representative tweets, improving the reconstruction accuracy towards their completeness.

Acknowledgment

Dr. Dimitris Milioris would like to thank Bell Labs for sharing the Twitter dataset, and MIT SCL Consortium for supporting this research.

References

[1] S. Tata, R. Hankins and J. Patel, "Practical Suffix Tree Construction", *in Proceedings of the 30th VLDB Conference*, 2004.

[2] P. Jacquet, D. Milioris and W. Szpankowski, "Classification of Markov Sources Through Joint String Complexity: Theory and Experiments", *in IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, July 2013.

[3] S. Nikitaki, G. Tsagkatakis and P. Tsakalides, "Efficient Training for Fingerprint Based Positioning Using Matrix Completion", *in Proc. 20th European Signal Processing Conference (EUSIPCO)*, Boucharest, Romania, August 27–31, 2012.

[4] M. Li and P. Vitanyi, "Introduction to Kolmogorov Complexity and its Application", Springer–Verlag, Berlin, Aug. 1993.

[5] T. Hellseth and H. Niederreiter, "Some computable complexity measures for binary sequences", *in Sequences and Their Applications*, Eds. C. Ding, Springer–Verlag, 67–78, 1999.

[6] P. Jacquet, "Common words between two random strings", *in IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007.

[7] D. Milioris and P. Jacquet, "Joint Sequence Complexity Analysis: Application to Social Networks Information Flow", *in Bell Labs Technical Journal*, Vol. 18, No. 4, 2014 (doi: 10.1002/bltj.21647).

[8] G. Burnside, D. Milioris and P. Jacquet. "One Day in Twitter: Topic Detection Via Joint Complexity", *Proceedings of SNOW 2014 Data Challenge, WWW'14, Seoul, South Korea*, April 2014.

[9] E. Candès and B. Recht, "Exact matrix completion via convex optimization", in *Foundations of Computational Mathematics*, vol. 9, pp. 717–772, 2009.

[10] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming', Version 1.21", 2011.

[11] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion", in *SIAM Journal on Optimization*, vol. 20, pp. 1956–1982, 2010.

[12] L. M. Aiello *et al.*, "Sensing Trending Topics in Twitter", *in IEEE Transactions on Multimedia*, Vol. 15, Iss. 6, pp. 1268–1282, Oct 2013.

[13] H. Becker et. al., "Beyond Trending Topics: Real-World Event Identification on Twitter", *in 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[14] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *in IEEE Transactions on Information Theory*, Vol. 52, pp. 489–509, Feb. 2006.

[15] D. Milioris and P. Jacquet, "Topic Detection and Compressed Classification in Twitter", *in IEEE European Signal Processing Conference (EUSIPCO'15)*, Nice, France, Sept. 2015.

[16] S. Ji, Y. Xue, and L. Carin, Bayesian compressive sensing, in *IEEE Transactions on Signal Processing*, 2008.

[17] G. Tzagkarakis and P. Tsakalides, "Bayesian compressed sensing imaging using a Gaussian scale mixture", in *Proc. 35th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010.

[18] G. Tzagkarakis et al., "Multiple-Measurement Bayesian Compressive Sensing using GSM Priors for DOA Estimation", *in Proc. 35th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010.