Classification Encryption via Compressed Permuted Measurement Matrices

Dimitrios Milioris

Classification Encryption via Compressed Permuted Measurement Matrices

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

***Abstract*—In this paper we present an efficient encryption system based on Compressive Sensing for topic detection and classification in Twitter. The proposed method first employs Joint Complexity to perform topic detection. Then based on the spatial nature of the data, we apply the theory of Compressive Sensing to perform classification from a small number of random sample measurements. The breakthrough of the method is the encryption based on the permutation of measurements which are generated when solving the classification optimization problem. The experimental evaluation with real data from Twitter presents the robustness of the encryption accuracy, without using a specific cryptographic layer, while maintaining a low computational complexity.**

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 order to extract the relevant information from social networks we have to (a) analyze the relationship between its members and reveal important teams, (b) search for quality and timeliness of the content besides the relevance, (c) avoid the growth of irrelevant information by detecting advertisements and spam messages, and (d) address these challenges in a secure way.

The evaluation of the proposed method is based on the detection of topics like the categories of a mainstream news portal. First we propose a method which is based on Joint Complexity (JC) to perform topic detection and then we apply the theory of Compressive Sensing (CS) to perform optimized classification, which generates the intuition of the weak encryption properties of CS. 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 [2].

The paper is organized as follows: Section II introduces the JC method for topic detection, while Section III describes the application of CS on classification. Sections IV and V describe the encryption architecture based on the already weakly encrypted measurements, while Section VI evaluates the performance with real data obtained from Twitter. Finally, Section VII summarises our main results and provides directions for future work.

II. Sequence Complexity

Several studies [3], [4] 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.

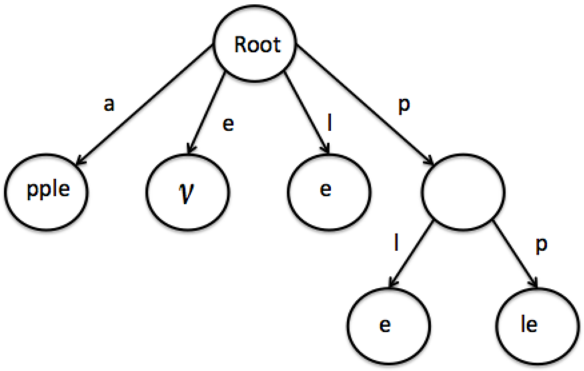


Fig. 1. Suffix Tree for the sequence *apple*, where ν is the empty string.

In order to measure the similarity of two sequences, we have to study the common subsequences between them. In [5] 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 *maple* respectively, which have nine common factors, *i.e.* $J(X,Y) = 9$.

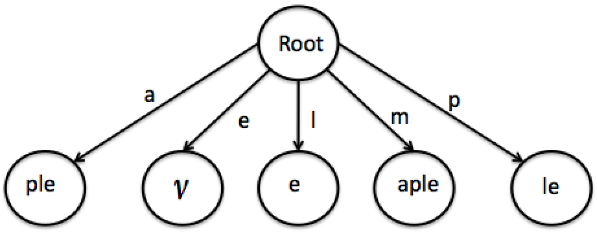


Fig. 2. Suffix Tree for the sequence *maple*, where ν is the empty string.

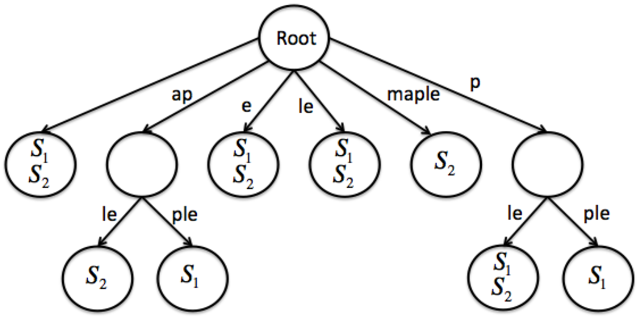


Fig. 3. Suffix Tree superposition for the sequences S_1 = *apple* and S_2 = *maple*, with $J(S_1, S_2) = 9$.

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. In a prior work, we extend 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], [6].

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 perform topic detection and classification in Twitter, where posts are ≤ 140 characters.

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 [7]. 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 maximizes 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. Compressive Sensing

First, we describe the theoretical concepts of the CS application [8], [9] in the problem of classification. Let $\mathbf{x} \in \mathbb{R}^N$ denote the signal of interest. Such signal can be represented as a linear combination of a set of basis $\{\psi_i\}_{i=1}^{N}$. By constructing a $N \times N$ basis matrix $\boldsymbol{\Psi} = [\psi_1, \psi_2, \dots, \psi_N]$, the signal x can be expressed as $x = \sum_{i=1}^{N} s_i \psi_i = \boldsymbol{\Psi} s$. In fact the signal is represented as $x = \boldsymbol{\Psi} s + \theta$, with $\theta \in \mathbb{R}^N$ being the noise, where $\mathbb{E}(\theta) = 0$ and $var(\theta) = O(|\Psi s|)$.

The efficiency of a CS method for signal approximation or reconstruction depends highly on the sparsity structure of the signal in a suitable transform domain associated with an appropriate sparsifying basis $\boldsymbol{\Psi}$. The measurement model in the original space-domain is expressed as $\mathbf{g} = \boldsymbol{\Phi}\mathbf{x}$, where $\mathbf{g} \in \mathbb{R}^M$ is the measurement vector and $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ denotes the measurement matrix. The measurement model has the following equivalent transform-domain representation

$$\mathbf{g} = \boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{s} + \boldsymbol{\Phi}\boldsymbol{\theta} \tag{2}$$

In fact when the length of the sequence $m \to \infty$ and $N \to \infty$, $\mathbb{E}(\boldsymbol{\Psi}\boldsymbol{s}) = O(mN)$, with $var(\theta) = O(mN)$, $std(\theta) = O(\sqrt{|\Phi|m})$ and $\mathbb{E}(\boldsymbol{\Phi}\boldsymbol{\theta}) = 0$. The second part of (2), $\boldsymbol{\Phi}\boldsymbol{\theta}$ is of relative order $O(\frac{1}{\sqrt{mN}})$, and is negligible compare to $\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{s}$ due to the law of large numbers.

Examples of measurement matrices $\boldsymbol{\Phi}$, which are incoherent with any fixed transform basis $\boldsymbol{\Psi}$ with high probability (universality property [10]), are random matrices with independent and identically distributed (i.i.d.) Gaussian or Bernoulli entries.

The problem of classifying a tweet is reduced to a problem of recovering the one-sparse vector $\mathbf{s}$. Of course in practice we do not expect an exact sparsity, thus, the estimated class corresponds simply to the largest-amplitude component of $\mathbf{s}$. According to [10], [11], $\mathbf{s}$ can be recovered perfectly with high probability by solving the following optimization problem

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \Big(\|\mathbf{s}\|_1 + \tau \|\mathbf{g} - (\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{s})\|_2 \Big) \tag{3}$$

where τ is a regularization factor that controls the trade-off between the achieved sparsity and the reconstruction error.

A. Preprocessing phase

During the Preprocessing phase, we built our classes as described in Section II and for each class we extract the most representative tweet(s) (CTs) based on the Joint Complexity method. The vector $\boldsymbol{\Psi}_T^i$ consists of the highest JC scores of the i-th CT. The matrix $\boldsymbol{\Psi}_T$ is used as the appropriate sparsifying dictionary for the training phase. Moreover, a measurement matrix $\boldsymbol{\Phi}_T^i$ is associated with each transform matrix $\boldsymbol{\Psi}_T^i$, while T denotes the preprocessing phase.

The matrix $\boldsymbol{\Psi}_T^i \in \mathbb{R}^{N_i \times C}$ is used as the appropriate sparsifying dictionary for the i-th CT, since in the ideal case the vector of tweets at a given class j received from CT i should be closer to the corresponding vectors of its neighboring classes, and thus it could be expressed as a linear combination of a small subset of the columns of $\boldsymbol{\Psi}_T^i$. Moreover, a measurement matrix $\boldsymbol{\Phi}_T^i \in \mathbb{R}^{M_i \times N_i}$ is associated with each transform matrix $\boldsymbol{\Psi}_T^i$, where M_i is the number of CS measurements. In the proposed algorithm, a standard Gaussian measurement matrix is employed, with its columns being normalized to unit ℓ_2 norm. A random matrix or a PCA matrix could be also used.

B. Run phase

A similar process is followed during the runtime phase. More specifically, we denote $\mathbf{x}_{c,R}$ as the Joint Complexity score of the incoming tweet with the CT_i classified at the current class c, where R denotes the runtime phase. The runtime CS measurement model is written as

$$\mathbf{g}_c = \boldsymbol{\Phi}_R \mathbf{x}_{c,R} \tag{4}$$

where $\boldsymbol{\Phi}_R^i \in \mathbb{R}^{M_{c,i} \times N_{c,i}}$ denotes the corresponding measurement matrix during the runtime phase. In order to overcome the problem of the difference in dimensionality between the preprocessing and run phase, while maintaining the robustness of the reconstruction procedure, we select $\boldsymbol{\Phi}_R^i$ to be a subset of $\boldsymbol{\Phi}_T^i$ with an appropriate number of rows such as to maintain equal measurement ratios.

The measurement vector $\mathbf{g}_{c,i}$ is formed for each CT i according to (4) and transmitted to the server, where the reconstruction takes place via the solution of (3), with the training matrix $\boldsymbol{\Psi}_T^i$ being used as the appropriate sparsifying dictionary. We emphasize at this point the significant conservation of the processing and bandwidth resources of the wireless device by computing only low-dimensional matrix-vector products to form $\mathbf{g}_{c,i}$ $(i = 1, \dots, P)$ and then transmitting a highly reduced amount of data ($M_{c,i} \ll N_{c,i}$). Then, the CS reconstruction can be performed remotely (*e.g.*, at a server) for each CT independently.

Last, we would like to note the assumption that the CS-based classification method involves the mobile device that collects the tweets from the Twitter API and a server that performs the core CS algorithm.

IV. Security System Architecture

The method consist of two parts: (IV-A) Privacy system, and (IV-B) Key description.

A. Privacy system

Due to their acquisition process, CS measurements can be viewed as *weakly encrypted* for a malicious user without knowledge of the random matrices Φ^i, which have independent and identically distributed (i.i.d.) Gaussian or Bernoulli entries [10].

The encryption property of a CS approach relies on the fact that the matrix Φ is unknown to an unauthorized entity, since Φ can be generated using a (time-varying) cryptographic key that only the device and the server share.

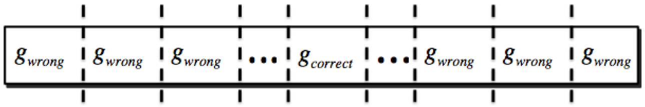
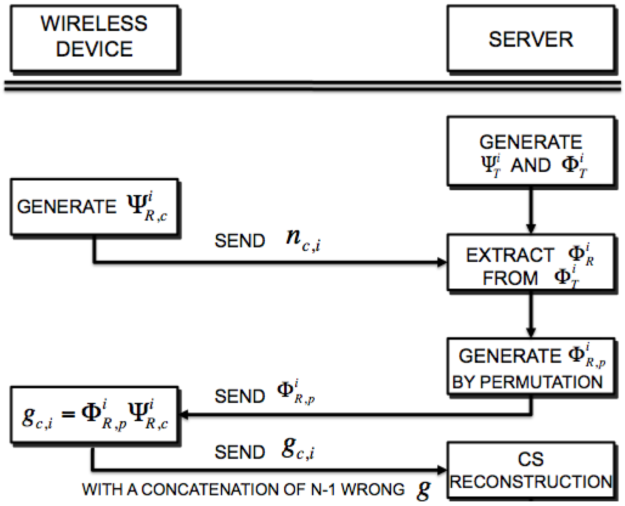


Fig. 4. N-1 false vectors plus the correct one. This *key*, i.e., the sequence of the measurement vectors reaches the server.



WIRELESS DEVICE
SERVER
GENERATE Ψ^i_T AND Φ^i_T
GENERATE $\Psi^i_{R,c}$
SEND $n_{c,i}$
EXTRACT Φ^i_R FROM Φ^i_T
GENERATE $\Phi^i_{R,p}$ BY PERMUTATION
SEND $\Phi^i_{R,p}$
$g_{c,i} = \Phi^i_{R,p}\Psi^i_{R,c}$
SEND $g_{c,i}$
WITH A CONCATENATION OF N-1 WRONG g
CS RECONSTRUCTION

Fig. 5. System's security architecture

More specifically, the server extracts the runtime sub-matrix Φ^i_R from the training Φ^i_T. The lines of Φ^i_R are permuted and the *key* of the merge of the false measurement vectors and the correct one is used, as shown in Fig. 4 and Fig. 5 and described extensively in [8].

B. Key description

The device sends the measurement vector g to the server along with $N-1$ false vectors, where the reconstruction takes place. Then, the server uses the information of the topic detection and the representative tweets based on their JC scores, etc., and performs classification as described in Section III.

V. Possible Attacks from Malicious Users

A possible attack could follow two directions:

1. Find Φ matrix by intercepting the server. Modern network cryptographic protocols could guarantee that the decryption of Φ_p – where p denotes the permutation of the lines – is almost infeasible in practice due to the combinatorial nature of the inverse problem.

2. Find g by intercepting the opposite direction. The exact knowledge of g is insufficient, resulting in a significantly increased estimation error, when the attacker does not achieve the exact estimate of Φ^i_R. Finding the correct measurement vector g, which increases the estimation error (without an exact knowledge).

VI. 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].

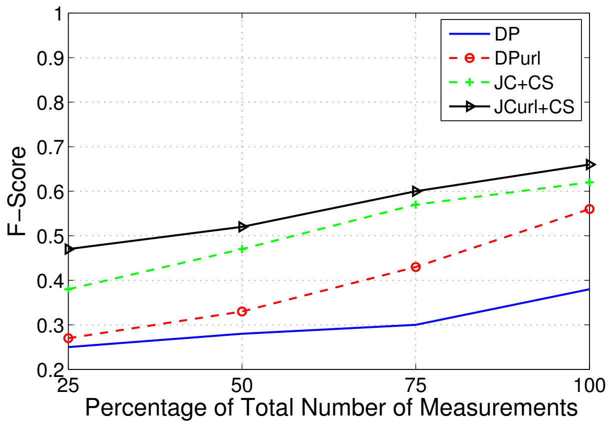


Fig. 6. Classification accuracy measured by F-Score for the DP, DPurl and JC+CS, JCurl+CS method as a function of the number of measurements (%) by using the ℓ_1-norm min.

The classification performance is compared for: (a) Document Pivot (DP), (b) Joint Complexity with Compressive Sensing (JC+CS), (c) Document Pivot with URL (DPurl), (d) Joint Complexity and Compressive Sensing with URL (JCurl+CS), 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. 6 compares the classification accuracy (increased by 37%) of the DP, DPurl and JC+CS, JCurl+CS method as a function of the number of measurements by using the ℓ_1-norm minimization. Fig. 7 compares the reconstruction performance between several widely-used norm-based techniques and Bayesian CS algorithms. More specifically, the following methods are employed[1]: 1) ℓ_1-norm minimization using the primal-dual interior point method (L1EQ-PD), 2) Orthogonal

[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

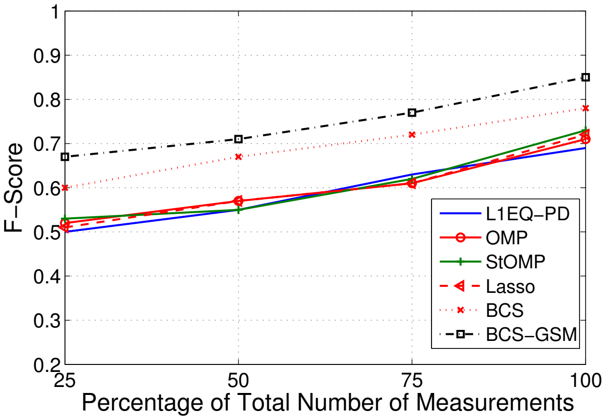


Fig. 7. Classification accuracy measured by F-Score as a function of the number of measurements (%) by using several reconstruction techniques, for the JCurl+CS method.

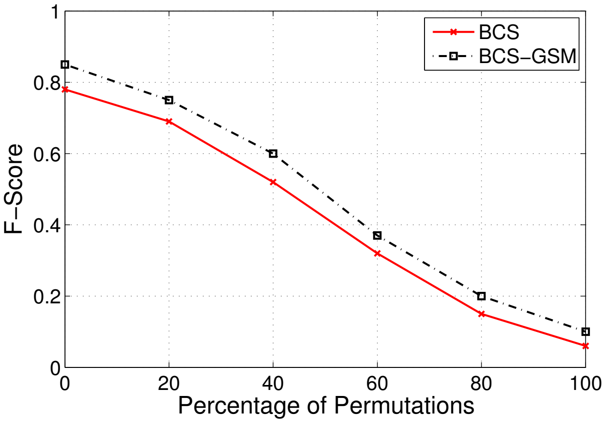


Fig. 8. Evaluation of the encryption property using BCS and BCS-GSM, for a varying number of permuted lines of Φ_R^i.

Matching Pursuit (OMP), 3) Stagewise Orthogonal Matching Pursuit (StOMP), 4) LASSO, 5) BCS [14], and 6) BCS-GSM [15], [16].

Fig. 8 shows the encryption capability of the method for the Bayesian Compressive Sensing (BCS) [14] and Bayesian Compressive Sensing Gaussian Scale Mixture (BCS-GSM) [15], [16] reconstruction algorithms, which outperformed in the classification accuracy.

The average *F-score* is shown as a function of the percentage of permuted lines from 0% to 100%, of the true matrices Φ_R^i, where the reconstruction is performed by considering exact knowledge of the measurement vectors g. The results agree with the intuition that as the complexity of the permutation increases, the *F-score* classification accuracy decreases (approx. 78%) without an exact estimate of the true measurement matrix.

VII. Conclusions

In this paper, we study the encryption property of Compressive Sensing in order to secure the classification process in Twitter without an extra cryptographic layer. First we performed topic detection based on Joint Complexity and then we employed the theory of Compressive Sensing to classify the tweets by taking advantage of the spatial nature of the problem. The measurements obtained are considered to be weakly encrypted due to their acquisition process, which was verified by the experimental results. As a future work, we intend to exploit the robustness of the weakly encryption properties of the sparsity structure among the several representative tweets.

Acknowledgment

Dr. Dimitris Milioris would like to thank Bell Labs for sharing the Twitter dataset, and the MIT Senseable City Lab 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] M. Li and P. Vitanyi, “Introduction to Kolmogorov Complexity and its Application”, Springer–Verlag, Berlin, Aug. 1993.

[4] 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.

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

[6] 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).

[7] 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.

[8] D. Milioris *et al.*, “Low-dimensional signal-strength fingerprint-based positioning in wireless LANs”, *in Ad Hoc Networks Journal, Elsevier*, Vol. 12, pp. 100–114, Jan. 2014.

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

[10] D. Donoho, “Compressive sensing”, *in IEEE Transactions on Information Theory*, Vol. 52, No. 4, pp. 1289–1306, April 2006.

[11] E. Candés, 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.

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

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

[16] 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.