SPARSEHASH: EMBEDDING JACCARD COEFFICIENT BETWEEN SUPPORTS OF SIGNALS

D. Valsesia, S. M. Fosson, C. Ravazzi, T. Bianchi, E. Magli

Politecnico di Torino - DET, Italy
{name.surname}@polito.it

ABSTRACT

Embeddings provide compact representations of signals to be used to perform inference in a wide variety of tasks. Random projections have been extensively used to preserve Euclidean distances or inner products of high dimensional signals into low dimensional representations. Different techniques based on hashing have been used in the past to embed set similarity metrics such as the Jaccard coefficient. In this paper we show that a class of random projections based on sparse matrices can be used to preserve the Jaccard coefficient between the supports of sparse signals. Our proposed construction can be therefore used in a variety of tasks in machine learning and multimedia signal processing where the overlap between signal supports is a relevant similarity metric. We also present an application in retrieval of similar text documents where SparseHash improves over MinHash.

Index Terms— Embedding, Jaccard coefficient, random projections, sparse matrices, MinHash

1. INTRODUCTION

Recent trends in signal processing are increasingly pushing researchers to investigate compact signal representations that build on signal sparsity. Such compact representations can be naturally used for signal acquisition and recovery, as it has been extensively studied in the compressed sensing literature [1–3]. However, similar representations are also very useful when one is not interested in signal recovery, but only in performing some signal classification tasks that are based on signal properties preserved by the compact representation.

In this second case, these representations are usually referred to as embeddings. Formally, an embedding is a transformation that maps a set of signals in a high dimensional space to a lower dimensional space, in such a way that the geometry of the set is approximately preserved. An important class of signal embeddings are those preserving the distances among pair of signals. The most famous embedding is probably the one proposed by Johnson and Lindenstrauss [4], which preserves Euclidean distances using random projections. Several extensions have been later proposed, allowing one to embed the angle between signals [5, 6], or controlling the maximum distance that is embedded [7].

The concept of embedding has been successfully used also in the more general context of information retrieval [8], where it is usually called "hashing". For example, it is a fundamental ingredient of efficient indexing techniques known as locality sensitive hashing [9]. In several information retrieval problems "bag-of-features" representations [10, 11] are used to describe complex objects (e.g. images or text documents) by counting if and how many times a particular feature from a dictionary is present in the objects. In such problems, like the search of near-duplicate documents, or similar images, the usual metric is not the Euclidean distance but a similarity index between sets, where the elements in the sets are the vocabulary elements present in the objects under examination. One of the most used techniques for measuring set similarity is min-wise hashing (also known as MinHash) [12–14], which approximately preserves the Jaccard similarity coefficient between pairs of sets and is used in a wide range of applications [15].

In this paper, we introduce an alternative embedding for the Jaccard coefficient which is based on the concepts of random projections and signal sparsity. The proposed embedding builds on recent results showing that measurements acquired using a sparse random matrix can be used to estimate the number of nonzero components, i.e., the size of the support, of the acquired signal [16, 17]. Based on this result, we show that, given a pair of signals, their measurements can be efficiently used for estimating the size of both the union and the intersection of the signal supports. Hence, random projections obtained from a sparse random matrix provide an embedding of the Jaccard coefficient of the signal supports. Moreover, since these projections can be quantized using a single bit, they represent an efficient alternative to widely used MinHash.

This paper is organized as follows. In Sec.2 we provide some background on random projections and hashing techniques for set similarity, namely MinHash. Sec.3 discusses the proposed method to use random projections as an embedding of the Jaccard coefficient between signal supports, also

This work is supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement n.279848.

providing some theoretical results. Sec.4 validates the proposed technique with synthetic and real datasets. Finally, we draw some conclusions and explore future lines of work in Sec.5.

2. BACKGROUND

In this section we provide some background material on known embeddings of common distance measures such as Euclidean distance, angular distance and Jaccard distance for set similarity.

2.1. Similarity search and embeddings

Let $\mathcal{X} = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^n$ be a collection of data points and $d_{\mathcal{X}}$ be a metric defined on $\mathcal{X}$. Given a query item ξ, the problem of proximity search is to find the items $\mathcal{Q}$ that are within the distance τ from ξ: $\mathcal{Q} = \{x \in \mathcal{X} : d_{\mathcal{X}}(x, \xi) \leq \tau\}$. It should be noticed that the computation generally requires $O(Nn)$ operations, which can be prohibitive for large N and n. An embedding is a function $f : \mathcal{X} \mapsto \mathcal{Y} \subseteq \mathbb{R}^m$, which maps vectors in the high dimensional space into a lower dimensional one ($m \ll n$) equipped with the distance metric $d_{\mathcal{Y}}$, preserving the geometry of the set with a low distortion. Then, distances can be computed in the low dimensional embeddings, rather than the original space, implying a cost reduction in the computation from $O(Nn)$ to $O(Nm)$ operations. In the following paragraphs we review two popular approaches of transforming the data to a low dimensional representation: random projections and MinHash.

2.2. Random projections

Random projections have been used as embeddings in order to reduce the dimensionality of points. Johnson-Lindenstrauss lemma [4] states that random linear mappings $f(x) = Ax$ with $A \in \mathbb{R}^{m \times n}$, if properly designed, preserve the Euclidean distances of points within a small tolerance with high probability. More precisely, given $\epsilon \in (0, 1)$, $\beta > 0$, and $N, m \in \mathbb{N}$ such that

$$m \geq \frac{24 + 12\beta}{3\epsilon^2 - 2\epsilon^3} \log N,$$

then there exists a distribution over $\mathbb{R}^{m \times n}$ from which the matrix A is drawn such that for all $u, v \in \mathcal{X}$

$$(1 - \epsilon)\|u - v\|_2^2 \leq \|Au - Av\|_2^2 \leq (1 + \epsilon)\|u - v\|_2^2$$

with probability exceeding $1 - N^{-\beta}$ (the interested reader can refer to [18] for the proof).

The most common choice for the distribution of the entries of the matrix A is i.i.d. Gaussian $\mathcal{N}(0, 1/m)$. In [8] other distributions are proposed in order to speed up the computation using sparse random projections of the form

$$A_{ij} = \begin{cases} \sqrt{\frac{s}{m}} & \text{w.p. } \frac{1}{2s} \\ 0 & \text{w.p. } 1 - \frac{1}{s} \\ -\sqrt{\frac{s}{m}} & \text{w.p. } \frac{1}{2s} \end{cases}$$

where only $1/s$ of the data need to be processed. In [19] it is shown that, under suitable conditions, one can use $s = n/\log(n)$ to significantly speed up the computation.

Finally, another popular embedding is constituted by Sign Random Projections [5] for angle-based distance formed by any two vectors $u, v \in \mathcal{X}$

$$\theta(u, v) = \frac{1}{\pi} \arccos\left(\frac{u^\top v}{\|u\|_2 \|v\|_2}\right).$$

The hash function is formulated as $f(x) = \text{sign}(Au)$, where $A \in \mathbb{R}^{m \times n}$ with i.i.d. Gaussian entries. It can be shown that $\mathbb{P}(f(u) \neq f(v)) = \theta(u, v)$. Then the vectors can be compared in the reduced space using Hamming distances for which efficient algorithms are available in the literature [20]. Compared to regular random projections, for each data point, Sign Random Projections need to store just one bit per projection.

In this paper, we show that sparse random matrices, if properly designed, can embed the Jaccard coefficient between the supports of sparse signals.

2.3. MinHash

The Jaccard coefficient is a similarity measure between two sets $S_1, S_2 \subseteq \Omega = \{1, \ldots, n\}$ and is defined as

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}.$$

The related distance is $1 - J(S_1, S_2)$.

The most popular technique to estimate the Jaccard coefficient is represented by MinHash, which works as follows. Let $S \subseteq \Omega = \{1, \ldots, n\}$, π be a uniformly chosen permutation on Ω, then the hash function $h : \Omega \to \Omega$ is

$$h(S) = \min_{a \in S} \pi(a).$$

It can be easily shown that

$$\mathbb{P}[(h(S_1) = h(S_2)] = J(S_1, S_2).$$

Then, given m hash values of two sets (all permutations are generated independently), the Jaccard coefficient is estimated as

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(\{h_i(S_1) = h_i(S_2)\})$$

where $\mathbb{1}$ be the indicator function. In [15] the authors propose to use only the least significant b-bits of the MinHash value,

instead of using 64 bits or 40 bits as in [21] and [12], respectively. The most common solution adopted in practice is to keep a single bit, thus estimating the Jaccard coefficient from the Hamming distance between hash vectors as

$$1 - \frac{2}{m} \sum_{i=1}^{m} \mathbb{1}\left(\{[h_i(S_1) \bmod 2] \neq [h_i(S_2) \bmod 2]\}\right)$$

3. PROPOSED METHOD

We propose sparse random projections as a tool to estimate the Jaccard coefficient between supports of signals in high dimensional space. We define the support of a signal $u \in \mathbb{R}^n$ as the set of nonzero elements of u:

$$\text{supp}(u) = \{i \in \{1, \dots, n\} : u_i \neq 0\}.$$

Given $u, v \in \mathbb{R}^n$ we are interested in estimating the Jaccard coefficient $J(S_u, S_v)$ of the two sets $S_u = \text{supp}(u)$ and $S_v = \text{supp}(v)$. To simplify the notation from now on we denote $J(S_u, S_v)$ with $J(u, v)$.

The hash function we consider is $f(x) = \mathbb{1}(\{Ax = 0\})$ where $A \in \mathbb{R}^{m \times n}$ with $m < n$ is a γ-sparsified matrix, whose entries are i.i.d. according to

$$A_{ij} \sim \begin{cases} \mathcal{N}(0, \frac{1}{\gamma}) & \text{w.p. } \gamma, \\ \delta_0 & \text{w.p. } 1 - \gamma \end{cases} \tag{1}$$

where δ_0 denotes a Dirac delta centered at zero. Also in this case, as in Sign Random Projections, each data point needs to store just one bit per projection.

Let now $y, z \in \mathbb{R}^m$ and define

$$I_1(y, z) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(\{y_i = 0, z_i = 0\}),$$
$$I_2(y, z) = \frac{\sum_{i=1}^{m} \mathbb{1}(\{y_i = 0\}) \sum_{j=1}^{m} \mathbb{1}(\{z_j = 0\})}{m \sum_{i=1}^{m} \mathbb{1}(\{y_i = 0, z_i = 0\})},$$

then the signals can be compared in the reduced space using the following similarity index:

$$I(y, z) = \frac{\log(I_2(y, z))}{\log(I_1(y, z))}. \tag{2}$$

In the following proposition, we state that the proposed similarity index concentrates around the Jaccard coefficient between the supports of the original signals. Due to space contraints, we delay the proof to a future article.

Proposition 1. *Let u, v be a pair of arbitrary vectors. Fix $\epsilon > 0$, then there exists $q = q(\epsilon) \in (0, 1)$ such that*

$$\mathbb{P}[|I(Au, Av) - J(u, v)| > \epsilon] \leq q^m. \tag{3}$$

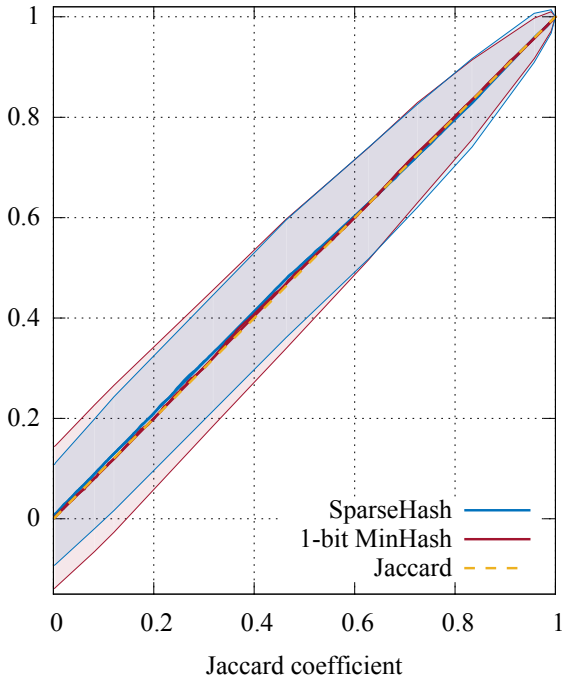

Fig. 1. Jaccard coefficient estimation with $m = 50$.

In this work, the choice of distribution $\mathcal{N}(0, \frac{1}{\gamma})$ for the nonzero entries of A is arbitrary and can be replaced with any continuous distribution with zero mean and variance $1/\gamma$. However, different choices of distribution can change the error tail bound.

4. EXPERIMENTAL RESULTS

In this section we perform some experiments with the proposed embedding technique. First, we experimentally show that the embedding metric we introduced in Eq. (2) indeed concentrates around the Jaccard coefficient as stated in Eq. (3). Then, we address a classification problem using a real dataset of text data.

4.1. Embedding validation

We perform some experiments to validate the theoretical result that the similarity metric $I(Au, Av)$ between the random projections of two signals of interest concentrates around the Jaccard coefficient between their original supports. In order to show this result, we generate a large number of signals at random with varying amount of support overlap and computed their random projections. The signals used in this experiment have $n = 1000$ and the cardinality of the support is $k = 230$. The dimension of the reduced space has been fixed to $m = 50$. The mean and variance of the similarity index I are evaluated over 500 iterations. The γ parameter controlling the sparsity of the sensing matrix is set as the value that max-

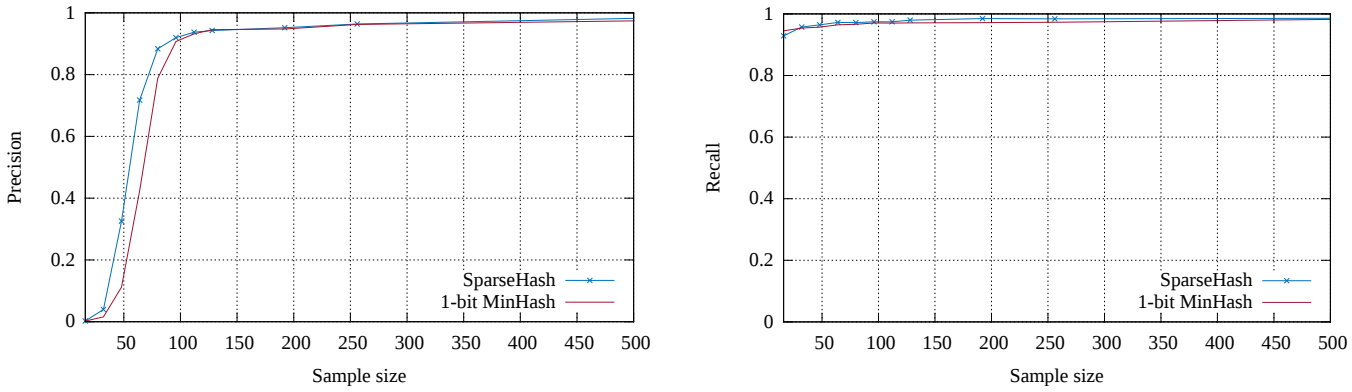

Fig. 2. Precision and recall, threshold 0.5.

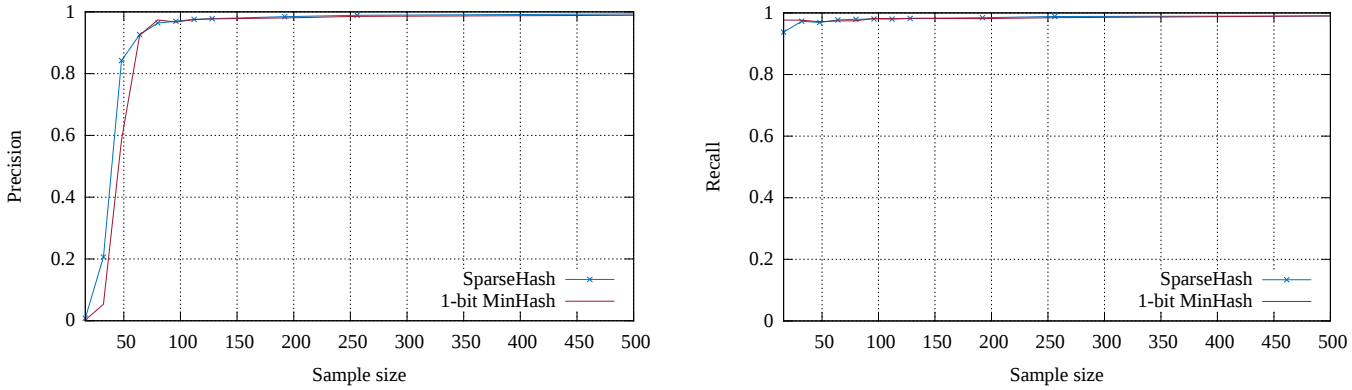

Fig. 3. Precision and recall, threshold 0.6.

imizes the entropy of the binary measurements, i.e. generates zero or nonzero measurements with equal probability. Since

$$\mathbb{P}(f_i(u) = 0) = (1 - \gamma)^k,$$

then we set

$$\gamma = 1 - 2^{-\frac{1}{k}} \approx 3 \cdot 10^{-3}.$$

Fig. 1 shows that the mean value of $I(Au, Av)$ (solid blue curve) computed between every pair of random projections is close to $J(u, v)$ (dashed yellow line). The shaded area represents an interval of width equal to one standard deviation above and below the mean value.

Since the proposed SparseHash method only requires to store 1 bit per measurement, we compared it to the binary version of MinHash applied to the same signals. It can be noticed that SparseHash and MinHash show a similar behaviour with a slight reduction in variance for SparseHash.

4.2. Classification with real dataset

The goal of this section is to test the performance of SparseHash on a classification experiment with a real dataset. Finding near-duplicate or similar documents in an archive of text data has been an important problem for a long time and several works [12,21,22] from the early days of the Web to more recent times have addressed the issue. Documents can be represented with bag of words or bag of shingles (groups of consecutive words) models, where what we called "signal" is the count of how many times a particular word or shingle appears. Such models are typically very sparse signals because the number of different words/shingles appearing in a particular document is typically small compared to the size of the vocabulary. Since our goal is to ascertain the quality of the embedding provided by SparseHash, we perform an experiment similar to the one reported in [15], where the authors compared how various quantization rates affected the performance of MinHash. We use the standard and publicly available UCI dataset of New York Times articles [23]. This dataset is composed of about 300000 news articles, with a bag of words representation given for each article. In terms of the signal parameters that we used in this paper, $n = 102660$. The mean sparsity (i.e. the number of different words used in each article) is $k = 232$. As in Sec.4.1, γ is set as the value that maximizes the entropy. Since the sparsity degree varies, we approximate it with the mean value k, so that $\gamma = 1 - 2^{-\frac{1}{k}}$.

We compare the performance of SparseHash and 1-bit

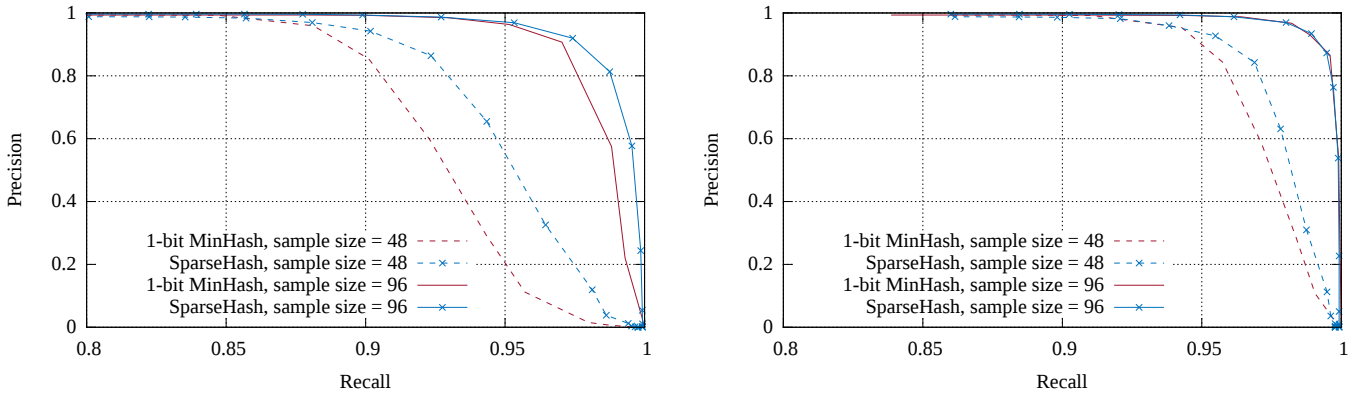

Fig. 4. Precision vs recall, with ground truth threshold 0.5 (left) and 0.6 (right)

MinHash, in terms of precision and recall. Specifically we define as similar the documents with Jaccard coefficient larger than a certain threshold, and we try to retrieve them. If Q is the set of similar documents and $\widehat{Q}$ its estimate, the precision is defined as $\frac{|Q \cap \widehat{Q}|}{|\widehat{Q}|}$, while the recall is $\frac{|Q \cap \widehat{Q}|}{|Q|}$. In figures 2 and 3, we set the thresholds 0.5 and 0.6, respectively, and we show precision and recall as function of the sample size. Concerning the precision, we notice that SparseHash outperforms 1-bit MinHash, in particular for sample sizes smaller than 100. For larger small sizes, both methods are efficient, with precision very close to 1. On the other hand, the recall is very close to one for both methods, for all the tested sample sizes.

In Figure 4, we depict the behavior of the precision as a function of the recall. In this experiment, the goal is to recover all the documents with Jaccard coefficient larger than 0.5 (left graph) and 0.6 (right graph). We consider sample sizes in $\{48, 96\}$. In all these settings, SparseHash outperforms 1-bit MinHash, i.e., at same recall the precision of SparseHash is higher. The gain is more evident for smaller sample size.

The best performance of SparseHash with respect to 1-bit MinHash is consistent with the results in Figure 1, in which we noticed a smaller variance for SparseHash in a numerical setting close to that of the dataset here considered.

5. CONCLUSIONS AND FUTURE WORK

This paper introduced SparseHash, an embedding technique that reduces the dimensionality of signals while preserving the Jaccard coefficient between their supports. Contrary to other techniques present in literature for embedding set similarity, such as MinHash, we derived the method starting from the literature on random projections and compressed sensing. We showed that the method is an interesting alternative to binary MinHash, improving over it by providing lower variance for the same number of bits required by the hash. Moreover, we also tested SparseHash on a classification experiment with a real dataset of news articles. This test confirmed the superior performance of the proposed method with respect to MinHash. Future work will focus on providing more detailed theoretical results, as well as a fast technique to compute the measurements that does not require the matrix-vector product.

6. REFERENCES

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[4] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, 1984.

[5] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 2002, STOC '02, pp. 380–388, ACM.

[6] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2082–2102, April 2013.

[7] P. T. Boufounos and S. Rane, "Efficient coding of signal distances using universal quantized embeddings," in *Data Compression Conference (DCC), 2013*, March 2013, pp. 251–260.

[8] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[9] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, Jan. 2008.

[10] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, London, UK, 1998, ECML '98, pp. 137–142, Springer-Verlag.

[11] Y. Jiang, C. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, New York, NY, USA, 2007, CIVR '07, pp. 494–501, ACM.

[12] A. Broder, "On the resemblance and containment of documents," in *Proceedings of the Compression and Complexity of Sequences 1997*, Washington, DC, USA, 1997, pp. 21–29, IEEE Computer Society.

[13] A. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," *Journal of Computer and System Sciences*, vol. 60, no. 3, pp. 630 – 659, 2000.

[14] P. Indyk, "A small approximately min-wise independent family of hash functions," in *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA, 1999, SODA '99, pp. 454–456, Society for Industrial and Applied Mathematics.

[15] P. Li and C. König, "b-bit minwise hashing," in *Proceedings of the 19th International Conference on World Wide Web*, New York, NY, USA, 2010, pp. 671–680, ACM.

[16] V. Bioglio, T. Bianchi, and E. Magli, "On the fly estimation of the sparsity degree in compressed sensing using sparse sensing matrices," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2015, pp. 3801–3805.

[17] C. Ravazzi, S. M. Fosson, T. Bianchi, and E. Magli, "Signal sparsity estimation from compressive noisy projections via γ-sparsified random matrices," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, p. to appear.

[18] S. Dasgupta and A. Gupta, "An Elementary Proof of a Theorem of Johnson and Lindenstrauss," *Random Struct. Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[19] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2006, KDD '06, pp. 287–296, ACM.

[20] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[21] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," *Comput. Netw. ISDN Syst.*, vol. 29, no. 8-13, pp. 1157–1166, Sept. 1997.

[22] M. Henzinger, "Finding near-duplicate web pages: A large-scale evaluation of algorithms," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2006, SIGIR '06, pp. 284–291, ACM.

[23] M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2013.