BEYOND "PROJECT AND SIGN" FOR COSINE ESTIMATION WITH BINARY CODES

Raghavendran Balu, Teddy Furon and Hervé Jégou

Inria

ABSTRACT

Many nearest neighbor search algorithms rely on encoding real vectors into binary vectors. The most common strategy projects the vectors onto random directions and takes the sign to produce so-called sketches. This paper discusses the sub-optimality of this choice, and proposes a better encoding strategy based on the quantization and reconstruction points of view. Our second contribution is a novel asymmetric estimator for the cosine similarity. Similar to previous asymmetric schemes, the query is not quantized and the similarity is computed in the compressed domain.

Both our contribution leads to improve the quality of nearest neighbor search with binary codes. Its efficiency compares favorably against a recent encoding technique.

Index Terms— Locality sensitive hashing, similarity search, approximate nearest neighbors, Hamming embedding

1. INTRODUCTION

From a large collection of vectors in a high dimensional space, the approximate most similar search aims at extracting the most similar vectors to a query. Similarity is usually measured as the cosine between two vectors. This problem has a practical interest in many applications such as multimedia content indexing, automatic medical diagnosis or recommendation systems based on nearest neighbors regression.

It is mandatory to care about the memory usage of the index when addressing large datasets including millions to billions vectors. A first class of methods partitions the vectors into clusters offline, in order to only compute the similarities between the query and the vectors belonging to a subset of clusters deemed relevant. A second method, called Hamming Embedding, designs a function that maps vectors in $\mathbb{R}^D$ to binary sketches in $\mathbb{B}^L$ such that the Hamming distance between sketches estimates the similarity between vectors. In recent papers [1, 2, 3, 4, 5], LSH is no longer considered in the context of probe algorithms, but employed as a Hamming Embedding. To the best of our knowledge, Charikar [6] was the first to estimate the angle between two Euclidean vectors based on their LSH sketches. Section 2 reviews his approach and the improvements about the hash function design as well as the asymmetric scheme proposed in the literature.

The asymmetric scheme computes similarity measurements from the query vector and the database sketches. In other words, the sketch of the query is not processed. We would like to find a new design with the following properties: (i) it is an Hamming embedding for the similarity based on the cosine between two vectors, (ii) it allows a simple reconstruction of the original vector from its sketch. The first property is crucial for efficiently finding a subset of the database containing similar vectors while the second property yields to an asymmetric scheme re-ranking these vectors by computing a better estimate from the query and their reconstructions. Yet, Section 3 outlines the suboptimalities of LSH from the viewpoint of reconstruction. In a previous paper, we have already proposed a design fulfilling the two properties but its complexity prevents its application to high dimensional space and/or large scale database. This is the reason why we propose in Section 4 a simple modification of LSH to boost its reconstruction ability while maintaining its efficiency. Section 5 shows experimental results demonstrating the good performances of our two-step approximate search which strikes a better trade-off between complexity and quality of search when compared to previous schemes.

These results have been partly produced in the framework of the common research lab between INRIA and Alcatel-Lucent Bell labs, and partly in the context of the ANR project Secular (ANR-12-CORD-014).

2. BACKGROUND

2.1. Cosine sketches

Cosine sketches are usually constructed [6] with random projections, each being defined by a vector $\mathbf{w}_j$, $j = 1 \dots L$. For any vector $\mathbf{x} \in \mathbb{R}^D$, each projection produces a bit:

$$b_j(\mathbf{x}) = \operatorname{sign} \mathbf{w}_j^\top \mathbf{x}. \tag{1}$$

The sketch of $\mathbf{x}$ is just the concatenation of these bits:

$$\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}), \dots, b_L(\mathbf{x})]\,. \tag{2}$$

Let assume that the projection direction is random and uniformly drawn on the unit sphere. The hyper-plane whose normal vector is $\mathbf{w}_j$ separates two vectors $\mathbf{x}$ and $\mathbf{y}$ with a probability related to the unoriented angle θ between $\mathbf{x}$ and $\mathbf{y}$. This gives the following key property:

$$\mathbb{P}(b_j(\mathbf{x}) \neq b_j(\mathbf{y})) = \frac{\theta}{\pi} \tag{3}$$

The expectation of the Hamming distance between the sketches is also related to this probability if the $\mathbf{w}_j$ are independently drawn: $\mathbb{E}(d_\mathsf{h}(\mathbf{b}(\mathbf{x}), \mathbf{b}(\mathbf{y}))) = L\mathbb{P}(b_j(\mathbf{x}) \neq b_j(\mathbf{y}))$. Therefore, the Hamming distance gives an unbiased estimator of the angle as

$$\hat{\theta} = \frac{\pi}{L} d_\mathsf{h}(\mathbf{b}(\mathbf{x}), \mathbf{b}(\mathbf{y}))\,. \tag{4}$$

The cosine function is decreasing over the range of the unoriented angle $[0, \pi]$. Therefore, ranking vectors by increasing order of the Hamming distance between their sketches and the sketch of a query approximates the ranking by increasing angle or decreasing cosine similarity.

Several researchers have proposed extensions to this initial framework, *e.g.*, by proposing other kernel estimations [7, 3, 8, 9]. Note that this approach has been introduced in different communities: For instance, spectral hashing [3], universal quantizer [10], and ℓ_2 binary sketches [11] are very similar.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2.2. Hash function design

The performance of sketches depends on the design of the hash functions. The random projections proposed by Charikar [6] are widely used for cosine similarity, however they do not offer the best results. We distinguish two cases.

$\star$ $L \leq D$**:** A set of orthogonal vectors yields better results than random projections [2, 12]. The methods performing a PCA rotation learned in a training set, such as spectral hashing [3], implicitly use orthogonal vectors.

$\star$ $L > D$**:** It is no longer possible to generate L orthogonal projections. The L projection vectors form an over-complete frame $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_L]$ [13]. A tight frame satisfying $\mathbf{W}.\mathbf{W}^\top \propto \mathbf{I}_D$ is better than random projections [12, 14]. Another concurrent strategy [15] takes the union of subsets of orthogonal vectors (called *super-bits*). This construction has not been compared to an uniform tight frame.

Another track of research aims at optimizing the projection directions in order to better reconstruct the small distances [16]. Similarly, a rotation matrix is optimized to balance the variance on the different components [17, 9] so that each bit gives the same approximation error. These works mainly differ by the way the optimization is carried out.

2.3. Asymmetric scheme with sketches

The main interest of sketches is their compactness. In a typical scenario, they allow storing a representation of millions to billions vectors in memory. However, the memory constraint is not critical for the query, as this one is processed online.

This observation motivates the use of asymmetric methods [11, 4, 17, 18, 19], in which databases vectors are encoded into short sketches but the query is kept uncompressed to avoid quantization error. The first proposal considered the Euclidean distances from the query $\mathbf{y}$ to separating hyperplanes to weight the Hamming distance [11]:

$$d_a(\mathbf{y}, \mathbf{b}(\mathbf{x})) = \sum_{j=1}^{L} (\mathbf{y}^\top \mathbf{w}_j).b_j(\mathbf{x}) \tag{5}$$

3. SUBOPTIMALITY OF PROJECT AND SIGN

Instead of considering the analysis which maps $\mathbf{x}$ into $\mathbf{b}(\mathbf{x})$, we take a look at the synthesis, *i.e.* the reconstruction of the direction pointed by a vector from its sketch. From now on, we restrict to vectors on the hypersphere: $\|\mathbf{x}\| = 1$. We only consider a very simple reconstruction:

$$\hat{\mathbf{x}} \propto \sum_{j=1}^{L} b_j(\mathbf{x}) \mathbf{w}_j = \mathbf{W}\mathbf{b}(\mathbf{x}). \tag{6}$$

The proportionality constant is set such that $\|\hat{\mathbf{x}}\| = 1$. In the sequel, we exclude degenerated cases s.t. $\sum_{j=1}^{L} b_j \mathbf{w}_j = \mathbf{0}$.

3.1. 'project and sign' is not a good quantizer

The new point of view of reconstruction/quantization stems in an interesting question about the binarization strategy. Formally, we have defined a codebook $\mathcal{C}$ comprising at most 2^L distinct centroids over the hypersphere. Does the centroid $\mathbf{c} \propto \mathbf{W}\mathbf{b}(\mathbf{x})$, induced by the selected sketch $\mathbf{b}(\mathbf{x})$, provides the best possible choice from a reconstruction point of view? The best centroid is the one maximizing the co-linearity to the input unitary vector $\mathbf{x}$ as

$$\mathbf{c}^\star(\mathbf{x}) = \underset{\mathbf{c} \in \mathcal{C}}{\text{arg-max}}\, \mathbf{x}^\top \mathbf{c} \tag{7}$$

$$\propto \mathbf{W} \underset{\mathbf{b} \in \mathbb{B}^L}{\text{arg-max}} \frac{\sum_{j=1}^{L} b_j\, \mathbf{x}^\top \mathbf{w}_j}{\| \sum_{j=1}^{L} b_j \mathbf{w}_j \|}. \tag{8}$$

Let first consider the case of an orthonormal set of vectors: $\mathbf{W}^\top \mathbf{W} = \mathbf{I}_L$. The denominator is then constant and the optimum is therefore obtained when b_j and $\mathbf{x}^\top \mathbf{w}_j$ have the same sign. Therefore, the "project+take sign" method is optimal for orthogonal frames with respect to quantization.

For the case $L > D$, the frame cannot be orthogonal and the above property does not hold, meaning that the best reconstruction may *not* take the sign of $\mathbf{x}^\top \mathbf{w}_j$.

$\star$ **Example:** Consider the frame

$$\mathbf{W} = [\mathbf{w}_1 \mathbf{w}_2 \mathbf{w}_3] = \begin{bmatrix} 1 & 0 & \cos\frac{\pi}{3} \\ 0 & 1 & \sin\frac{\pi}{3} \end{bmatrix} \tag{9}$$

The vector $\mathbf{x} \propto \mathbf{w}_1 + \mathbf{w}_2 - \mathbf{w}_3$ happens to have a sketch $\mathbf{b}(\mathbf{x}) = [1, 1, 1]$, whereas the best centroid is obviously $\mathbf{c}^\star(\mathbf{x}) \propto \mathbf{W}.[1, 1, -1]^\top = \mathbf{x}$. In other terms, projecting and taking the sign is suboptimal in this case. Indeed, the function $\mathbf{x} \mapsto \mathbf{b}(\mathbf{x})$ is not necessarily surjective as some sketches might never be selected. This implies a loss of capacity in the encoding scheme: although computed on L bits, the entropy of the sketches is lower than L bits.

At this stage, we mention that this problem is not solely due to the choice of the frame operator, but to the quantization procedure as well. Selecting the closest centroid in $\mathcal{C}$ to the input vector yields a better quantization as reported in Section 5. Yet this quantization is not possible for large values of L, for which browsing the whole set of centroids is not tractable.

3.2. Spread representations

These observations motivate a recent approach [12] for a better encoding strategy based on spread representations [20]. It reduces the quantization error underpinning the sign function.

The "anti-sparse coding" strategy first looks at

$$\mathbf{v}^\star(\mathbf{x}) = \arg \min_{\mathbf{v} \in \mathbb{R}^L : \mathbf{W}\mathbf{v} = \mathbf{x}} \|\mathbf{v}\|_\infty. \tag{10}$$

This goal resembles the objective of sparse coding, except that the ℓ_0 norm is replaced by ℓ_∞. As a result, instead of concentrating the signal representation on few components, anti-sparse coding has the opposite effect: It tends to spread the signal over all components, whose magnitude is comparatively less informative.

Interestingly, $L - D + 1$ components of $\mathbf{v}^\star(\mathbf{x})$ are stuck to the limit, *i.e.*, equal to $\pm\|\mathbf{v}^\star(\mathbf{x})\|_\infty$. As a result, this vector can be seen as a "pre-binarized version". The subsequent binarization to $\mathbf{b}(\mathbf{x}) = \text{sign}(\mathbf{v}^\star(\mathbf{x}))$ introduces less quantization loss than with the regular "project and sign" approach.

The main problem of anti-sparse coding is its low efficiency: Encoding a vector requires several matrix inversions, and the complexity strongly depends on the vector dimensionality [12]. Although this encoding step is done offline, it remains the bottleneck in practical setups involving billions of descriptors and is not tractable for high-dimensional vectors.

4. PROPOSED APPROACH: QUANTIZATION-OPTIMIZED LSH (qoLSH)

This section explains how we improve the cosine sketch detailed in Section 2.1 by adopting a quantization point of view. The section 3 illustrated the suboptimality of the "project and sign", from a reconstruction point of view, and the prohibitive cost of the optimal strategy due to the exponential increase in the number of centroids $|\mathcal{C}|$ with L.

Matrix W: We only use tight frames, as they generally offer better performance in this context [12, 14]. We randomly draw a $L \times D$ matrix with i.i.d. Gaussian entries. Then we compute its QR decomposition and set $\mathbf{W}$ as the first D rows of $\mathbf{Q}$, so that

$$\mathbf{W}.\mathbf{W}^\top = \mathbf{I}_D. \tag{11}$$

Computation of the sketch: Our approach is to alter the cosine sketch $\mathbf{b}$ of (1) in such a way that it decreases the reconstruction error. This way the Hamming distance between sketches still approximate the angle between their real vector counterparts. We alter the cosine sketch $\mathbf{b}$ by flipping sequentially individual bits that improves the reconstruction error.

For a given vector $\mathbf{x}$, its sketch $\mathbf{b}(\mathbf{x})$ and reconstructed vector $\hat{\mathbf{x}} = \mathbf{W}\mathbf{b}(\mathbf{x})$, we define the objective function as :

$$\mathcal{L}(\mathbf{b}) = \frac{\mathbf{x}^\top \hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|} \tag{12}$$

We start with sketch $\mathbf{b}_{(0)}$ of (1) and compute the reconstruction vector $\hat{\mathbf{x}}_{(0)}$. By flipping the j-th bit in $\mathbf{b}_{(0)}$, we get a new sketch $\mathbf{b}_{(1j)}$ and reconstruction vector

$$\hat{\mathbf{x}}_{(1j)} = \hat{\mathbf{x}}_{(0)} - 2b_j\mathbf{w}_j. \tag{13}$$

For L such bits in $\mathbf{b}$, we get L possible $\mathbf{b}_{(1j)}$ sketches. Out of L such sketches, we choose the one that maximizes the improvement in the objective function and call it $\mathbf{b}_{(1)}$, *i.e.*

$$\mathbf{b}_{(1)} = \text{arg-max}\, \mathcal{L}(\mathbf{b}_{(1j)}). \tag{14}$$

We now take $\mathbf{b}_{(1)}$ as the base sketch and find the next best bit to flip, which gives a new sketch $\mathbf{b}_{(2)}$, as described before. We continue this iteration until no bit flipping improves the objective function, or we reach a predefined number of iterations, say M.

The discrete nature of $\mathbf{b}$ makes exact optimization impossible. Changing a single coordinate at a time is suboptimal but it has a limited complexity.

Asymmetric scheme: The estimation of the similarity is based on the cosine of the angle between the query $\mathbf{y}$ and the reconstructed vector:

$$\cos(\mathbf{y}, \hat{\mathbf{x}}) = \frac{\mathbf{y}^\top \mathbf{W}\mathbf{b}(\mathbf{x})}{\|\mathbf{W}\mathbf{b}(\mathbf{x})\|} \tag{15}$$

$$= \frac{\sum_{j=1}^{L} (\mathbf{y}^\top \mathbf{w}_j) b_j(\mathbf{x})}{\|\mathbf{W}\mathbf{b}(\mathbf{x})\|}. \tag{16}$$

The major difference with (5) comes from the denominator.

5. EXPERIMENTS

This section evaluates our approach against the popular LSH sketch for cosine estimation [6] and a recent state-of-the-art search technique based on Anti-Sparse coding [12].

Table 1. Comparison of the properties of different binary sketch constructions on the synthetic dataset.

	MSE	entropy	Query time μs/vector
LSH	0.434	11.39	0.12
LSH+frame	0.207	12.47	0.12
Anti-sparse	0.142	14.23	1,307.40
Optimal	0.075	15.75	324.40
qoLSH	0.107	15.43	3.89

5.1. Evaluation protocol

The methods are evaluated on both synthetic and real datasets.

Synthetic dataset. We draw i.i.d vectors uniformly on the D-dimensional unit sphere, $D = 8$. For this purpose, we draw the vectors with normal distribution and normalized them to Euclidean unit norm. We produce $N = 1$ million vectors as database (indexed) vectors and $10,000$ queries vectors. The ground-truth is the (exact) cosine similarity.

Real dataset: SIFT1M. We also use a public dataset [21][1] of SIFT descriptors [22]. This dataset, referred to as SIFT1M, consists of $1,000,000$ database and $10,000$ query vectors of dimensionality $D = 128$.

Evaluation metrics. For both datasets, we compare the different methods based on recall@R curves: For each rank R, we measure the proportion of queries for which the true NN (Nearest Neighbor) appears in a position lower or equal to R.

Re-ranking. We adopt a two-stage retrieval procedure for all the methods. The first stage computes the similarities based on the binary codes and produce a short-list of 1,000 vector candidates based on fast Hamming-based computation: we order the vectors based on (4). This short-list is subsequently re-ordered with the asymmetric cosine estimation in (5), *i.e.*, we use the un-approximated query vector and compare it with short-list vectors reconstructed from their binary codes.

Encoding parameters. All the binarization methods considered in this section produce L-dimensional binary sketches. We set $L = 16$ for the synthetic dataset, in order to get a tractable complexity for the exhaustive optimal quantizer. For SIFT1M, we set $L = 256$. Note the optimal quantizer $\mathbf{c}^\star$ is not tractable for the SIFT dataset, as it is not possible to exhaustively list set of 2^L possible reconstruction values. Similarly, we mention that anti-sparse coding is not tractable with this parameter. The comparison with these two approaches is therefore only performed on the synthetic dataset.

For our method, we set $M = 5$ for the synthetic dataset and $M = 10$ for SIFT1M. The reconstruction quality is always better with higher values of M, however large values of M (*e.g.*, $M = L/2$) suffer the same problem as the optimal quantizer: the sketch is less stable w.r.t. perturbations of the input vector, yielding inferior results w.r.t. binary comparison.

[1] http://corpus-texmex.irisa.fr

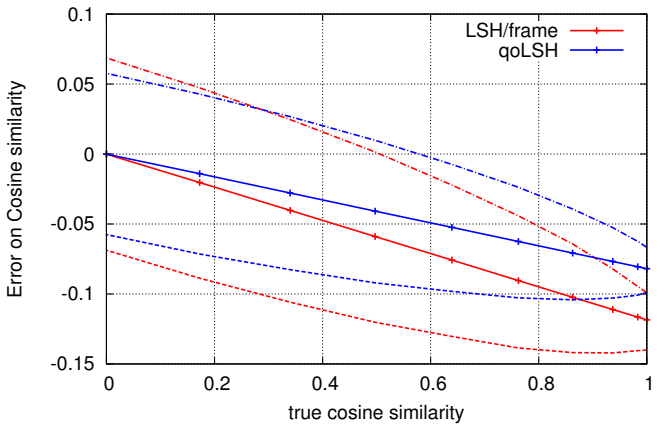

Fig. 1. Statistics about the difference between estimated and cosine similarities ($D = 128, L = 256$): mean (plain), 5% quantile (dash), 95% quantile (dot-dash).

5.2. Encoding analysis

Table 1 compares several sketch encoding methods based on (1) the quantizer performance measured by mean square error (MSE), (2) the empirical entropy and (3) the encoding time. We use the same tight frame for all the methods except "LSH": for the others, including LSH+Frame, only the encoding strategy differs. The optimal quantizer, by construction, offers the best quantization performance, see (8). Our method qoLSH is the best among the tractable encoding strategies. In particular the reconstruction is much better than LSH encoding (with the same frame). The encoding cost of qoLSH is larger than that of LSH, however it remains very efficient: encoding 1,000 vectors takes less than 4 ms. As a reference, computing 1,000×1 million Hamming distances takes 15.5 seconds. For larger datasets, the binary sketch computation associated with the query is negligible compared to Hamming distance computation.

Figure 1 plots some statistics (mean, 5% and 95% quantiles) of the difference between the true and estimate cosine similarities, *i.e.*, we show $(\cos(\mathbf{y}, \hat{\mathbf{x}}) - \cos(\mathbf{y}, \mathbf{x}))$ as a function of $\cos(\mathbf{y}, \mathbf{x})$. Compared to LSH, qoLSH decreases the bias and the estimation noise.

5.3. Search quality

Figure 2 compares the search performance of our algorithm with LSH and anti-sparse on both synthetic and real data with a 2-stage retrieval procedure (short-listing with binary codes and then asymmetric computation). Again, the optimal quantizer achieves the best results on the synthetic dataset, which confirms the importance of improving the quantizer. However, it is slow for $L \geq 20$, typically.

Our approach outperforms the optimal quantizer for large values of R. This is because the sketch comparison based on binary codes is better with our method than with this optimal quantizer, for which two nearby vectors may have very different sketches. This explains the saturation effect of the optimal quantizer observed in the figure. Note that all techniques gives different trade-offs from this point of view: anti-sparse coding is also appealing as the binary codes are even more stable than in our approach for large values of R.

For lower R values the performance of our algorithm is much better than LSH and anti-sparse. This assures higher chances of finding nearest neighbors in the top positions. The performance deteriorates after $R = 100$, because of the first binary filter.

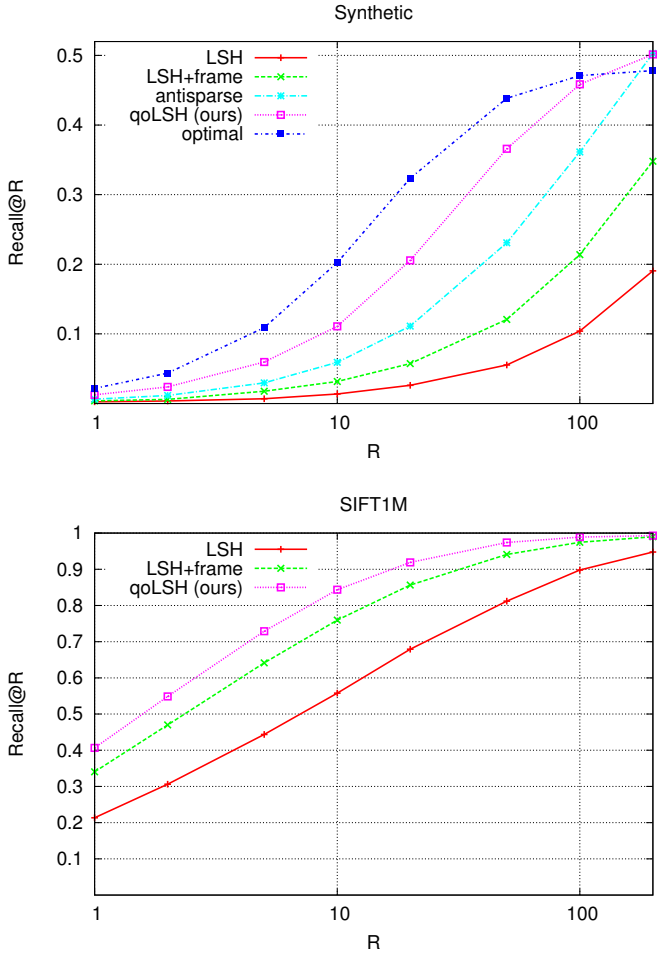

Fig. 2. Performance comparison: Recall@R on synthetic (top) and SIFT1M (bottom) dataset.

Overall, our approach gives a competitive trade-off: For typical values of M, the binary comparison is significantly better than that of regular LSH and slightly better than that of LSH+Frame and anti-sparse coding. After re-ranking with asymmetric distance computation, qoLSH exhibits a large gain over the other tractable methods.

6. CONCLUSION

This paper discusses the "project and sign" sketch construction method commonly used to estimate the cosine similarity in the compressed domain, and evidences that the method is sub-optimal when seen as a spherical quantizer. This is problematic in a context where the search is refined by considering the explicit reconstruction of a short-list of database vectors.

This leads us to define an alternative encoding strategy that offers significantly better performance both from quantization and approximate search points of view.

7. REFERENCES

[1] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large databases for recognition," in *CVPR*, June 2008.

[2] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, October 2008.

[3] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS*, December 2009.

[4] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *CVPR*, June 2010.

[5] A. Joly and O. Buisson, "Random maximum margin hashing," in *CVPR*, 2011.

[6] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *STOC*, May 2002, pp. 380–388.

[7] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *NIPS*, 2007.

[8] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *NIPS*, 2010.

[9] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *CVPR*, June 2011.

[10] P. T. Boufounos, "Universal rate-efficient scalar quantization," *IEEE Trans. Inform. Theory*, vol. 58, no. 3, March 2012.

[11] W. Dong, M. Charikar, and K. Li, "Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces," in *SIGIR*, July 2008, pp. 123–130.

[12] H. Jégou, T. Furon, and J. J. Fuchs, "Anti-sparse coding for approximate nearest neighbor search," in *ICASSP*, March 2012.

[13] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized overcomplete expansions in $\mathcal{R}^N$: Analysis, synthesis, and algorithms," *IEEE Trans. Inform. Theory*, vol. 44, no. 1, pp. 16–31, January 1998.

[14] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," Tech. Rep., Department of Engineering Science, University of Oxford, 2013.

[15] J. Ji, J. Li, S. Yan, B. Zhang, and Q. Tian, "Super-bit locality-sensitive hashing," in *NIPS*, December 2012.

[16] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *NIPS*, December 2009.

[17] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, June 2010.

[18] M. Jain, H. Jégou, and P. Gros, "Asymmetric hamming embedding," in *ACM Multimedia*, October 2011.

[19] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. PAMI*, vol. 33, no. 1, pp. 117–128, January 2011.

[20] J. J. Fuchs, "Spread representations," in *ASILOMAR*, November 2011.

[21] H. Jégou, R. Tavenard, M. Douze, and L. Amsaleg, "Searching in one billion vectors: re-rank with source coding," in *ICASSP*, Prague Czech Republic, 2011.

[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.