Learning Reconfigurable Hashing for Diverse Semantics

Yadong Mu[1], Xiangyu Chen[2,3], Tat-Seng Chua[3], Shuicheng Yan[1]
[1]Department of Electrical and Computer Engineering,
[2]NUS Graduate School for Integrative Sciences and Engineering,
[3]School of Computing,
National University of Singapore, 117576, Singapore
{elemy, chenxiangyu, chuats, eleyans}@nus.edu.sg

ABSTRACT

In recent years, locality-sensitive hashing (LSH) has gained plenty of attention from both the multimedia and computer vision communities due to its empirical success and theoretic guarantee in large-scale visual indexing and retrieval. Conventional LSH algorithms are designated either for generic metrics such as Cosine similarity, ℓ_2-norm and Jaccard index, or for the metrics learned from user-supplied supervision information. The common drawbacks of existing algorithms are their incapability to be adapted to metric changes, along with the inefficacy when handling diverse semantics (*e.g.*, more than 1K different categories in the well-known *ImageNet* database). For the metrics underlying the hashing structure, even tiny changes tend to nullify previous indexing efforts, which motivates our proposed framework towards "reconfigurable hashing". The basic idea is to maintain a large pool of over-complete hashing functions embedded in the ambient feature space, which serves as the common infrastructure of high-level diverse semantics. At the runtime, the algorithm dynamically selects relevant hashing bits by maximizing the consistency to specific semantics-induced metric, thereby achieving reusability of the pre-computed hashing bits. Such a reusable scheme especially benefits the indexing and retrieval of large-scale dataset, since it facilitates one-off indexing rather than continuous computation-intensive maintenance towards metric adaptation. We propose a sequential bit-selection algorithm based on local consistency and global regularization. Extensive studies are conducted on large-scale image benchmarks to comparatively investigate the performance of different strategies on reconfigurable hashing. Despite the vast literature on hashing, to our best knowledge rare endeavors have been spent toward the reusability of hashing structures in large-scale datasets.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding

ICMR '11, April 17-20, Trento, Italy

General Terms

Algorithms, Experimentation

Keywords

Locality sensitive hashing, reconfigurable hashing, random anchor, Shannon entropy

1. INTRODUCTION

With the explosive growth of available visual data in domains such as shared photos or video clips on the Web, efficient indexing for large-scale datasets becomes increasingly critical for image and video retrieval. Usually in such database all items are stored as uniformly-formatted high-dimensional feature vectors, and one simple yet essential operation is to efficiently find a set of nearest neighbors for an arbitrary query by comparing inter-feature proximity. A naive linear-scan implementation involves pairwise computations between the query and all items in the database. Fortunately, in most applications there is no need to identify the exact k-nearest neighbors. Instead, *approximate nearest neighbors* (ANN) achieve comparable performance in many scenarios, meanwhile greatly decreasing the computational cost. Recent progress witnessed the popularity of *locality sensitive hashing* (LSH) as an invaluable tool for retrieving approximate nearest neighbors in large-scale datasets. The basic idea of LSH is to cast data into independently generated hash buckets and normally it guarantees higher collision probability for similar data. The line of work has gained considerable empirical success in a variety of tasks such as image search, template matching, near-duplicate image or video retrieval, human pose estimation etc.

The key ingredient of an LSH algorithm is the unique metric that it works for. Original LSH algorithms are devised for uniform-length feature vectors equipped with "standard" metrics, including Jaccard Index [2], Hamming distance [8], ℓ_1-norm [1], ℓ_2-norm [1], Cosine similarity [3], or general ℓ_p-norm ($p \in (0, 2]$) [4]. Although accompanied with strict collision bound analysis, unfortunately it is seldom the case that pairwise similarity between visual identities (*e.g.*, images, three-dimensional shapes, video clips) are gauged using aforementioned metrics. Instead, the so-called Mercer kernels [16] provide more flexibility by implicitly embedding original features into high-dimensional Hilbert spaces. Representative Mercer kernels widely used by multimedia practitioners include the Radial Basis Function (RBF) kernel [16] and Pyramid Match Kernel (PMK) [6]. Previous study [9, 11, 7] shows the extension of LSH algorithms to the kernelized case is feasible.

Note that all of aforementioned metrics (including those induced from Mercer kernels) are explicitly pre-defined. More complications stem from the ambiguous metrics implicitly defined by a bunch

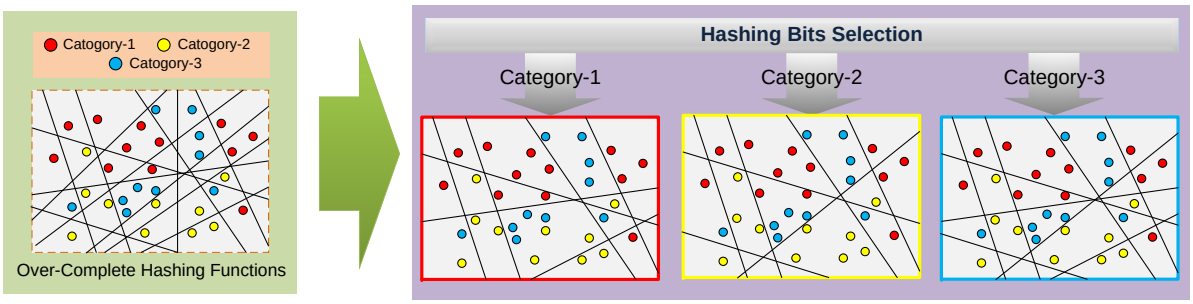

Figure 1: Illustration of reconfigurable hashing. The left subfigure shows a toy dataset and the pre-computed redundant hashing functions, while the contents on the right panel sketch the idea of optimal hashing bit selection toward specific semantic category.

of pairwise similarity (or dissimilarity) constraints, which frequently occurs in the research field of metric learning [22]. Hashing with this kind of partial supervision is challenging. Previous efforts address this task towards two directions: 1) hashing with learned metric [9], which transfigures the original metrics (typically via the modulation of Mahalonobius matrix) and then applies standard hashing techniques. 2) data-dependent hashing with weak supervision [11, 18], which seeks most consistent hashing hyperplanes by constrained optimization. Despite their success, existing techniques fail to handle the diverse semantics in real-world multimedia applications. The cruxes of the dilemma originate from two factors:

- The ambiguity and inconstancy of the semantics. An example is the visual semantics induced from pairwise affinity relationship, which is either constructed from user labeling or community-contributed tags. Unfortunately, both information sources are usually arbitrarily created and suffer from errors. Moreover, the collection of constraints is subject to frequent update, which potentially causes semantic drifting. Since both the hashing scheme and the resultant indexing structure are seriously hinged on the underlying semantics or metric, one-off data indexing is unfeasible under the circumstance of unstable semantics, which triggers unnecessary labors spent on indexing structure maintenance.

- The diversity of the semantics [20]. Prior study assumes the data are associated with limited number of disparate semantics, which is usually not the true case in real-world benchmarks. For example, the hand-labeled ImageNet dataset[1] contains more than ten million images depicting 10,000+ object categories. Simultaneous pursuit of optimal hashing for all categories is unwise considering the unknown and complex intrinsic data structures. Another possible solution is to separately conduct hashing for each unique category and concatenate all to form the final indexing structure, which unfortunately is uneconomic in terms of storage (actually the overlapped semantic subspace between two categories implies that several hashing bits can be shared to save the storage) and vulnerable to semantic changes and new emerging categories owing to the expensive re-indexing the large-scale dataset.

The above-mentioned drawbacks of existing methods motivate "reconfigurable hashing" proposed in this paper. Figure 1 illustrates the basic idea of reconfigurable hashing, whose basic operation is to generate a set of over-complete hash functions and perform one-off data indexing. When the semantic-related annotation or constraints arrive, the algorithms optimally choose a small portion of relevant hashing bits from the pool to best fit the target semantic metric. Unlike conventional methods, the proposed indexing framework supports unlimited number of diverse semantics based on one-off indexing, and also admits the adaptation to the changed metrics at low computational cost and zero re-indexing effort. In detail, our contributions in this paper can be summarized as below: 1) a novel hashing algorithm named *random-anchor-random-projection* (RARP), which is equivalent to redundant random partition of the ambient feature space and proves superior to other candidate LSH algorithms. Strict collision analysis for RARP is also supplied. 2) we discuss different strategies for optimal hash function selection and further proposed a sequential algorithm based on local consistence and global regularization. 3) the idea of reconfigurable hashing is content agnostic and consequently domain independent, but the performances of different selection strategies vary. Comparative investigation about the proposed and other candidate strategies is provided on four popular multiple-semantics image benchmarks, which validates the effectiveness of reconfigurable hashing and its scalability to large-scale data set.

The rest of the paper is organized as follows. Section 2 provides a brief survey of relevant literature. Section 3 defines the notations used in this paper and formally states the problem to be solved. Section 4 elaborates on the proposed formulation and also other alternative strategies. More details of the hashing collision analysis are found in Section 4.4. Extensive experiments are conducted on four real-world benchmarks in Section 5 and in Section 6 we give the concluding remarks and point out several directions for future work.

2. RELATED WORK

Existing LSH algorithms can be roughly cast into the following categories:

- **Element sampling or permutation**: Well-known examples include Hamming distance [8] and Jaccard Index [2]. For the Hamming case, the work in [8] presents a hashing scheme $h(x) = x_i$, where i is randomly sampled from the dimension index set $\{1, \ldots, d\}$ and x_i is the binary value of the i-th dimension. The guarantee of the locality sensitive property is also given in [8].

- **Project-Shift-Segment**: The idea is to map a point onto $\mathbb{R}^1$ along a projection direction in $\mathbb{R}^d$, randomly shift the projection values, and finally partition into intervals of length l_w (l_w is data-dependent parameter and need fine tuning). Examples include the algorithm for ℓ_1 norm [1], for Cosine similarity [3, 5] and for ℓ_p norm [4].

[1] http://www.image-net.org/challenges/LSVRC/2010/

- **Prototype-based methods**: Another LSH family uses predefined prototypes, such as polytopes on 24-D Leech lattice in ℓ_2 space [1] (i.e., E2LSH) or 8-D lattice [14].

- **Learning-based methods**: Assisted with semantic annotations or labels, LSH can be adapted via various learning methods like the classic SpectralHash [21] and SemanticHash [15]. Recent progress has also been made on hashing with weak supervision [18, 11] and sequential optimization [19].

From the brief survey in this section, it is observed that prior research is mainly focusing on designing LSH algorithms for specific metrics, while the task of our work aims to provide a meta-hashing method applicable in the existence of scalable diverse semantics and adaptive metrics. To our best knowledge rare related work can be found. It still lacks in-depth exploration and remains an open problem.

3. NOTATIONS AND PROBLEM SETTING

Before continuing, let us formally establish the notations and the problem setting. Denote $\mathcal{X} = \{x_1, \dots, x_n\}$ to be the set of feature vectors in $\mathbb{R}^d$. Let $h_i : \mathbb{R}^d \mapsto \{0, 1\}, i = 1 \dots m$ be m independently-generated hashing functions, where m is large enough to form an over-complete hashing pool. All samples in $\mathcal{X}$ are hashed to obtain binary bits according to the collection $\{h_i\}$. The hashing operation is performed only once and not required to be redone any more. The aim of reconfigurable hashing is to select compact hashing bit configuration from the pool to approximate any unknown metrics in terms of Hamming distance. Normally the maximum number of activated hashing functions (denoted as l) is budgeted and $l \ll m$. To define the target semantics or metrics, assume a fraction of data in $\mathcal{X}$ are associated with side information. Specifically we focus on the widely-used pairwise relationship [11, 18] throughout this paper,which reveals the proximal extent of two samples. Denote two sets as $\mathcal{M}$ or $\mathcal{C}$. Two arbitrary sample pair $(x_i, x_j) \in \mathcal{M}$ reflects the acknowledgement from the annotators that x_i, x_j semantically form a neighbor-pair in the context of target category. Similarly, $(x_i, x_j) \in \mathcal{C}$ implies that they are far away in the unknown metric space or have different class labels. Note that manual annotation is typically labor-intensive, therefore normally we assume that the labeled samples only cover a small portion of the whole dataset. Also for large-scale dataset associated with diverse semantics, the annotation is heavily unbalanced. In other words, the cardinality of $\mathcal{M}$ is far less than that of $\mathcal{C}$, which mainly follows from the fact that $\mathcal{C}$ is the amalgamation of all other non-target categories. A qualified algorithm on reconfigurable hashing is expected to survive in such settings.

Generally we can regard the hashing function h_i as a black box and only visit the binary hashing bits during the optimization. However, different hashing schemes notably affect the retrieval quality given budgeted hashing bits. For normalized feature vectors, one of the most popular hashing scheme is proposed by Charikar et al. [3], which is defined as below:

$$h(x) = \begin{cases} 0, & \text{if} \quad \omega^\top x < 0 \\ 1, & \text{if} \quad \omega^\top x \geq 0 \end{cases} \tag{1}$$

where the hashing vector ω is uniformly sampled from the unit hypersphere S^{d-1}. The collision probability is $Pr[h(x) = h(y)] = 1 - \Theta(x, y)/\pi$, where $\Theta(x, y) = \arccos\left(\frac{x \cdot y}{\|x\|\|y\|}\right)$ corresponds to the inter-feature angle. In this paper we target the data lying in the ℓ_p-normed spaces ($0 < p \leq 2$), and propose a hashing scheme named *random-anchor-random-projection* (called RARP hereafter), which belongs to the random projection based hash family yet differentiates itself from others by taking data distribution into account. To generate a hashing function, a sample x^o is randomly sampled from the dataset to serve as the so-called "anchor point". Also a random vector ω is sampled uniformly from the p-stable distribution [4]. The projection value can be evaluated as $\langle \omega, x - x^o \rangle = \langle \omega, x \rangle - b_{\omega, x^o}$, where $b_{\omega, x^o} = \langle \omega, x^o \rangle$ is actually the hashing threshold, *i.e.*,

$$h(x) = \begin{cases} 0, & \text{if} \quad \langle \omega, x \rangle < b_{\omega, x^o} \\ 1, & \text{if} \quad \langle \omega, x \rangle \geq b_{\omega, x^o} \end{cases} \tag{2}$$

where $\langle \omega, x \rangle$ denotes the inner product between ω and x. The collision analysis for RARP is discussed in Section 4.4.

In the hashing literature, it is common to utilize Hamming distance to approximate the distance or similarity in the original feature space, which is defined as below:

$$H(x, x') = \sum_{b=1}^{B} (h_b(x) \oplus h_b(x')), \tag{3}$$

where $\oplus$ denotes the logical XOR operation. Recall that the range of each hashing function is $\{0, 1\}$. Equation (3) can be expressed in a more tractable form:

$$\left\|h(x) - h(x')\right\| = \left(h(x) - h(x')\right)^T \left(h(x) - h(x')\right). \tag{4}$$

Here we adopt a generalized Hamming distance to ease numerical optimization. Specifically, we introduce the parametric Mahalonoisbu matrix M for modulating purpose. To ensure the positiveness of the resulting measure, M is required to reside in the positive semi-definite (p.s.d) cone, or mathematically $M \succeq 0$.

$$\|h(x) - h(x')\|_M = \left(h(x) - h(x')\right)^T \cdot M \cdot \left(h(x) - h(x')\right). \tag{5}$$

4. THE PROPOSED ALGORITHM

As a meta-hashing framework, the crucial operation in reconfigurable hashing is the selection of hashing bits. In this section, we present our proposed algorithm based on the averaged margin and global regularization, along with other four possible algorithms for the same task, based on random selection, maximum varaince, maximum local margin and Shannon information entropy respectively. The empirical evaluation of above methods is postponed to the experimental section.

4.1 Formulation

As stated above, we rely on sets $\mathcal{M}$ and $\mathcal{C}$ to determine the underlying semantics. However, the construction of pairwise relationship has quadratic complexity of the sample number. To mitigate the annotation burden, a practical solution is instead building two sets $\mathcal{L}_+$ and $\mathcal{L}_-$. The former set consists of the samples assigned to the target semantic label, and $\mathcal{L}_-$ collects the rest samples. We further generate *random homogeneous pair* and *random heterogeneous pair* to enrich $\mathcal{M}$ and $\mathcal{C}$ respectively. For each sample $x_i \in \mathcal{L}_+$, we randomly select $x_j \in \mathcal{L}_+$ with the guarantee $i \neq j$. The pair (x_i, x_j) is called *random homogeneous pair*. Likewise, given $x_k \in \mathcal{L}_-$, (x_i, x_k) constitutes a *random heterogeneous pair*. In such a way the construction of $\mathcal{M}$ and $\mathcal{C}$ is efficient.

Matrix M in Equation (5) can be eigen-decomposed to obtain $M = \sum_{k=1}^{K} \sigma_k w_k w_k^T$. To simplify numerical optimization, we impose $\sigma_k = 1$ such that $M = WW^T$ where $W = [w_1, \dots, w_K]$. Denote the index set $\mathcal{I}$ to be the collection of selected hashing bits at current iteration. Let $h_{\mathcal{I}}(x_i)$ be the vectorized hashing bits for x_i. Two margin-oriented data matrices can be calculated by traversing $\mathcal{M}$, $\mathcal{C}$ respectively and piling the difference column vectors,

i.e.,

$$X_m = \left\{h_{\mathcal{I}}(x_i) - h_{\mathcal{I}}(x_j)\right\}_{(x_i,x_j)\in\mathcal{M}}$$
$$X_c = \left\{h_{\mathcal{I}}(x_i) - h_{\mathcal{I}}(x_j)\right\}_{(x_i,x_j)\in\mathcal{C}}$$

We adopt the *averaged local margin* [17] based criterion to measure the empirical gain of $\mathcal{I}$, which is defined as below:

$$J(W) = \frac{1}{n_c} tr\{W^T X_c X_c^T W^T\} - \frac{1}{n_m} tr\{W^T X_m X_m^T W^T\} \quad (6)$$

where n_c and n_m are cardinalities of $\mathcal{C}$, $\mathcal{M}$ respectively. Intuitively $J(W)$ maximizes the difference between *random heterogeneous pair* and *random homogeneous pair* in terms of averaged Hamming distances, analogous to the concept of margin in kernel-based learning [16].

Moreover, prior work such as the well-known *spectral hashing* [21] observes an interesting phenomena, *i.e.*, hashing functions with balanced bit distribution tend to bring superior performance. In other words, the entire dataset is split into two equal-size partitions. Intuitively, balanced hashing function separates more nearest neighbor pairs. Coupling the independence condition of different bits, such a scheme results in more buckets. Consequently the collisions of heterogeneous pairs are reduced with high probability. Motivated by this observation, we introduce a global regularizer regarding bit distribution, *i.e.*,

$$\mathcal{R}(W) = \mathbb{E}\left(\|W^T(h_{\mathcal{I}}(x_i) - \mu)\|_2^2\right), \quad (7)$$

where μ represents the statistical mean of all hashing-bit vectors. In practice a small subset $\mathcal{X}_s$ with cardinality n_s is sampled and serves as statistical surrogate. Equation (7) can be rewritten as:

$$\mathcal{R}(W) = \frac{1}{n_s} tr(W^T X_s X_s^T W) - tr(W^T \mu\mu^T W). \quad (8)$$

For brevity, denote $L_J = X_c X_c^T/n_c - X_m X_m^T/n_m$ and $L_R = X_s X_s^T/n_s - \mu\mu^T$. Putting all together, finally we get the regularized objective function:

$$F(W) = tr(W^T L_J W) + \eta \cdot tr(W^T L_R W), \quad (9)$$

where $\eta > 0$ is a free parameter to control the regularizing strength. It is easily verified that

$$\max F(W) = \sum_{k=1}^{K} \lambda_k, \quad (10)$$

where $\{\lambda_k\}$ constitutes the non-negative eigenvalues of matrix $L_J + \eta \cdot L_R$ (the negative eigenvalues stem from the indefinite property of L_J) and the value of K is thereby automatically determined.

4.2 Greedy Sequential Bit Selection

Due to the large number of the hashing pool, global optimization is computationally forbidden. Here we employ a greedy strategy for sequential bit selection. In the t-th iteration, each unselected hashing function h_p is individually added into current index set $\mathcal{I}^{(t)}$ and the optimum of $F(W)$ under $\mathcal{I}^{(t)} \cup \{p\}$ is computed. The hashing function that maximizes the gain will be eventually added into $\mathcal{I}^{(t)}$. The procedure iterates until the hashing bit budget is reached.

Unfortunately, one potential selection bias is rooted in the term $tr\{W^T X_c X_c^T W^T\}$ in Equation (6), which can be equivalently expressed as $\sum_{(x_i,x_j)\in\mathcal{C}} W^T h_{ij} h_{ij}^T W^T$ with $h_{ij} = h_{\mathcal{I}}(x_i) - h_{\mathcal{I}}(x_j)$. Owing to the summation operation over the constraint set $\mathcal{C}$, the estimation is smooth and robust. However, recall that $\mathcal{C}$ is randomly rendered. In some extreme case, the selected optimal hashing functions may be trapped in the regions where the density of (x_i, x_j) is relatively high, resulting the zero-norm values of some difference vectors (i.e., $\|h_{ij}\|_0$) are extremely high.

To mitigate this selection bias, we truncate too-high zero-norm to avoid over-penalizing. Given a pre-defined threshold θ (in implementation we set $\theta = 5$, which is a conservative parameter since hashing buckets with distances larger than 5 are rarely visited in approximate nearest neighbor retrieval), we re-scale the difference vector via the following formula:

$$h_{ij} = \frac{\min\left(\|h_{ij}\|_0, \theta\right)}{\|h_{ij}\|_0} \cdot h_{ij}. \quad (11)$$

4.3 Alternative Strategies

Besides our proposed hashing bit selection strategy, we also explore other alternatives. In detail, we choose the following:

Method-I: random selection (RS). In each iteration, select a hashing bit from the pool by uniform sampling. The procedure terminates when maximum budgeted number of hashing functions is reached.

Method-II: maximum unfolding (MU). As previously mentioned, previous research has revealed the superior performance of balanced (or max-variance) hashing function. In other words, it prefers hashing schemes with maximum unfolding. This strategy selects top-ranked maximum-variance hashing bits from the pool.

Method-III: maximum averaged margin (MAM). Similar to Equation (6), we can compute the *averaged margin* of each hashing function in the pool according to the formula and keep top-scored hashing bits via greedy selection.

$$score(h_p) = \mathbb{E}_{(x_i,x_j)\in\mathcal{C}}(h_p(x_i) \oplus h_p(x_j)) - \mathbb{E}_{(x_i,x_j)\in\mathcal{M}}(h_p(x_i) \oplus h_p(x_j)). \quad (12)$$

Method-IV: weighted Shannon entropy (WSE). For each candidate in the pool, we calculate a score based on the Shannon entropy [10]. For completeness we give its definition. Assume the index set of data as L, two disjoint subsets L_l and L_r can be created by a Boolean test $\mathcal{T}$ induced by a hashing function $h(\cdot)$. The Shannon entropy is computed as:

$$S_C(L, \mathcal{T}) = \frac{2 \cdot I_{C,\mathcal{T}}(L)}{H_C(L) + H_{\mathcal{T}}(L)}, \quad (13)$$

where H_C denotes the entropy of the category distribution in L. Formally,

$$H_C(L) = -\sum_c \frac{n_c}{n} \log_2 \frac{n_c}{n}, \quad (14)$$

where n is the cardinality of L and n_c is the number of samples in the category with index c. Maximal value is achieved when all n_c are the same. Similarly, the *split entropy* $H_{\mathcal{T}}$ is defined for the test $\mathcal{T}$, which splits the data into two partitions:

$$H_{\mathcal{T}}(L) = -\sum_{p=1}^{2} \frac{n_p}{n} \log_2 \frac{n_p}{n}, \quad (15)$$

where n_p (p = 1 or 2) denotes the sample number in L_l or L_r. The maximum of $H_{\mathcal{T}}(L)$ is reached when the two partitions have equal sizes. Based on the entropy of L, the *impurity* of $\mathcal{T}$ can be calculated by the mutual information of the split, i.e.,

$$I_{C,\mathcal{T}}(L) = H_C(L) - \sum_{p=1}^{2} \frac{n_p}{n} H_C(L_p). \quad (16)$$

Intuitively, $S_C(L, \mathcal{T})$ prefers $\mathcal{T}$ that is as balanced as possible and meanwhile separates different categories. As aforementioned, in the setting of reconfigurable hashing, the numbers of labeled samples from target category and non-target categories are heavily unbalanced, therefore we re-scale the sample weights such that the summed weights for the target category and non-target categories are equal. Finally the hashing functions with the highest scores are kept.

4.4 Hashing Collision Probability

Before diving into the experimental results, we would like to highlight the asymptotic property of the proposed *random-anchor-random-projection* (RARP) hashing functions.

For two samples x_1 and x_2, let $c = \|x_1 - x_2\|_p$. In the hashing literature, it is well acknowledged [8] that the computational complexity of a hashing algorithm is dominated by $\mathcal{O}(n^\rho)$, where n is the dataset size and $\rho < 1$ is dependent on algorithm choice and c. Suppose ω determines the parametric random hashing hyperplane. It is known that $\langle \omega, x_1 - x_2 \rangle$ is distributed as cX, where X is drawn from the p-stable distribution. Denote the range of projected values as $R = \max_i \langle \omega, x_i \rangle - \min_i \langle \omega, x_i \rangle$ and let $\eta = \frac{\langle \omega, x^o \rangle - \min_i \langle \omega, x_i \rangle}{R}$ (x^o is the random anchor). Without loss of generality, we assume $\eta > 0.5$. Let $g_p(t)$ be the probability density function of the absolute value of the p-stable distribution. The collision probability of RARP can be written as

$$Pr\left(h_{\omega,x^o}(x_1) = h_{\omega,x^o}(x_2)\right) \approx \int_0^{\eta R} \frac{1}{c} g_p(\frac{t}{c})(\eta - \frac{t}{R})dt + \int_0^{(1-\eta)R} \frac{1}{c} g_p(\frac{t}{c})(1 - \eta - \frac{t}{R})dt \quad (17)$$

The two terms in Equation (17) reflect the chances that x_1, x_2 collides in the two sides of x^o respectively. Note that the equality relationship only approximately holds in (17) due to the uneven data distribution (computing the accurate probability involves double integrals along ω), and rigorously holds in case of uniform distribution. Moreover, when R is large enough and the uniformity holds, analytic bound for ρ exists. Analysis in this section follows closely [4], and the detailed proofs are omitted due to space limit:

THEOREM 1. *For any $p \in (0, 2]$ and $c > 1$, there exists hashing family $\mathcal{H}$ for ℓ_p-norm such that for any scalar $\gamma > 0$,*

$$\lim_{R \to \infty} \rho \leq (1 + \gamma) \cdot \max\left(\frac{1}{c}, \frac{1}{c^p}\right). \quad (18)$$

5. EXPERIMENTS

In this section we justify the effectiveness of the proposed reconfigurable hashing through empirical evaluations on four benchmarks: Caltech-101[2], MNIST-Digit[3], CIFAR-10 and CIFAR-100[4]. In the experiments we compare the proposed hashing bit selecting strategy with other alternatives presented in Section 4.3. To reduce the effect of randomness, all experiments are iterated 30 times to get the statistical average. By default we set $\eta = 0.5$ and choose both 4 samples from target category and non-target categories to construct $\mathcal{M}$ and $\mathcal{C}$. The size of the hashing pool is fixed to be 10K in all experiments unless otherwise mentioned. Figure 2 shows selected images in the adopted benchmarks.

[2]http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[3]http://yann.lecun.com/exdb/mnist/
[4]http://www.cs.utoronto.ca/~kriz/cifar.html

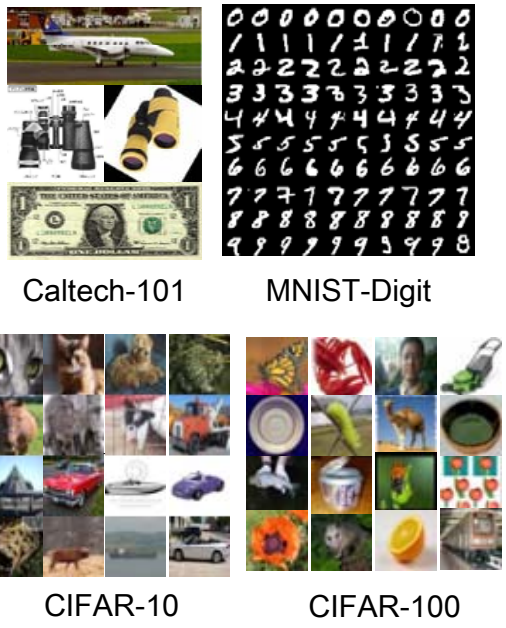

Figure 2: Exemplar images on selected benchmarks.

Table 1: Exemplar labels in the CIFAR-100 dataset.

COARSE LABELS	FINE LABELS
FISH	AQUARIUM FISH, FLATFISH, RAY, SHARK, TROUT
FLOWERS	ORCHIDS, POPPIES, ROSES, SUNFLOWERS, TULIPS
FOOD CONTAINERS	BOTTLES, BOWLS, CANS, CUPS, PLATES
TREES	MAPLE, OAK, PALM, PINE, WILLOW
PEOPLE	BABY, BOY, GIRL, MAN, WOMAN
VEHICLES 1	BICYCLE, BUS, MOTORCYCLE, PICKUP TRUCK, TRAIN
VEHICLES 2	LAWN-MOWER, ROCKET, STREETCAR, TANK, TRACTOR
REPTILES	CROCODILE, DINOSAUR, LIZARD, SNAKE, TURTLE
INSECTS	BEE, BEETLE, BUTTERFLY, CATERPILLAR, COCKROACH

5.1 Caltech-101 and CIFAR-100

Caltech-101 is constructed to test object recognition algorithms for semantic categories of images. The data set contains 101 object categories and 1 background category, with 40 to 800 images per category. As preprocessing, the maximum dimension of each image is normalized to be 480-pixel. We extract 5000 SIFT descriptors from each image whose locations and scales are determined in a random manner (see [13] for more details). For the visual vocabulary construction, we employ recently-proposed *randomized locality sensitive vocabularies* (RLSV) [12] to build 20 independent bag-of-words feature, each of which consists of roughly 1K visual words. Finally they are concatenated to form a single feature vector and reduced to be 1000-dimensional by dimensionality reduction.

CIFAR-100 is comprised of 60,000 images selected from 80M Tiny-Image dataset[5]. This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs). Table 1 presents some examples of these two-granularity categories. For the 32×32 pixel images, we extract ℓ_2-normalized 384-D GIST features.

[5]http://people.csail.mit.edu/torralba/tinyimages/

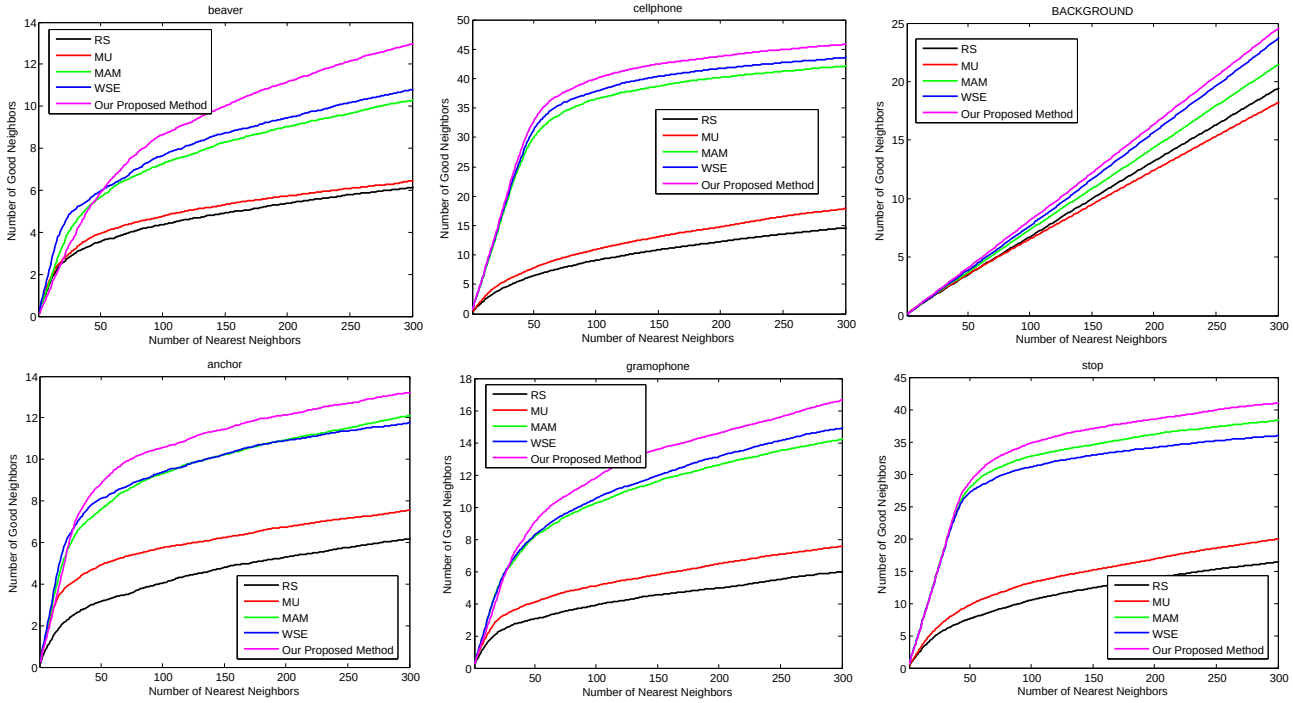

Figure 3: The evolution of accuracies of the first 300 retrieved samples on randomly-selected Caltech-101 categories.

We randomly generate 15 samples from each category in Caltech-101 and 30 samples for CIFAR-100 respectively, used for hashing bit selection. For each category, a unique hashing scheme is learned either by our proposed method or other methods mentioned in Section 4.3 with hashing bit budget equal to 14 on Caltech-100 or 16 on CIFAR-100, therefore in total 102 different hashing schemes for Caltech-101 and 100 for CIFAR-100. During the learning on specific category, the ensemble of the rest categories serves as the negative class. For each training sample from the target category, four random homogenous pairs and four random heterogenous pairs are generated by uniform sampling, forming the constraint sets $\mathcal{M}$ and $\mathcal{C}$ respectively.

We report the averaged results over 30 runs of our proposed method, along with the results obtained by four baselines, *i.e.*, random selection (RS), maximum unfolding (MU), maximum averaged margin (MAM) and weighted Shannon enropy (WSE). The results of naive linear scan (NLS) are also reported. However, recall that NLS utilizes no side information. There is no guarantee that NLS provides the upper-bound of the performance, as illustrated in the cases of Caltech-101 and CIFAR-100. We collect the proportions of "good neighbors" (samples belonging to the same category) in the first hundreds of retrieved samples (300 for Caltech-101, and 1000 for CIFAR-100). The samples within every bucket are randomly shuffled, and multiple candidate buckets with the same Hamming distance are also shuffled, so that the evaluation will not be affected by the order of the first retrieved samples (this operation is usually ignored in the evaluations of previous work). See Table 2 for the detailed experimental results, where the winning counts of each algorithm are also compared. To better illustrate the evolving tendencies of reconfigurable hashing, Figures 3 and 4 plot the accuracies of selected categories from Caltech-101 and CIFAR-100 respectively. It is observed that the plotted curves on CIFAR-100 have more gentle slopes compared with Caltech-101's, which reveals the different characteristics of underlying data distributions, *i.e.*, the samples from the same category in Caltech-101 gather more closely.

Table 2: Reconfigurable hashing on multi-category benchmarks Caltech-101 and CIFAR-100. The top table illustrates the averaged accuracies (%) of the first k retrieved samples ($k = 300$ for Caltech-101 and $k = 1000$ for CIFAR-100). We also count the number of categories on which an algorithm beats all the others (102 or 100 in total on these two benchmarks). The results are reported in the bottom table.

	RS	MU	MAM	WSE	OURS	NLS
CALTECH-101	3.99	4.92	10.31	10.15	11.08	10.20
CIFAR-100	1.75	2.01	3.93	3.93	4.26	4.09

	RS	MU	MAM	WSE	OURS
CALTECH-101	0	0	2	1	99
CIFAR-100	0	0	4	5	91

Although reconfigurable hashing is a meta-hashing framework, the ground hashing algorithms seriously affect the final performance. In Figure 5 we plot the logarithm of the accuracy for each category on Caltech-101, employing either our proposed RARP or conventional LSH as described in Equation (1). RARP shows superior performance, which indicates that data-dependent hashing algorithms such as RARP is promising for future exploration.

5.2 MNIST-Digit and CIFAR-10

The sample number of each category on Caltech-101 and CIFAR-100 is relatively small, ranging from 31 to 800. To complement the study in Section 5.1, we also conduct experiments on the benchmarks MNIST-Digit and CIFAR-10, which have larger sample number (6K or 7K) per category.

MNIST-Digit is constructed for handwritten digits recognition.

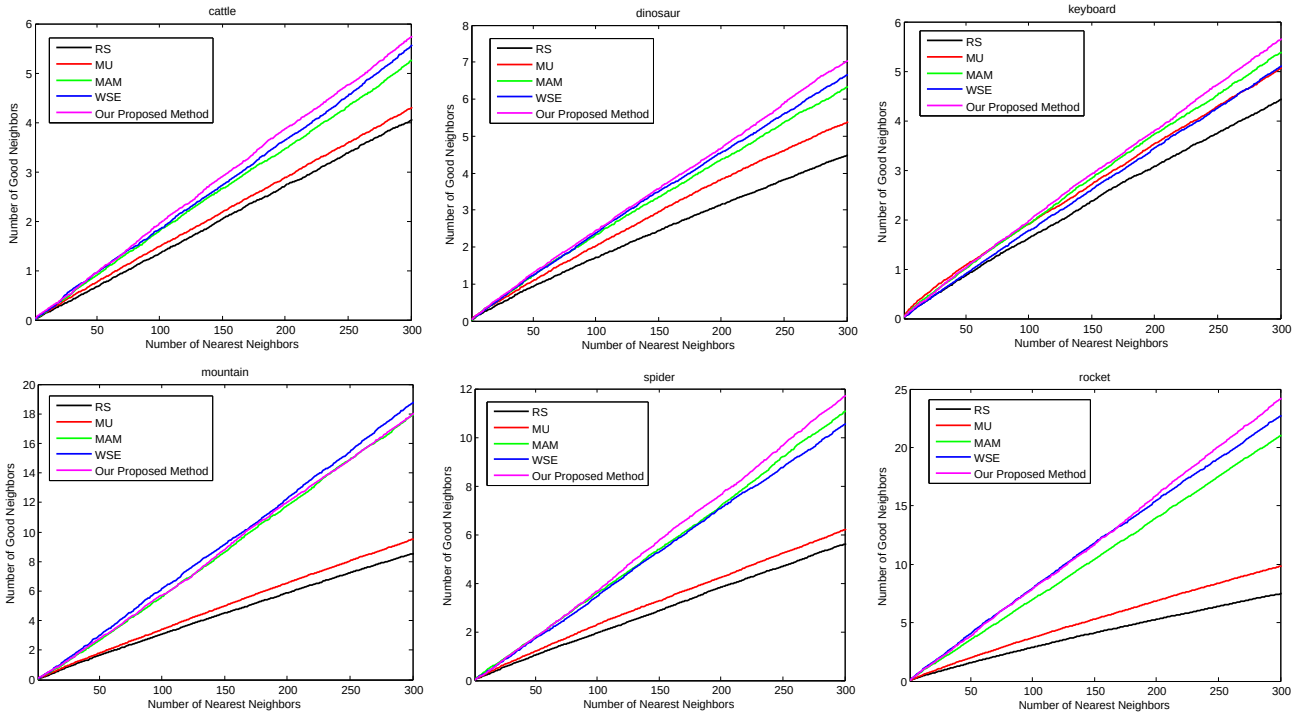

Figure 4: The evolution of accuracies of the first 1000 retrieved samples on randomly-selected CIFAR-100 categories.

It consists of totally 70,000 digit images, 7,000 images for each digit in $0 \sim 9$. The digits have been size-normalized to be 28×28 pixels. In our study, each digit image is transformed by matrix-to-vector concatenation and normalized to be unit-length feature. These raw grayscale vectors directly serve as the low-level feature for recognition purpose.

Similar to CIFAR-100, CIFAR-10 is also a labeled subset of the 80 million tiny images dataset, containing 60K 32×32 color images in 10 classes (6K images for each class). The dataset is constructed to learn meaningful recognition-related image filters whose responses resemble the behavior of human visual cortex. In the experiment we use the 387-d GIST image feature.

We learn category-dependent hashing schemes with 16 hashing bit budget. The experimental settings are identical to those on CIFAR-100, except that in the testing stage, only a portion of testing samples (300 in our implementation) are chosen for evaluation. Table 3 presents the results in terms of accuracy and winning count.

It is meaningful to investigate the correlation of the bucket number and the final performance. In Figure 6 we plot the bucket number for each of the ten categories averaged over 30 independent runs. It is observed that MU results in the largest bucket numbers, which is consistent to its design principle. However, the retrieval performance of MU is only slightly better than random selection (RS), which negates the hypothesis that increasing bucket number will promote the performance with high probability. In contrast, WSE has the fewest buckets compared with other three non-random algorithms, yet the performance is amazingly excellent (see Table 3). Intuitively, the Shannon entropy adopted in WSE favors hashing hyperplanes that cross the boundary between target category and its complemental categories. Such a strategy tends to keep the samples from target category stay closely in terms of Hamming distance and reduces unnecessary bucket creation. The high contrast between the small bucket number and high effectiveness suggests the intelligent category-aware bucket creation is crucial for reconfigurable hashing. On the other hand, although both MAM and our proposed strategy utilize the idea of averaged margin, the latter brings slightly larger bucket number, which is supposed to stem from the regularization term $\mathcal{R}(W)$ defined in Equation (7). And it is observed the combination of averaged margin and maximum unfolding improves the hashing quality.

Table 3: Reconfigurable hashing on two 10-category image benchmarks MNIST-Digit and CIFAR-10. The implications of the top and bottom tables are the same as in Table 2. Note that our proposed strategy wins on all the categories.

	RS	MU	MAM	WSE	OURS	NLS
CIFAR-10	14.23	15.58	21.06	20.51	21.92	25.14
MNIST-DIGIT	28.24	34.96	60.97	60.00	63.60	74.74

	RS	MU	MAM	WSE	OURS
CIFAR-10	0	0	0	0	10
MNIST-DIGIT	0	0	0	0	10

6. CONCLUSIONS

In this paper we investigate the possibility of effective hashing in the existence of diverse semantics and metric adaptation. We propose a novel meta-hashing framework based on the idea of reconfigurable hashing. Unlike directly optimizing the parameters of hashing functions in conventional methods, reconfigurable hashing constructs a large hash pool by one-off data indexing and then selects most effective hashing-bit combination at runtime. The contributions in this paper include a novel RARP based hashing algorithm for ℓ_p norm, a novel bit-selection algorithm based on averaged margin and global unfolding-based regularization, and a com-

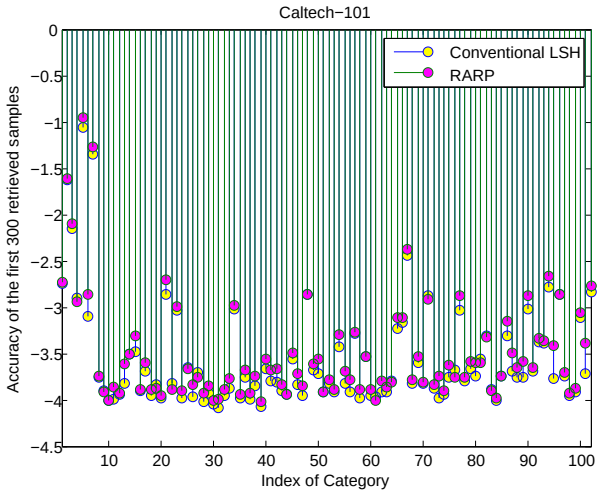

Figure 5: Comparison of conventional random projection based LSH (RP) and our proposed RARP on Caltech-101. Both utilize 14 bits in total. The accuracies for RARP and RP is 3.99% vs. 3.61% averaged on 102 categories, obtained by 30 independent runs. Note that the accuracies are transformed by logarithm function for better viewing.

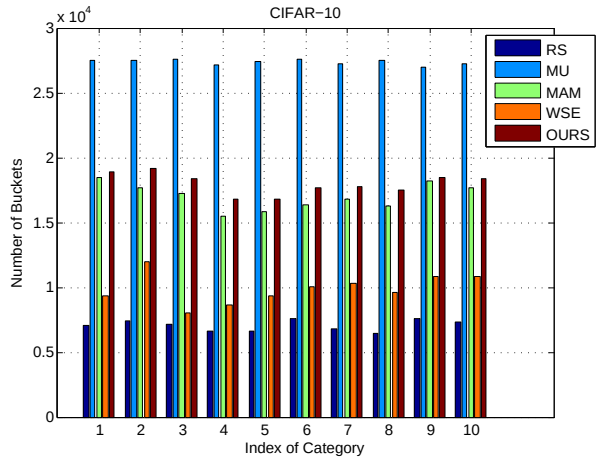

Figure 6: Comparison of bucket numbers on CIFAR-10.

parative study of various bit-selection strategies. For the future research direction, we are working towards two directions:

- How to identify the correlation of different hashing bits and then mitigate its adverse effect is still an open problem in reconfigurable hashing. Current techniques are far from satisfactory. We believe that some tools developed in the information theory community are helpful.
- The effectiveness of a hashing algorithm is heavily hinged on the characteristics of underlying data distributions. To develop a taxonomy about data distribution in the hashing context is especially useful.

Acknowledgement

This research is done for CSIDM Project No. CSIDM-200803 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

7. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, 2006.

[2] A. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, pages 327–336, 1998.

[3] M. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, 2002.

[4] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*, 2004.

[5] W. Dong, Z. Wang, M. Charikar, and K. Li. Efficiently matching sets of features with random histograms. In *ACM Multimedia*, 2008.

[6] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[7] J. He, W. Liu, and S.-F. Chang. Scalable similarity search with optimized kernel hashing. In *SIGKDD*, 2010.

[8] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*, 1998.

[9] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, 2009.

[10] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transactions on PAMI*, 30(9):1632 –1646, 2008.

[11] Y. Mu, J. Shen, and S. Yan. Weakly-supervised hashing in kernel space. In *CVPR*, 2010.

[12] Y. Mu, J. Sun, T. X. Han, L. F. Cheong, and S. Yan. Randomized locality sensitive vocabularies for bag-of-features model. In *ECCV*, 2010.

[13] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.

[14] L. Paulevé, H. Jégou, and L. Amsaleg. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*, 31(11):1348–1358, 2010.

[15] R. Salakhutdinov and G. Hinton. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, 2009.

[16] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[17] F. Wang and C. Zhang. Feature extraction by maximizing the average neighborhood margin. In *CVPR*, 2007.

[18] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 2010.

[19] J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. In *ICML*, 2010.

[20] M. Wang, Y. Song, and X.-S. Hua. Concept representation based video indexing. In *SIGIR*, 2009.

[21] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2008.

[22] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, 2002.