VECTOR COMPRESSION FOR SIMILARITY SEARCH USING MULTI-LAYER SPARSE TERNARY CODES

Sohrab Ferdowsi, Slava Voloshynovskiy, Dimche Kostadinov

Dep. of Computer Science, University of Geneva, Switzerland
{sohrab.ferdowsi, svolos, dimche.kostadinov}@unige.ch

ABSTRACT

It was shown recently that Sparse Ternary Codes (STC) posses superior "coding gain" compared to the classical binary hashing framework and can successfully be used for large-scale search applications. This work extends the STC for compression and proposes a rate-distortion efficient design. We first study a single-layer setup where we show that binary encoding intrinsically suffers from poor compression quality while STC, thanks to the flexibility in design, can have near-optimal rate allocation. We further show that single-layer codes should be limited to very low rates. Therefore, in order to target arbitrarily high rates, we adopt a multi-layer solution inspired by the classical idea of residual quantization. The proposed architecture, while STC in nature and hence suitable for similarity search, can add the "list-refinement" technique as a useful element to the similarity search setup. This can be achieved thanks to the excellent rate-distortion performance of this scheme which we validate on synthetic, as well as large-scale public databases.

Index Terms— learning to compress, Approximate Nearest Neighbor search, large-scale databases, rate-distortion

Notation: X and $\mathbf{X}$ denote random variables and random vectors, while their realizations are denoted as x and $\mathbf{x}$, respectively. All matrices use capital and up-right font as in X.

1. INTRODUCTION

Fundamental to any retrieval system is the ability to search for similar items within a database, w.r.t. query vectors provided by users. When this matching is performed based on a naïve exhaustive scan, however, as the scale of the database grows, this task becomes quickly intractable in terms of the computational complexity and also the memory storage requirements. Approximate Nearest Neighbor (ANN) search, therefore, tries to design compact codes to replace real-valued high-dimensional vectors in order to accommodate large-scale data in small memory and perform fast search within them.

The framework of Sparse Ternary Codes (STC) [1] was recently proposed as an alternative to the popular binary hashing to address this problem. While ternary encoding enjoys fast decoding speed similar to the efficient binary search, it was shown in [2] that for a fixed rate-budget, i.e., for a fixed amount of entropy, the STC framework preserves larger mutual information between encoded data and its noisy realizations as compared to the popular binary hashing. This higher "coding gain" of STC, enables it to achieve better trade-offs for performance, complexity and memory in general. Furthermore, interesting solutions [3], [4] based on STC were recently proposed for secure and privacy-preserving search.

In this work we focus on rate-distortion aspects of STC, i.e., we design compact STC such that we can further reconstruct the original high-dimensional features from them. While it is a known fact that better rate-distortion quality leads to better distance-preservation efficiency in general, we are further motivated by the idea of list-refinement, i.e., to perform an initial search very fast and in the domain of codes and further refine the short list of candidates by direct matching of their reconstructions with the query.

The important **contribution** of this paper is in the entire design of a network based on STC that "learns to compress" from the data. This is very challenging since from one hand, to be able to benefit from efficient similarity search, we are limited by constraints from STC, and from the other hand, we target arbitrarily low approximation distortions.

We realize this idea in two steps: First, we consider a single-layer code and study the optimality of projector and back-projector under some simplifying assumptions. We then provide an analytic expression for distortion and rate and show that the single-layer code has near asymptotically optimal performance, only at very small rates, or equivalently, higher sparsity levels. This is where we show that binary encoding (as a special case of STC with zero sparsity) falls into the high-rate regime and hence is suboptimal. Second, we use the classic idea of quantization of residuals to design multi-layer codes based on low-rate STC components.

The proposed structure is validated in two ways. First, on the synthetic *i.i.d.* and also correlated Gaussian data, we show rate-distortion performance close to the Shannon Lower Bound (SLB). We next show that the solution is "universal", meaning that from training sets, it can learn to compress test-set pixels from MNIST images and also Gist-1M set of descriptor features while largely outperforming state-of-the-art binary codes. We further show the effect of compression on similarity search recall measure for different rates for MNIST.

Section 2 formalizes the problem of compressive data-representation, particularly in view of fast search applications. After briefly reviewing the STC framework, section 3 focuses on the problem of reconstruction. The experiments are performed in section 4 and the paper is concluded in section 5.

2. PROBLEM FORMULATION: COMPRESSIVE DATA-REPRESENTATION FOR ANN SEARCH

Database $\mathrm{F} = [\mathbf{f}(1), \cdots, \mathbf{f}(N)]$, consists of data-points $\mathbf{f}(i)$'s $\in \Re^n$, as features representing an entity like human biometrics, images or image descriptors. When either N or n is large, for many tasks, it is crucial to store these vectors compressed. So we seek a compressive data representation scheme that should provide an encoder-decoder pair, $\mathbb{Q}[\cdot]$ and $\mathbb{Q}^{-1}[\cdot]$, such that the codes (representations) are as compact as possible. Furthermore, when decoded, the codes should closely approximate the original data. The first requirement is characterized by rate and the second by distortion.

More formally, for any realization vector $\mathbf{f}$, the rate of the representation $\mathbf{x} = \mathbb{Q}[\mathbf{f}]$ is defined as in (1a) and the distortion of the reconstruction $\hat{\mathbf{f}} = \mathbb{Q}^{-1}[\mathbf{x}]$ is defined as in (1b)[1], where we define the squared-error between two n-vectors $\mathbf{a}$ and $\mathbf{b}$ as $d(\mathbf{a}, \mathbf{b}) \triangleq \frac{1}{n}||\mathbf{a} - \mathbf{b}||_2^2$, and $\mathbb{E}[\cdot]$ is the expectation operator.

$$\mathcal{R} = \frac{1}{n}\mathbb{E}[\text{\# bits used}] \tag{1a}$$

$$\mathcal{D} = \mathbb{E}[d(\mathbf{F}, \hat{\mathbf{F}})] \tag{1b}$$

Within the similarity search domain, for many applications, a noisy query $\mathbf{q}$ is introduced and it is desired to find $\mathcal{L}(\mathbf{q}) = \{1 \leqslant i \leqslant N | d(\mathbf{f}(i), \mathbf{q}) \leqslant \epsilon n\}$, a list of most similar items to $\mathbf{q}$ among F. Since for large-scale problems, F is not available in memory and also direct matching with it is computationally expensive, an approximative $\hat{\mathcal{L}}(\mathbf{q})$ is preferred using low-complexity matching of $\mathbf{x}(i) = \mathbb{Q}[\mathbf{f}(i)]$ vs. $\mathbf{y} = \mathbb{Q}[\mathbf{q}]$, rather than $\mathbf{f}(i)$ vs. $\mathbf{q}$. So we require $\mathbb{Q}[\cdot]$ to be additionally compatible with this framework.

The efficiency of the decoder $\mathbb{Q}^{-1}[\cdot]$ can be doubly important for search applications since we can re-order and prune the inaccurate $\hat{\mathcal{L}}(\mathbf{q})$ by reconstructing $\mathbf{f}(i)$'s with $i \in \hat{\mathcal{L}}(\mathbf{q})$ and directly match them with $\mathbf{q}$ based on $d(\hat{\mathbf{f}}(i), \mathbf{q})$.

While this idea of "list-refinement" has not been particularly emphasized in the literature, lots of algorithms for fast search directly target distortion minimization as their main objective. Among the very broad literature, not to mention the family of VQ-based methods like PQ [5] and OPQ [6] which are codebook-based, many examples from the family of binary hashing methods also aim at distortion minimization. We can mention, e.g., the successful ITQ [7], which iteratively learns a projector matrix to minimize the distortion of the projected data and the corresponding binary codes. Another attempt is the Sparse Projections [8], an extension of ITQ for higher rates using similar objective.

3. PROPOSED: RECONSTRUCTION FROM STC

We first review the basic STC framework for fast search in section 3.1. Based on its specifications, we design the single-layer reconstruction from the STC in section 3.2 to its best. We then extend it to ML-STC, the multi-layer version to achieve near optimal distortion for all rate-regimes.

3.1. STC framework for fast search

The STC consists of a projection (assumed here to be square), followed by a ternary quantization. More formally, the STC corresponding to $\mathbf{f}$, i.e., $\mathbf{x} = \mathbb{Q}_{\text{STC}}[\mathbf{f}]$ with threshold λ_X is:

$$\mathbf{x} = \phi_{\lambda_X}(\mathrm{A}\mathbf{f}) \odot \boldsymbol{\beta}, \tag{2}$$

where $\phi_\lambda(x) = \text{sign}(x) \cdot \mathbb{1}_{\{|x| > \lambda\}}$ is the element-wise ternary thresholding operator, '$\odot$' is the Hadamard product and $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_n]^T$ is a re-weighting vector which is independent of a particular $\mathbf{f}$ and is fixed for all database vectors[2]. Therefore, each element of $\mathbf{x}$, i.e., x_i has a ternary alphabet $\mathcal{X}_i = \{\pm\beta_i, 0\}$. In practice, λ_X is chosen such that $\mathbf{x}$ is sparse. The motivation behind such an encoding comes from similarity search where the memory and computational complexity requirements encourage sparsity and a fixed-point alphabet.

After the enrollment of all $\mathbf{x}(i)$'s (usually stored in look-up-tables), a query vector $\mathbf{q}$ undergoes similar encoding, i.e., $\mathbf{y} = \phi_{\lambda_Y}(\mathrm{A}\mathbf{q})$ and is matched with $\mathbf{x}(i)$'s to produce $\hat{\mathcal{L}}(\mathbf{q})$. This matching can be performed very fast, e.g., using fixed-point sparse matrix multiplications.

3.2. Single-layer architecture

We formulate reconstruction of STC, i.e., $\hat{\mathbf{f}} = \mathbb{Q}_{\text{STC}}^{-1}[\mathbf{x}]$ as:

$$\hat{\mathbf{f}} = \mathrm{B}\mathbf{x} = \mathrm{B}\phi_\lambda(\mathrm{A}\mathbf{f}) \odot \boldsymbol{\beta}. \tag{3}$$

Where B is the reconstruction matrix that can be learned from the training data. However, in order to avoid over-fitting, the forward projection step using A should be imposed as a structure to help training. So we decompose as $\mathrm{B} = (\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T\mathrm{B}'$ and instead optimize B':

$$\mathrm{B}' = \underset{\mathrm{B}'}{\text{argmin}}\, ||\mathrm{F} - (\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T\mathrm{B}'\mathrm{X}||_{\mathcal{F}}^2, \tag{4}$$

where $|| \cdot ||_{\mathcal{F}}$ is the Frobenius norm for a matrix. This can easily be re-expressed as:

$$\begin{aligned}\mathrm{B}' &= \underset{\mathrm{B}'}{\text{argmin}}\, ||(\mathrm{A}^T\mathrm{A})\mathrm{F} - \mathrm{A}^T\mathrm{B}'\mathrm{X}||_{\mathcal{F}}^2 \\ &= \underset{\mathrm{B}'}{\text{argmin}}\, \mathrm{Tr}\Big[-2\mathrm{A}\mathrm{A}^T\mathrm{A}\mathrm{F}\mathrm{X}^T\mathrm{B}'^T + \mathrm{B}'\mathrm{X}\mathrm{X}^T\mathrm{B}'^T\mathrm{A}\mathrm{A}^T\Big].\end{aligned}$$

[1] For unknown $p(\mathbf{f}, \hat{\mathbf{f}})$,$\hat{\mathcal{D}} = \frac{1}{N}\sum_{i=1}^N d(\mathbf{f}(i), \hat{\mathbf{f}}(i))$, from a test set.

[2] In [2] and [1], reconstruction was not considered and we had $\boldsymbol{\beta} = \mathbf{1}$.

Derivating w.r.t. B' and equating to zero gives:

$$\mathrm{B}' = \mathrm{AFX}^T(\mathrm{XX}^T)^{-1}.$$

Assuming $\mathbf{F}$ to have a covariance matrix C_F, i.e., $\mathrm{C}_F = \frac{1}{n}\mathbb{E}[\mathbf{F}\mathbf{F}^T]$, we choose $\mathrm{A} = \mathrm{U}_F^T$ as in PCA, where $\mathrm{C}_F = \mathrm{U}_F \Sigma_F \mathrm{U}_F^T$ is the eigenvalue decomposition of C_F.

Therefore, the projected data $\tilde{\mathbf{x}} \triangleq \mathrm{A}\mathbf{f}$ is de-correlated as $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \Sigma_F)$[3], where $\Sigma_F = \mathrm{diag}\big([\sigma_1^2, \cdots, \sigma_n^2]^T\big)$ with σ_i^2's being the eigenvalues of C_F which are decaying in value for the correlated $\mathbf{F}$.

In our experiments, it turns out that with this choice of A, and the optimal re-weighting vector $\boldsymbol{\beta}$ for $\mathbf{x}$ which will be described shortly, B' indeed converges to the identity matrix as N, the number of training samples grows larger. This means that it suffices to choose $\mathrm{B}' = \mathbb{I}_n$. Equivalently stated, $\mathrm{B} = \mathrm{A}^T = \mathrm{U}_F$ would be the optimal back-projector of $\mathbf{x}$ to $\mathbf{f}$ under this setup.

We can characterize the expected distortion for reconstruction of a random vector $\mathbf{F}$ from $\mathbf{X}$. Emphasizing the orthonormality of A, we can then write:

$$\begin{aligned}\mathcal{D} &= \mathbb{E}\big[d(\mathbf{F}, \hat{\mathbf{F}})\big] = \frac{1}{n}\mathbb{E}\big[||\mathbf{F} - \mathrm{A}^T\mathbf{X}||_2^2\big] \\ &= \frac{1}{n}\mathbb{E}\big[||\mathrm{A}\mathbf{F} - \mathbf{X})||_2^2\big] = \frac{1}{n}\mathbb{E}\big[||\tilde{\mathbf{X}} - \phi_\lambda(\tilde{\mathbf{X}}) \odot \boldsymbol{\beta}||_2^2\big].\end{aligned}$$

This links the distortion in the original domain with that of the projection domain. Now we should find the optimal re-weighting vector $\boldsymbol{\beta}$.

We had that $\tilde{X}_i$'s, i.e., the elements of $\tilde{\mathbf{X}} = [\tilde{X}_1, \cdots, \tilde{X}_n]^T$ are distributed as $\tilde{X}_i \sim p(\tilde{x}_i) = \mathcal{N}(0, \sigma_i^2)$. The total distortion $\mathcal{D}$ is the sum of the distortions at each dimension as $\mathcal{D} = \sum_{i=1}^n D_i$, which can then be written as:

$$\begin{aligned}D_i &= \mathbb{E}\big[(\tilde{X}_i - \beta_i\phi_\lambda(\tilde{X}_i))^2\big] \\ &= \int_{-\infty}^{-\lambda}(\tilde{x}_i + \beta_i)^2 p(\tilde{x}_i)d\tilde{x}_i + \int_{-\lambda}^{+\lambda}\tilde{x}_i^2 p(\tilde{x}_i)d\tilde{x}_i + \\ &\int_{+\lambda}^{+\infty}(\tilde{x}_i - \beta_i)^2 p(\tilde{x}_i)d\tilde{x}_i,\end{aligned}$$

This integration leads to the expression of distortion and its minimizer β_i^* using the q-function $\mathcal{Q}(\cdot)$ as:

$$D_i = \sigma_i^2 + 2\beta_i^2\mathcal{Q}\Big(\frac{\lambda}{\sigma_i}\Big) - \frac{4\beta_i\sigma_i}{\sqrt{2\pi}}\exp\Big(\frac{-\lambda^2}{2\sigma_i^2}\Big), \tag{5a}$$

$$\beta_i^* = \underset{\beta_i}{\mathrm{argmin}}\, D_i = \frac{\sigma_i \exp\Big(\frac{-\lambda^2}{2\sigma_i^2}\Big)}{\sqrt{2\pi}\mathcal{Q}\Big(\frac{\lambda}{\sigma_i}\Big)}. \tag{5b}$$

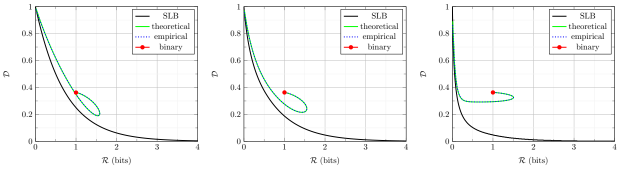
As a summary of the single-layer reconstruction from STC, first for the encoding, A is chosen as the eigenvectors of C_F which de-correlates the projected data $\tilde{\mathbf{x}}$. The ternarization is then performed according to (2), for which the elements of $\boldsymbol{\beta}$ are derived according to (5b). For decoding, the reconstruction is done by (3), where we showed that $\mathrm{B} = \mathrm{A}^T$ is the optimal choice.

Having calculated $\mathcal{D}$ as a function of C_F and λ, we now derive $\mathcal{R}$ using the ternary entropy $H_t(\cdot)$ as:

$$\begin{aligned}&\mathcal{R} = \frac{1}{n}H_t(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^n H_t(X_i) = \\ &-\frac{1}{n}\sum_{i=1}^n\Big(2\alpha_i\log_2(\alpha_i) + (1 - 2\alpha_i)\log_2(1 - 2\alpha_i)\Big),\end{aligned} \tag{6}$$

which follows from the fact that $\tilde{\mathbf{X}}$ and hence $\mathbf{X}$ are de-correlated and hence we can assume their approximative independence. For a ternary random variable X_i, α_i is defined as $\alpha_i = \mathbb{P}[X_i = +\beta_i] = \mathbb{P}[X_i = -\beta_i]$ and completely characterizes the ternary entropy $H_t(\cdot)$. For the above setup, this can be calculated for every X_i, simply as $\alpha_i = \mathcal{Q}\big(\frac{\lambda}{\sigma_i}\big)$.

Fig. 1 shows the rate-distortion behavior of the single-layer STC for 3 different sources: (a) *i.i.d.*, (b) AR(1) with $\rho = 0.5$ corresponding to mildly-correlated signals and (c) AR(1) with $\rho = 0.9$ corresponding to highly-correlated signals. For every figure, three curves are shown: the Shannon Lower Bound (SLB) derived from (7) which is the theoretical lower bound achieved only in the asymptotic case of $n \to \infty$ for any lossy source-coding scheme, the theoretical characterization of the STC distortion derived from (5a) and the empirical distortion calculated from simulations performed on $N = 10,000$ vectors of dimension $n = 500$ generated randomly. Also the case of binary encoding, i.e., zero sparsity, corresponding to $\mathcal{R} = 1$ is marked.



(a) *i.i.d.* (b) mid-correlation) (c) high-correlation

Fig. 1: Distortion-rate curves for single-layer STC (The 'theoretical' and 'empirical' curves coincide very closely.)

We clearly see that at lower sparsity levels (including the binary case) corresponding to higher rate-regimes, the single-layer structure has very poor performance. This phenomenon is due to sub-optimal rate allocation as we will describe next.

3.2.1. *Optimality of rate-allocation*

The Shannon theory characterized the optimal rate allocation for $\tilde{\mathbf{X}} \sim \mathcal{N}\big(\mathbf{0}, \Sigma\big)$ (see Ch.10 of [9]). For a given $\mathcal{D}$, this is achieved when $R_i = \frac{1}{2}\log_2\big(\frac{\sigma_i^2}{D_i}\big)$ bits are allocated for each

[3] Gaussianity assumption in the projected domain is justified from CLT.

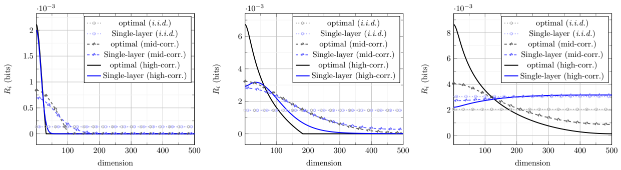
$\tilde{X}_i$, where D_i, corresponding distortion of each $\tilde{X}_i$ is:

$$D_i = \begin{cases} \lambda, & \text{if } \sigma_i^2 \geqslant \lambda \\ \sigma_i^2, & \text{if } \sigma_i^2 < \lambda, \end{cases} \tag{7}$$

and λ is chosen such that $\sum_{i=1}^{n} D_i = \mathcal{D}$. The total rate is then calculated as:

$$\mathcal{R}(\mathcal{D}) = \sum_{i=1}^{n} R_i = \sum_{i=1}^{n} \frac{1}{2} \log_2 \Big(\frac{\sigma_i^2}{D_i} \Big). \tag{8}$$

Comparing this optimal rate allocation of (8) with the single-layer STC of (6) reveals the fact that while they closely approximate the optimal rule at low rates, single-layer structure largely deviates from the optimal allocation at higher rates. This explains the saturating behavior of the rate-distortion curve at Fig. 1. This phenomenon is illustrated in Fig. 2.



(a) low-rate (b) mid-rate (c) high-rate

Fig. 2: Rate allocation of single-layer STC compared to the optimal rule, under three different rate regimes and for three different sources (same as in Fig. 1).

3.3. Multi-layer architecture

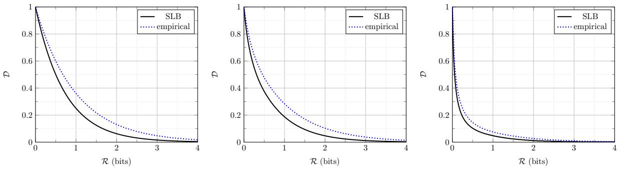
The mismatch between rate-allocation of the single-layer STC and the optimal rule at high rates limits their efficient use only for very low rate regimes. To compensate for this, we target higher rates in multi layers. This idea is related to some of the classical source-coding schemes like Residual Quantization (RQ) (see e.g., [10], [11] and [12]), although RQ is based on K-means and our solution here is essentially a projection-based encoding. This idea is as follows:

$$\begin{aligned} \mathbf{x}^{[l]} &= \phi_{\lambda_X}^{[l]}(\mathrm{A}^{[l]}\mathbf{f}^{[l-1]}) \odot \boldsymbol{\beta}^{[l]}, \\ \mathbf{f}^{[l]} &= \mathbf{f}^{[l-1]} - \mathrm{B}^{[l]}\mathbf{x}^{[l]}. \end{aligned} \tag{9}$$

The superscripts depict the index of the layer $l = 1, \cdots, L$. $\mathbf{f}^{[l]}$is the input to the algorithm at layer l which is the residual of the approximation from layer $l-1$ and is initialized as $\mathbf{f}^{[0]} = \mathbf{f}$. The rest of the procedure is the same as the single-layer case. Fig. 3 shows the success of this idea in rate-distortion within the same setup as Fig. 1.

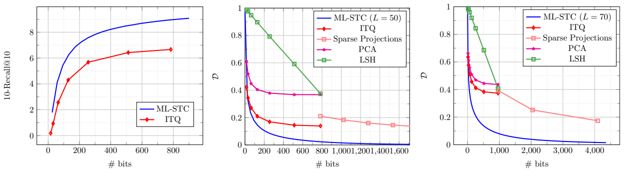
4. EXPERIMENTS

Here we demonstrate the performance of the proposed algorithm on the task of database compression and NN search from reconstructed vectors. We use two public databases: MNIST of mid-scale and the large-scale Gist-1M set [5]. The MNIST contains $60,000$ train and $10,000$ test images with 784 pixels which we consider as feature vectors. The Gist-1M comprises of 960-dimensional Gist descriptors with $500,000$ train and 1 million test vectors. Along with our ML-STC, we also experiment with the ITQ [7], the Sparse Projections [8] (using sparsity $= 50\%$), PCA hashing and the LSH (SimHash) [13]. We train all algorithms on the training and calculate the distortion on the test set. The reconstruction from the binary codes consists of pseudo-inversion and, for PCA hashing, the above-mentioned re-weighting stage followed by the inversion. For the ITQ and Sparse Projections, the vector $\boldsymbol{\beta}$ is irrelevant and also detrimental according to their objective functions. Instead, a scalar-valued optimal β is learned from the training set as $\beta = \frac{\mathrm{Tr}[\mathrm{F}\hat{\mathrm{F}}^T]}{\mathrm{Tr}[\hat{\mathrm{F}}\hat{\mathrm{F}}^T]}$ and multiplied globally as $\hat{\mathrm{F}} \leftarrow \beta\hat{\mathrm{F}}$. Fig. 4 sketches the results of these experiments. The ML-STC outperforms others with a large margin. **Python code** for our ML-STC can be found here in Github.



(a) low-correlation (b) mid-correlation) (c) high-correlation

Fig. 3: Rate distortion performance of multi-layer STC



(a) MNIST (b) MNIST (c) Gist-1M

Fig. 4: (a) Search quality from reconstructed codes measured by 10-Recall@-10 vs. Bits. (b) and (c) Distortion vs. Bits.

5. CONCLUSIONS

A universal compressor network is designed based on the Sparse Ternary Codes framework for similarity search where we demonstrate rate-distortion performance on synthetic as well as real data, superior to state-of-the-art methods from the binary hashing family. The intrinsic limitations of rate allocation w.r.t. optimality suggests a multi-layer design which is not applicable for binary encoding but applies very nicely to STC when they are set to be highly sparse. Thanks to the simplicity of encoding and the universality of the signals considered, these results can be useful for many applications. An immediate benefit would be for the idea of list-refinement in similarity search which we will address in a future work.

6. REFERENCES

[1] S. Ferdowsi, S. Voloshynovskiy, D. Kostadinov, and T. Holotyak, "Fast content identification in high-dimensional feature spaces using sparse ternary codes," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2016, pp. 1–6.

[2] Sohrab Ferdowsi, Sviatoslav Voloshynovskiy, Dimche Kostadinov, and Taras Holotyak, "Sparse ternary codes for similarity search have higher coding gain than dense binary codes," in *2017 IEEE International Symposium on Information Theory (ISIT) (ISIT'2017)*, Aachen, Germany, jun 2017.

[3] Behrooz Razeghi and Slava Voloshynovskiy, "Privacy-preserving outsourced media search using secure sparse ternary codes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, April 2018, pp. 1–5.

[4] Behrooz Razeghi, Slava Voloshynovskiy, Dimche Kostadinov, and Olga Taran, "Privacy preserving identification using sparse approximation with ambiguization," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Rennes, France, December 2017, pp. 1–6.

[5] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 1, pp. 117–128, 2011.

[6] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun, "Optimized product quantization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 4, pp. 744–755, April 2014.

[7] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, Dec 2013.

[8] Yan Xia, K. He, P. Kohli, and J. Sun, "Sparse projections for high-dimensional binary codes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3332–3339.

[9] T. Cover and J. Thomas, *Elements of Information Theory 2nd Edition*, Wiley-Interscience, 2 edition, 7 2006.

[10] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi, "Advances in residual vector quantization: a review," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 226–262, Feb 1996.

[11] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 957–971, Aug 1988.

[12] R. Venkataramanan, T. Sarkar, and S. Tatikonda, "Lossy compression via sparse linear regression: Computationally efficient encoding and decoding," *Information Theory, IEEE Transactions on*, vol. 60, no. 6, pp. 3265–3278, June 2014.

[13] M. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002.