MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Universal Embeddings For Kernel Machine Classification

Boufounos, P.T.; Mansour, H.

TR2015-070 May 2015

Abstract

Visual inference over a transmission channel is increasingly becoming an important problem in a variety of applications. In such applications, low latency and bit-rate consumption are often critical performance metrics, making data compression necessary. In this paper, we examine feature compression for support vector machine (SVM)-based inference using quantized randomized embeddings. We demonstrate that embedding the features is equivalent to using the SVM kernel trick with a mapping to a lower dimensional space. Furthermore, we show that universal embeddings-a recently proposed quantized embedding design-approximate a radial basis function (RBF) kernel, commonly used for kernel-based inference. Our experimental results demonstrate that quantized embeddings achieve 50% rate reduction, while maintaining the same inference performance. Moreover, universal embeddings achieve a further reduction in bit-rate over conventional quantized embedding methods, validating the theoretical predictions.

International Conference on Sampling Theory and Applications (SampTA)

Universal Embeddings For Kernel Machine Classification

Petros T. Boufounos and Hassan Mansour
Mitsubishi Electric Research Laboratories, Cambridge, MA 02139,
{*petrosb,mansour*}@merl.com

***Abstract*—Visual inference over a transmission channel is increasingly becoming an important problem in a variety of applications. In such applications, low latency and bit-rate consumption are often critical performance metrics, making data compression necessary. In this paper, we examine feature compression for support vector machine (SVM)-based inference using quantized randomized embeddings. We demonstrate that embedding the features is equivalent to using the SVM kernel trick with a mapping to a lower dimensional space. Furthermore, we show that universal embeddings—a recently proposed quantized embedding design—approximate a radial basis function (RBF) kernel, commonly used for kernel-based inference. Our experimental results demonstrate that quantized embeddings achieve 50% rate reduction, while maintaining the same inference performance. Moreover, universal embeddings achieve a further reduction in bit-rate over conventional quantized embedding methods, validating the theoretical predictions.**

I. Introduction

Visual inference applications are increasingly adopting a client/server model, in which inference is performed over a transmission channel by a remote server. For example, augmented reality, visual odometry, and scene understanding are some example applications which are often performed remotely, sometimes over the cloud. For the success of most of these applications, latency and bit-rate consumption are critical problems. Thus, efficient and low-complexity compression of the transmitted signals is essential for their operation.

Most visual inference systems operate by extracting visual features, such as the well-established SIFT, SURF, or HOG features [1]–[3], among many others. However, these features may sometimes consume more bandwidth than a compressed image, making them ill-suited for use over a transmission channel. Moreover, the complexity of image compression may introduce significant latency and complexity in the system.

Recently it was shown, in the context of Nearest Neighbor (NN)-based inference, that visual features can be compressed to a rate much lower than the underlying image using Locality-Sensitive-Hashing (LSH) based schemes—essentially randomized embeddings followed by 1-bit quantization [4], [5]. A more careful analysis of the properties of randomized embeddings, when combined with scalar quantization, demonstrated that carefully balancing the quantizer accuracy with the dimensionality of the random projections can further reduce the rate by more than 33% [6], [7]. A further 33% gain can be obtained by replacing the scalar quantizer with a universal scalar quantizer [8], [9]. The resulting universal embeddings only represent a range of signal distances and can be tuned to represent only the range of distances necessary for NN-based computation, at a significant gain in the bit-rate.

In this paper we examine quantized embeddings in the context of support vector machine (SVM)-based inferences. We demonstrate that using universal embeddings to encode features for an SVM classifier approximates a particular radial basis function (RBF) kernel which, in turn, is a good approximation for the commonly used and very successful Gaussian RBF kernel. In particular, the bit-rate determines the quality of the approximation. Our experiments using HOG features in an example multiclass image classification task demonstrate that randomized embeddings followed by appropriately designed scalar quantization significantly reduces the bit-rate required to code the features while maintaining high SVM-based inference accuracy. Furthermore, universal embeddings can further improve the classification accuracy while reducing the bit-rate.

The paper is organized as follows. In the next section, we present an overview of the quantized embeddings used in this paper as well as a brief summary of SVM-based classification. Section III discusses how embedding design affects their distance preserving performance, and highlights how randomized embeddings can be viewed as approximating RBF kernels in the context of kernel-based inference. Section IV presents our experimental investigation which validates expectations stemming from the theoretical discussion.

II. Background Overview

A. Support Vector Machines

Support vector machines (SVMs) are binary linear classifiers used in supervised learning problems that identify separating hyperplanes in a training data set. Given a training set $\mathcal{S} = \{(\mathbf{x}^{(i)}, z^{(i)},\ i = 1, \dots, m\}$ of data points $\mathbf{x}^{(i)} \in \mathbb{R}^N$ and binary labels $z^{(i)} \in \{-1, +1\}$, the SVM training problem can be cast as that of finding the hyperplane identified by $(\mathbf{w}, b)$ by solving

$$\min_{\mathbf{w}\in\mathbb{R}^N, b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t. } z^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1,\ i = 1, \dots, m. \tag{1}$$

Problem (1) is commonly reformulated and solved in its unconstrained form given by

$$\min_{\mathbf{w}\in\mathbb{R}^N, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{w}, b; \mathbf{x}^{(i)}, z^{(i)}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{2}$$

where $\ell(\mathbf{w}, b; \mathbf{x}^{(i)}, z^{(i)})$ is the hinge loss function

$$\ell(\mathbf{w}, b; \mathbf{x}^{(i)}, z^{(i)}) = \max\{0, 1 - z^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)} + b)\}, \quad (3)$$

and λ is a regularization parameter.

In some applications, it may be beneficial to find separating hyperplanes in a higher dimensional lifting space of the data. Let $\psi(\cdot)$ be a nonlinear lifting function from $\mathbb{R}^N$ to some higher dimensional space. Any positive semi definite function $K(\mathbf{x}, \mathbf{u})$ defines an inner product and a lifting $\psi(\cdot)$ so that the inner product between lifted datapoints can be quickly computed using $K(\mathbf{x}, \mathbf{u}) = \langle \psi(\mathbf{x}), \psi(\mathbf{u}) \rangle$. Since the SVM training algorithm can be written entirely in terms of inner products $\langle \mathbf{x}, \mathbf{u} \rangle$, we can replace all inner products with $K(\mathbf{x}, \mathbf{u})$ without ever lifting the data using $\psi(\cdot)$, a techniques known as the kernel trick.

In some cases, it is possible to compute or approximate certain kernels by explicitly mapping the data to a low-dimensional inner product space. For example, Rahimi and Recht [10] propose a randomized feature map $\phi(\cdot)$, that transforms the data into a low-dimensional Euclidean space. Using $\phi : \mathbb{R}^N \to \mathbb{R}^M,\ M \ll N$, as the feature map, the kernel $K(\mathbf{x}, \mathbf{u})$ can be computed in the lower-dimensional space as

$$K(\mathbf{x}, \mathbf{u}) = \phi(\mathbf{x})^T \phi(\mathbf{u}). \quad (4)$$

Such randomized feature maps have strong connections to the field of randomized embeddings, which we describe next.

B. Randomized Embeddings

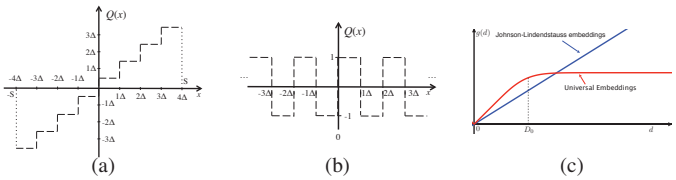
An embedding is a mapping of a set $\mathcal{S}$ to another set $\mathcal{V}$ that preserves some property of $\mathcal{S}$ in $\mathcal{V}$. Embeddings enable algorithms to operate on the embedded data, allowing processing and inference, so long as the processing relies on the preserved property.

In particular, Johnson-Lindenstrauss (JL) embeddings [11]—the most celebrated example—preserve the distances between pairs of signals. The JL lemma states that one can design an embedding $f(\cdot)$ such that for all pairs of signals $\mathbf{x}, \mathbf{x}' \in \mathcal{S} \subset \mathbb{R}^N$, their embedding, $\mathbf{y} = f(\mathbf{x})$ and $\mathbf{y}' = f(\mathbf{x}')$, with $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^M$ satisfies

$$(1 - \epsilon)\|\mathbf{x} - \mathbf{x}'\|_2^2 \leq \|\mathbf{y} - \mathbf{y}'\|_2^2 \leq (1 + \epsilon)\|\mathbf{x} - \mathbf{x}'\|_2^2 \quad (5)$$

for some ϵ, as long as $M = O\left(\frac{\log P}{\epsilon^2}\right)$, where P is the number of points in $\mathcal{S}$. Later work further showed that the JL map can be realized using a linear map $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$, where the matrix $\mathbf{A}$ can be generated using a variety of random constructions (e.g., [12], [13]).

The main feature of the JL lemma is that the embedding dimension M depends logarithmically only on the number of points in the set, and not on its ambient dimension N. Thus, the embedding dimension can typically be much lower than the ambient dimension, with minimal compromise on the embedding fidelity, as measured by ϵ. Any processing based on distances between signals—which includes the majority of inference methods—can operate on the much lower-dimensional space $\mathcal{V}$.



(a) (b) (c)

Fig. 1. (a) Conventional 3-bit (8 levels) scalar quantizer with saturation level $S = 4\Delta$. (b) Universal scalar quantizer. (c) The embedding map $g(d)$ for JL-based embeddings (blue) and for universal embeddings (red).

C. Quantized JL Embeddings

While dimensionality reduction through embedding can be very useful in reducing the complexity of processing or inference algorithms, in a number of applications the desirable goal is also to reduce the transmission rate before processing. In such applications, quantized embeddings have been shown to be highly successful at preserving Euclidean distances while significantly reducing the bit-rate requirements. Specifically, [6] considers a finite-rate uniform scalar quantizer $Q(\cdot)$, as shown in Fig. 1(a), with stepsize $\Delta = S2^{-B+1}$, where S is the saturation level of the quantizer, and B the number of bits per coefficient. Using such a quantizer, a JL map $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ can be quantized to $\mathbf{q} = Q(\mathbf{A}\mathbf{x})$ and satisfy

$$\begin{aligned}(1 - \epsilon)\|\mathbf{x} - \mathbf{x}'\|_2 - S2^{-B+1} \\ \leq \|\mathbf{q} - \mathbf{q}'\|_2 \leq \\ (1 + \epsilon)\|\mathbf{x} - \mathbf{x}'\|_2 + S2^{-B+1}, \end{aligned} \quad (6)$$

assuming the saturation level S is set such that saturation does not happen or is negligible. This quantized JL (QJL) embedding uses a total rate of $R = MB$ bits.

The design of QJL embeddings exhibits a trade-off between the number of bits B per coefficient and the embedding space dimension M, i.e., the number of coefficients. For a fixed rate R, a larger B and smaller M will increase the error due to the JL embedding, ϵ, while a larger M and smaller B will increase the error due to quantization. The design choice should balance the two errors. For example, the optimal B was experimentally determined to be 3 or 4 for NN-based inference examples in [6], [7]. This is not a universal optimum; the optimal B depends on the application.

D. Universal Embeddings

More recently, [8], [9] introduced an alternative approach using a non-monotonic quantizer combined with dither instead of a finite-range uniform one. This approach only preserves distances up to a radius, as determined by the embedding parameters.

Universal embeddings exhibit a different design trade-off. Given a fixed total rate, R, the quality of the embedding depends on the range of distances it is designed to preserve. At a fixed bit-rate, increasing the range of preserved distances also increases the ambiguity of how well the distance are preserved.

Specifically, universal embeddings use a map of the form

$$\mathbf{q} = Q(\mathbf{A}\mathbf{x} + \mathbf{w}), \quad (7)$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ is a matrix with entries drawn from an i.i.d. standard normal distribution, $Q(\cdot)$ is the quantizer, and $\mathbf{w} \in \mathbb{R}^M$ is a dither vector with entries drawn from a $[0, \Delta]$ uniform i.i.d. distribution. An important difference with conventional embeddings is that the quantizer $Q(\cdot)$ is not a conventional quantizer shown in Fig. 1(a). Instead, the non-monotonic 1-bit quantizer in Fig. 1(b) is used. This means that values that are very different could quantize to the same level. However, for local distances that lie within a small radius of each value, the quantizer behaves as a regular quantizer with dither and stepsize Δ. This behavior is highlighted in Fig. 1(c).

Universal embeddings have been shown to satisfy

$$g\left(\|\mathbf{x}-\mathbf{x}'\|_2\right) - \tau \leq d_H\left(f(\mathbf{x}), f(\mathbf{x}')\right) \leq g\left(\|\mathbf{x}-\mathbf{x}\|_2\right) + \tau, \tag{8}$$

where $d_H(\cdot,\cdot)$ is the Hamming distance of the embedded signals and $g(d)$ is the map

$$g(d) = \frac{1}{2} - \sum_{i=0}^{+\infty} \frac{e^{-\left(\frac{\pi(2i+1)d}{\sqrt{2}\Delta}\right)^2}}{\left(\pi(i+1/2)\right)^2}. \tag{9}$$

Similarly to JL embeddings, universal embeddings hold with overwhelming probability as long as $M = O\left(\frac{\log P}{\tau^2}\right)$, where, again, P is the number of points in $\mathcal{S}$.

Furthermore, the map $g(d)$ can be bounded as follows

$$g(d) \geq \frac{1}{2} - \frac{1}{2}e^{-\left(\frac{\pi d}{\sqrt{2}\Delta}\right)^2}, \tag{10}$$

$$g(d) \leq \frac{1}{2} - \frac{4}{\pi^2}e^{-\left(\frac{\pi d}{\sqrt{2}\Delta}\right)^2}, \tag{11}$$

$$g(d) \leq \sqrt{\frac{2}{\pi}}\frac{d}{\Delta}, \tag{12}$$

and is very well approximated using

$$g(d) \approx \begin{cases} \frac{d}{\Delta}\sqrt{\frac{2}{\pi}}, & \text{if } d \leq \frac{\Delta}{2}\sqrt{\frac{\pi}{2}} \\ 0.5 & \text{otherwise} \end{cases} \tag{13}$$

III. Quantized Embeddings For Kernel Machines

A. Embedding Ambiguity Analysis

Typical embedding guarantees, such as (5), (6), and (8), characterize the ambiguity of the embedded distance as a function of the original signal distance. A general embedding guarantee has the form

$$(1-\epsilon)g\left(d_{\mathcal{S}}(\mathbf{x},\mathbf{x}')\right) - \tau \leq d_{\mathcal{W}}\left(f(\mathbf{x}), f(\mathbf{x}')\right) \leq (1+\epsilon)g\left(d_{\mathcal{S}}(\mathbf{x},\mathbf{x}')\right) + \tau, \tag{14}$$

where $g : \mathbb{R} \to \mathbb{R}$ is an invertible function mapping distances in $\mathcal{S}$ to distances in $\mathcal{W}$ and ϵ and τ quantify, respectively, the multiplicative and the additive ambiguities of the map. For JL and QJL, that map is $g(d) = d$. In universal embeddings the map is given by (9).

However, in practical inference applications the inverse is desired. Processing computes distances in the embedding domain, assuming they are approximately equal with the corresponding signal distances in the signal space $\mathcal{S}$. The more ambiguous this correspondence is, the more the inference algorithm is affected. To expose the ambiguity in original space $\mathcal{S}$, we rearrange and approximate (14) for small ϵ, τ using

$$\widetilde{d}_{\mathcal{S}} - \frac{\tau + \epsilon d_{\mathcal{W}}\left(f(\mathbf{x}), f(\mathbf{x}')\right)}{g'\left(\widetilde{d}_{\mathcal{S}}\right)} \lesssim d_{\mathcal{S}}(\mathbf{x},\mathbf{x}') \lesssim \widetilde{d}_{\mathcal{S}} + \frac{\tau + \epsilon d_{\mathcal{W}}\left(f(\mathbf{x}), f(\mathbf{x}')\right)}{g'\left(\widetilde{d}_{\mathcal{S}}\right)}, \tag{15}$$

where $\widetilde{d}_{\mathcal{S}} = g^{-1}\left(d_{\mathcal{W}}\left(f(\mathbf{x}), f(\mathbf{x}')\right)\right)$ estimates the signal distance given the embedding distance. Thus, the additive and multiplicative ambiguities remain approximately additive and multiplicative and get scaled by the gradient of the map $g'(\cdot)$.

In JL and QJL embeddings, this gradient is constant throughout the map since the map is linear. In universal embeddings, however, the gradient is inversely proportional to Δ in the range of distances preserved, and approximately zero beyond that:

$$g'(d) \approx \begin{cases} \frac{1}{\Delta}\sqrt{\frac{2}{\pi}}, & \text{if } d \leq \frac{\Delta}{2}\sqrt{\frac{\pi}{2}} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Thus, universal embeddings have ambiguity proportional to Δ for a range of distances also proportional to Δ and approximately infinite ambiguity beyond that. Taking their ratio, one can easily derive the following remark:

Remark In universal embeddings, the embedding ambiguity over the preserved distances is approximately equal to 2τ times the range of preserved distances.

For the majority of inference applications, only local distances need to be preserved by the embedding. For example, NN methods only require that the radius of distances preserved is such that the nearest neighbors can be determined. For SVM-based inference, this can be formalized using the machinery of kernel-based SVMs.

B. Quantized Embeddings Imply Radial Basis Function Kernels

Radial basis function (RBF) kernels, also known as shift invariant kernels, for SVMs have been very successful in a number of applications, as they regularize the learning to improve inference [14]. Their defining property is that the kernel function $K(\mathbf{x},\mathbf{x}')$ is only a function of the distance of the two points, i.e., $K(\mathbf{x},\mathbf{x}') = \kappa(\|\mathbf{x}-\mathbf{x}'\|_2)$.

While [10] demonstrates that randomized feature maps can approximate certain radial basis kernels, the constructed maps are not quantized, and, therefore, not very useful for transmission. Universal embeddings, however, also approximate a shift-invariant kernel. This kernel further approximates the commonly used Gaussian radial basis kernel.

Proposition 3.1: Let $\phi(\mathbf{x}) : \mathbb{R}^N \to \{-1, 1\}^M$ be a mapping function defined as $\phi(\mathbf{x}) = Q(\mathbf{A}\mathbf{x} + \mathbf{e})$, with $\mathbf{q} = \phi(\mathbf{x})$. The kernel function $K(\mathbf{x}, \mathbf{x}')$ given by

$$K(\mathbf{x}, \mathbf{x}') = \frac{1}{2M}\mathbf{q}^T\mathbf{q}' \tag{17}$$

is shift invariant and approximates the radial basis function

$$K(\mathbf{x}, \mathbf{x}') \approx \frac{1}{2} - g\left(\|\mathbf{x} - \mathbf{x}'\|_2\right), \tag{18}$$

with $g(d)$, as defined in (9). Furthermore, this RBF approximates the Gaussian RBF.

Proof By expressing the Hamming distance in $\{+1, -1\}^M$ as a Euclidian distance in $\mathbb{R}^M$, i.e., using $d_H\left(\phi(\mathbf{x}), \phi(\mathbf{x}')\right) = \frac{1}{4M}\|\mathbf{q} - \mathbf{q}'\|_2^2$, we obtain

$$\begin{aligned}\frac{1}{4M}\|\mathbf{q} - \mathbf{q}'\|_2^2 &= \frac{1}{4M}\|\mathbf{q}\|_2^2 + \frac{1}{4M}\|\mathbf{q}'\|_2^2 - \frac{1}{2M}\mathbf{q}^T\mathbf{q}' \\ &= \frac{1}{4} + \frac{1}{4} - \frac{1}{2M}\mathbf{q}^T\mathbf{q}' \qquad (19)\\ \Rightarrow K(\mathbf{x}, \mathbf{x}') &= \frac{1}{2M}\mathbf{q}^T\mathbf{q}' = \frac{1}{2} - d_H\left(\phi(\mathbf{x}), \phi(\mathbf{x}')\right). \qquad (20)\end{aligned}$$

Exploiting the bounds in (8), we can approximate the kernel $K(\mathbf{x}, \mathbf{x}')$ using

$$\begin{aligned}&\frac{1}{2} - g(d) + \tau \\ &\qquad \le K(\mathbf{x}, \mathbf{x}') = \frac{1}{2} - d_H\left(\phi(\mathbf{x}), \phi(\mathbf{x}')\right) \le \\ &\qquad\qquad \frac{1}{2} - g(d) + \tau, \qquad (21)\end{aligned}$$

where $d = \|\mathbf{x} - \mathbf{x}'\|_2$ is the distance between the two signals. Thus, the kernel approximates the RBF kernel $K(\mathbf{x}, \mathbf{x}') = \frac{1}{2} - g(\|\mathbf{x} - \mathbf{x}'\|)$ within τ. Furthermore, using (9), (11) and (10), we can further approximate the kernel $K(\mathbf{x}, \mathbf{x}')$ as

$$\frac{1}{2}e^{-\left(\frac{\pi d}{\sqrt{2}\Delta}\right)^2} - \tau \le K(\mathbf{x}, \mathbf{x}') \le \frac{4}{\pi^2}e^{-\left(\frac{\pi d}{\sqrt{2}\Delta}\right)^2} + \tau. \tag{22}$$

In other words, the resulting kernel is an approximation of the Gaussian RBF kernel, $K_G(\mathbf{x}, \mathbf{x}') = ce^{-\left(\frac{\|\mathbf{x}-\mathbf{x}'\|}{\sqrt{2}\sigma}\right)^2}$, with $\sigma = \frac{\Delta}{\pi}$. Note that in practice the constant scaling c does not matter in the kernel computation and is typically set to 1. ∎

Note that the approximation is not very accurate near the origin, where $d \approx 0$, but $d \neq 0$. In that region, the Gaussian RBF is flatter, while our kernel is steeper. An appropriate approximation there is the first order polynomial RBF kernel $K(\mathbf{x}, \mathbf{x}') = c\|\mathbf{x} - \mathbf{x}'\|_2$. However, the ambiguity due to τ will dominate that effect in practice.

Of course, QJL are approximations of the standard inner product kernel $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$. The accuracy of the approximation depends on the rate used for the embedding and the allocation of the rate between projection dimensionality M and bits per dimension B, as described in Sec. II-C.

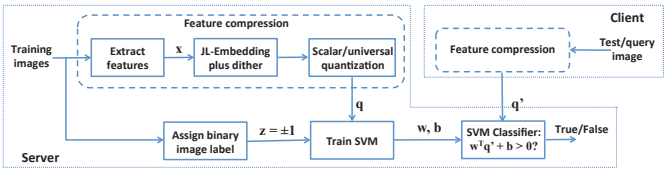


Fig. 2. Block diagram illustrating the feature compression, classifier training, and object detection stages of a binary classification task. Every training image is assigned a binary label $z \in \{-1, +1\}$ indicating whether or not it corresponds to the target object class.

C. Feature Compression For Classification

Quantized and universal embeddings are quite useful for visual inference over a network. In this section, we put everything together and formulate the object classification task using binary linear SVMs. These SVMs are trained on image feature vectors, or descriptors, that are compressed using quantized embeddings.

Fig. 2 shows a block diagram of our classification framework for a single object class $\mathcal{C}$. Given a database of training images at the server, indexed by $i \in \{1 \ldots, m\}$ and corresponding labels $z^{(i)} \in \{-1, +1\}$, we first extract feature vectors $\mathbf{x}^{(i)} \in \mathbb{R}^N$ from every image. We then generate quantized or universal embeddings $\mathbf{q}^{(i)}$ of the feature vectors, as described in Sections II-C and II-D.

The quantized embeddings of image features $\mathbf{q}^{(i)}$ are then used to train an SVM classifier to find a separating hyperplane, identified by the vector $\mathbf{w}$ and bias term b, by solving (2). The separating hyperplane divides the embedding space into points that generate positive versus negative labels.

When a visual query is executed, the client computes quantized embeddings $\mathbf{q}'$ from features extracted from the query image, and transmits the quantized embeddings over a channel to the server. Classification is performed at the server according to the sign of $\mathbf{w}^T\mathbf{q}' + b$, such that,

$$\text{if } \mathbf{w}^T\mathbf{q}' + b > 0, \text{then query image} \in \mathcal{C}. \tag{23}$$

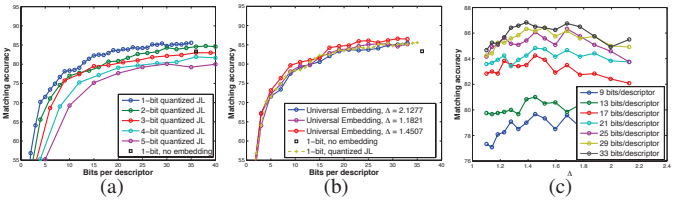
The same framework can be extended to multiclass classification by computing a new separating hyperplane identified by $(\mathbf{w}^{(j)}, b^{(j)})$ for each class $\mathcal{C}_j$ for $j \in \{1, \ldots, J\}$. However, classification is now performed by choosing the class that induces the largest positive margin to the query point, i.e.

$$\text{query image} \in \mathcal{C}_{j^*}, \text{ where } j^* = \arg\max_j \mathbf{w}^{(j)T}\mathbf{q}' + b^{(j)}. \tag{24}$$

IV. Experimental results

We test the performance of our compressed feature representation on a multiclass classification problem. The goal is to identify the class membership of query images belonging to one of 8 different classes.

To set up this problem, we extract Dalal-Triggs Histogram of Oriented Gradients (HOG) features [3] from 15 training and 15 test images. The HOG algorithm extracts a 36 element feature vector (descriptor) for every 8×8 pixel block in an image. The descriptors encode local 1-D histograms of gradient



(a) (b) (c)

Fig. 3. Classification accuracy as a function of the bit-rate achieved using (a) quantized JL (QJL) embeddings; and (b) universal embeddings. (c) Classification accuracy as a function of the quantization step size Δ used in computing the universal embeddings.

directions in small spatial regions in an image. Every HOG feature is compressed using either quantized JL embeddings or universal quantized embeddings. The compressed features are then stacked to produce a single compressed feature vector for each image. Next, the compressed features of the training images are used to train a binary linear SVM classifier. In the testing stage, compressed HOG features of the test/query images are computed and classification is performed using the trained SVM classifier. In our simulations, we used tools from the VLFeat library [15] to extract HOG features and train the SVM classifier.

We consider eight image classes. One is the persons from the INRIA person dataset [3], [16]. The other seven—car, wheelchair, stop sign, ball, tree, motorcycle, and face—extracted from the Caltech 101 dataset [17], [18]. All images are standardized to 128×128 pixels centered around the target object in each class.

Fig. 3(a) shows the classification accuracy obtained by quantized JL embeddings of HOG descriptors using the trained SVM classifier. The black square corresponds to 1-bit scalar quantization of raw non-embedded HOG descriptors, using a bit-rate of 36 bits—one bit for each element of the descriptor.

The figure shows that 1-bit quantized JL embeddings allow us to achieve a 50% bit-rate reduction, compared to non-embedded quantized descriptors, without reduction in performance (classification accuracy). This is obtained using an 18-dimensional JL embedding of every HOG descriptor, followed by 1-bit scalar quantization. Furthermore, increasing the embedding dimension, and, therefore, the bit-rate, above 18 improves the inference performance beyond that of the 1-bit quantized non-embedded HOG features. Note that, among all quantized JL embeddings, 1-bit quantization achieves the best rate-inference performance.

Fig. 3(b) compares the classification accuracy of universal embeddings for varying values of the step size parameter Δ with that of the 1-bit quantized JL embeddings and the 1-bit quantized non-embedded HOG descriptors. With the choice of $\Delta = 1.4507$, the universal embedded descriptors further improve the rate-inference performance over the quantized JL embeddings. In particular, they also achieve the same classification accuracy as any choice of quantization for non-embedded HOG descriptors, or, even, unquantized ones, at significantly lower bit-rate—points not shown in the figure, as they are out of the interesting part of the bit-rate scale.

Figure 3(c) illustrates the effect of the parameter Δ by plotting the classification accuracy as a function of Δ for different embedding rates. The figure shows that, similar to the findings in [9], if Δ is too small or too large, the performance suffers.

As evident, an embedding-based system design can be tuned to operate at any point on the rate vs. classification performance frontier, not possible just by quantizing the raw HOG features. Furthermore, with the appropriate choice of Δ, universal embeddings improve the classification accuracy given the fixed bit-rate, compared with quantized JL embeddings, or reduce the bit-rate required to deliver a certain inference performance.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, Jun. 2008.

[3] D. Navneet and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, June 2005, pp. 886–893.

[4] K. Min, L. Yang, J. Wright, L. Wu, X.-S. Hua, and Y. Ma, "Compact projection: Simple and efficient near neighbor search with practical memory requirements," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, June 13–18 2010.

[5] C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient visual correspondences using random projections," in *Proc. IEEE International Conference on Image Processing (ICIP)*, San Diego, CA, October 12-15 2008.

[6] M. Li, S. Rane, and P. T. Boufounos, "Quantized embeddings of scale-invariant image features for mobile augmented reality," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Banff, Canada, Sept. 17–19 2012.

[7] S. Rane, P. T. Boufounos, and A. Vetro, "Quantized embeddings: An efficient and universal nearest neighbor method for cloud-based image retrieval," in *Proc. SPIE Applications of Digital Image Processing XXXVI*, San Diego, CA, August 25-29 2013.

[8] P. T. Boufounos, "Universal rate-efficient scalar quantization," *IEEE Trans. Info. Theory*, vol. 58, no. 3, pp. 1861–1872, March 2012.

[9] P. T. Boufounos and S. Rane, "Efficient coding of signal distances using universal quantized embeddings," in *Proc. Data Compression Conference (DCC)*, Snowbird, UT, March 20-22 2013.

[10] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," 2007.

[11] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189 – 206, 1984.

[12] D. Achlioptas, "Database-friendly Random Projections: Johnson-lindenstrauss With Binary Coins," *Journal of Computer and System Sciences*, vol. 66, pp. 671–687, 2003.

[13] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[14] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural networks*, vol. 11, no. 4, pp. 637–649, 1998.

[15] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[16] "INRIA Person Dataset," http://pascal.inrialpes.fr/data/human/.

[17] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR), Workshop on Generative-Model Based Vision.*, June 2004, pp. 178–178.

[18] "Caltech 101 dataset," http://www.vision.caltech.edu/Image_Datasets/Caltech101/.