Differential Geometric Retrieval of Deep Features

Y Qian and E Vazquez
Cortexica Vision Systems Limited
30 Stamford Street SE1 9LQ
London, UK
yu.qian@cortexica.com

B Sengupta
Cortexica Vision Systems Limited
Imperial College London
London, UK
b.sengupta@imperial.ac.uk

Abstract—**Comparing images to recommend items from an image-inventory is a subject of continued interest. Added with the scalability of deep-learning architectures the once 'manual' job of hand-crafting features have been largely alleviated, and images can be compared according to features generated from a deep convolutional neural network. In this paper, we compare distance metrics (and divergences) to rank features generated from a neural network, for content-based image retrieval. Specifically, after modelling individual images using approximations of mixture models or sparse covariance estimators, we resort to their information-theoretic and Riemann geometric comparisons. We show that using approximations of mixture models enable us to compute a distance measure based on the Wasserstein metric that requires less effort than other computationally intensive optimal transport plans; finally, an affine invariant metric is used to compare the optimal transport metric to its Riemann geometric counterpart – we conclude that although expensive, retrieval metric based on Wasserstein geometry is more suitable than information theoretic comparison of images. In short, we combine GPU scalability in learning deep feature vectors with statistically efficient metrics that we foresee being utilised in a commercial setting.**

1. Introduction

A common problem in computer vision lies in finding similarity between 2 (or 3)-dimensional images (or tensors). This is attained by measuring distances between the two objects, primarily using normalised co-relation, Euclidean distance, Bhattacharyya distance, Jensen-Shannon divergence, amongst many others. The distances are measured after the images are encoded in some latent space wherein such a latent structure is learnt using a variety of classifiers – support vector machines (SVMs), logistic regression, etc. Recently, due to the advantages of scalability, large-scale classifier frameworks based on deep-learning have been used for music recommendation [29], image recommendation [25] as well as general recommendation architectures. [5]. Most of these frameworks do not take the underlying

---

YQ and BS contributed equally to this manuscript

geometry of the feature space into account while making a recommendation. This becomes increasingly important when the similarity between objects is measured in terms of 'perception', a quantity that is oblivious to the commonly used distance metrics. The non-trivial problem lies in (a) collecting perceptual similarity between objects in a database (via psychophysics), and (b) using this similarity to construct a metric for classification and retrieval (metric learning, learning to rank, etc.). To compound the problem further, metric used for comparing images might be very different than those for comparing sounds. Regardless of the modality, a large stream of work in neuroscience hypothesise that perception is based on minimising the prediction error between what is observed and what we predict we will observe [11].

In this paper, we start with the 0th order problem, i.e., compare how different distance metrics fair against one another when objects that are to be compared are represented as feature spaces induced by a deep neural network; furthermore, we use approximations of the probability density to compute a metric based on the principle of optimal transport [30] and Riemann geometry [1] that takes into account the geometry of transport between two images, an idea that is inherently essential for solving the perceptual similarity problem.

Technically, the problem lies in searching an image database (ranking) with millions of image features ($A_{n_i \times r_m \times f_j}$) for sets of images (say up to 10-20 images) that have similar properties to a query image ($\hat{a}_{query}$). Here, A is a $n_i \times r_m \times f_j$ tensor where n_i is the total number of images, r_m is the length of each feature vector and f_j is the total number of features extracted from one image; $\hat{a}_{query}$ is a $r_m \times f_j$ query matrix. Although, one can manually construct feature-vectors based on wavelet decomposition, low-rank approximations, etc., we rely on using a convolution neural network (CNN) to compute the feature signature ($r_m \times f_j$) for images in the database, as well as the query.

One way to operationalise a solution lies in weighting each image in the database using a weight vector, and subsequently, extremise the mutual information (or another comparison metric) between the query and the database with respect to the weights. The result of this optimisation

problem leads us to a weight vector that provides a rank for all the images in the database when compared to the query image. This is equivalent to measuring distances where each image lies on a continuous probability manifold. There are two contributions of this paper – (a) in order to describe each image with its deep feature, we use either a computationally efficient approximation of Gaussian Mixture Models (GMMs) or a sparse covariance estimator based on Given rotations, and (b) we provide comparison between these probability distributions using a variety of information-theoretic and geometric metrics. This work leads us to a much deeper problem where geometric similarity measures can be possibly combined to approximate the metric governing **perceptual similarity**.

2. Methods

2.1. Dataset and deep-feature generation

In this paper, Describable Textures Dataset (DTD) [6] is used to evaluate geometric similarity measures for image retrieval. Images in DTD are collected from wild images (Google and Flickr) and classified based on human visual perception [28], such as directionality (line-like), regularity (polka-dotted and chequered), etc. DTD is therefore selected in this research to evaluate the similarity measurements base on human visual perception. DTD contains 5640 wild texture images with 47 describable attributes drawn from the psychological literature and is publicly available on the web at http://www.robots.ox.ac.uk/vgg/data/dtd/.

Textures can be described via orderless pooling of filter bank response [12]. In Deep CNN, the convolutional layers are akin to non-linear filter banks; these have in fact been proved to be better for texture descriptions [7]. Here, the deep local features are extracted from last convolutional layer of a pre-trained VGG-M [4]. This is represented by $A = (a_1, ..., a_i, ..., a_N : a \in \mathbb{R}^D)$; the size of last convolutional layer is $H \times W \times D$, where D denotes the dimension vector of filter response at the ith pixel of last convolution layer; $N = H \times W$ is the total number of local features.

For image level representation, two methods are applied to local features – one is to generate a Gaussian Mixture Model (GMM) model on local descriptors and the second is to estimate a shrunk yet sparse co-variance matrix from the deep feature representation of individual images. A statistical similarity metric is then applied to rank images. As a baseline for distance calculations, we compute Euclidean distances ($\|x\|_2 = \sqrt{x_1^2 + \ldots + x_n^2}$) between the query image and the database. Similarly, to establish a baseline for feature extraction, we use the Bag of Words (BoW) composed of scale-invariant feature transform (SIFT) features. For further details on SIFT and BoW, please refer to [25].

2.2. Retrieval and Ranking

Image retrieval using Euclidean norm with bag-of-words feature encoding has been described elsewhere [25]. In subsection 2.2.1-2.2.3, we describe three approaches to rank images in terms of their 'statistical similarity' (not perceptual similarity). For the first, we use an information-theoretic divergence while the second and third distances are based on the cost involved in transporting one image to another, and geodesic distance on a Riemannian manifold, respectively.

To rank images in the database we use two methods, one is to build a Gaussian Mixture Model (GMM) [20], and the second is to estimate a covariance matrix from deep features. For each image, we model the $r_m \times f_j$ feature matrix using a GMM. Specifically, for computational and analytical efficiency (baseline measure), we approximate the GMM with a Normal distribution, such that the sufficient statistics read,

$$
\begin{aligned}
\tilde{\mu} &= \sum_a \omega_a \mu_a \\
\tilde{\Sigma} &= \sum_a \omega_a \left(\Sigma_a + (\mu_a - \tilde{\mu})(\mu_a - \tilde{\mu})^T \right) \quad (1)
\end{aligned}
$$

μ_a, Σ_a and ω_a are the mean, co-variance and the mixing weights of each Normal distribution (subscript a).

The second approximation to an image relies on estimating the co-variance matrix from the feature matrix generated from a deep convolutional neural network. Although the geometry of the co-variance matrix can be utilised to estimate it using low-rank and sparse penalisation, for the sake of computational efficiency, we use an alternative treatment due to [2], [3], i.e., a fast sparse matrix transformation (SMT). Briefly, the SMT imposes sparsity constraint on the manifold of co-variance matrices yet maintains a full-rank representation. This is useful as the computation is $\mathcal{O}(f_j)$; the SMT can also be seen as a generalisation of FFT and orthonormal para-unitary wavelet transform.

We will assume that each feature vector is *i.i.d* zero mean Normal random vectors, and the sample covariance is simply, $\frac{1}{n} A A^T$; it is a unbiased estimate of the true covariance matrix, $R = \mathbb{E}[S] = E \Lambda E^T$. Often time S is singular, and shrinkage estimators [15] are used to regularise the covariance matrix by shrinking it towards a target structure such as an identity matrix, a diagonal matrix with sample variances, amongst others. Sparsity can also be imposed, as in Graphical Lasso [10] by imposing a 1-norm constraint on the precision matrix. The maximum likelihood (ML) estimate of the eigenvectors (E) and the eigenvalues (Λ) give us,

$$
\begin{aligned}
\hat{E} &= \operatorname*{arg\,min}_{E \in \Omega_k} \left\{ \left| diag\left(E^T S E \right) \right| \right\} \\
\hat{\Lambda} &= diag\left(\hat{E}^T S \hat{E} \right) \quad (2)
\end{aligned}
$$

The SMT constrains the feasible set of Ω_k to a set of orthonormal transformations that are selected as an SMT of order K. A matrix E is an SMT of order K if it can be factorised to K sparse orthonormal matrices, i.e.,

$$
\begin{aligned}
E &= \prod_{k=1}^{K} E_k = E_1 E_1 \dots E_k \\
E_k &= I + \Theta(i_k, i_j, \theta_k) \quad (3)
\end{aligned}
$$

Each sparse matrix E can be constructed as a orthonormal Givens rotation on a pair of co-ordinate indexes (i_k, i_j) of Givens rotations such that,

$$
[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1, & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k), & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k), & \text{if } i = j_k \text{ and } j = i_k \\ 0, & \text{otherwise} \end{cases} \quad (4)
$$

Using greedy minimization [2], [3] we have,

$$
\begin{aligned}
\hat{E}_k &= \arg\min \left| diag\left(E_k^T S_k E_k\right) \right| \\
S_{k+1} &= \hat{E}_k^T S_k \hat{E}_k
\end{aligned}
$$

$$
\begin{aligned}
\hat{E} &= \prod_{k=1}^{K} \hat{E}_k \\
\hat{\Lambda} &= diag\left(S_{k+1}\right) \quad (5)
\end{aligned}
$$

As a final step, we obtain a shrunk co-variance matrix where the shrinkage parameter α is selected using cross-validation,

$$
\begin{aligned}
\Sigma_{SMT} &= \hat{E}\hat{\Lambda}\hat{E}^T \\
\Sigma &= \alpha \cdot \Sigma_{SMT} + (1 - \alpha) \cdot S \quad (6)
\end{aligned}
$$

2.2.1. Ranking by KL-divergence. Since there is no analytical solution for the KL-divergence between two GMMs $(V_a \sim \mathcal{N}_a(\omega_a, \mu_a, \Sigma_a))$ and $(V_b \sim \mathcal{N}_b(\omega_b, \mu_b, \Sigma_b))$, we utilize two approximations: in the first we approximate the GMM with Eqn. 1. The KL-divergence $(D_{KL}(V_a \| V_b))$ now reads,

$$
\frac{1}{2}\left[\log\frac{|\Sigma_b|}{|\Sigma_a|} - N_d + tr\left(\Sigma_b^{-1}\Sigma_a\right) + (\mu_b - \mu_a)^T \Sigma_b^{-1}(\mu_b - \mu_a)\right] \quad (7)
$$

In our experiments, we compute a symmetric-KL divergence which is simply $D_{KL}^{Normal} = \frac{1}{2}D_{KL}(V_a \| V_b) + \frac{1}{2}D_{KL}(V_b \| V_a)$. Sorting the KL-divergence provides us with a similarity rank.

This is a gross-approximation wherein a more subtle approximation relies in bounding the KL-divergence. Particularly, using results from information theory [14], [21], we provide retrieval results using a variational approximation to the KL divergence. Particularly, since the log-function is concave, using Jensen's inequality we have,

$$
\begin{aligned}
D_{KL}(V_a \| V_b) &= \mathbb{E}_{V_a}[V_a] - \mathbb{E}_{V_a}[V_b] \\
\mathbb{E}_{V_a}[V_b] &= V_a \log V_b \\
&= \sum_a \omega_a \int V_a \log \sum_b \phi_{b|a} \frac{\omega_b V_b}{\phi_{b|a}} \\
&\geqslant \sum_a \omega_a \int V_a \sum_b \phi_{b|a} \log \frac{\omega_b V_b}{\phi_{b|a}} \\
&= \sum_a \omega_a \sum_b \phi_{b|a} \left(\log\left(\frac{\omega_b}{\phi_{b|a}}\right) + \int V_a \log V_b \right)
\end{aligned}
$$
(8)

Here, $\phi_{b|a}$ is a variational parameter that is positive and sums to one. Maximizing *w.r.t* $\phi_{b|a}$ yields,

$$
\mathbb{E}_{V_a}[V_b] \geqslant \sum_a \omega_a \log \sum_b \omega_a e^{-D_{KL}(V_a\|V_b)} - \sum_a \omega_a \mathcal{H}(V_a) \quad (9)
$$

$\mathcal{H}$ is the entropy functional. Subsequently, the variational bound becomes,

$$
D_{KL}^{\text{variational}}(V_a \| V_b) = \sum_a \omega_a \log \frac{\sum_{a'} \omega_{a'} e^{-D_{KL}(V_a\|V_{a'})}}{\sum_b \omega_b e^{-D_{KL}(V_a\|V_b)}} \quad (10)
$$

We symmetrize the variational KL by using $D_{KL}^{\text{var}} = \frac{1}{2}D_{KL}^{\text{var}\,iational}(V_a\|V_b) + \frac{1}{2}D_{KL}^{\text{var}\,iational}(V_b\|V_a)$. Note that such a divergence is the difference of two variational approximations, not a bound in itself.

2.2.2. Ranking *via* Kantorovich relaxation. Let (Ψ, ψ_m) and (Λ, λ_m) denote two Polish probability spaces depicting *image 1* and *image 2*, respectively – ψ_m and λ_m. The trivial coupling between the two exists if Ψ and Λ are independent so that the coupling is simply a tensor product $\psi_m \otimes \lambda_m$. A more useful coupling exists when there is a function $S : \Psi \to \Lambda$ such that $\lambda = S(\psi)$. The transport map S is equivalently the change of variables from ψ_m to λ_m.

Definition of a transport map: Let S be a Borel map: $\Psi \to \Lambda$, the push forward of ψ_m through S is the Borel measure, denoted $S_{\#\psi_m}$ defined on Λ by $S_{\#\psi_m}(\Lambda) = \psi_m(S^{-1}(\Lambda))$. A Borel map: $\Psi \to \Lambda$ is said to be a transport map if $S_{\#\psi_m} = \lambda_m$.

In optimal transport [30], there is a cost entailed by transporting one measure into another. The transport map then relies on finding the infimum of $\left(\int_\Psi c(x, S(x))\, d\psi_m : S_{\#\psi_m} = \lambda_m\right)$. Optimal transference plans are important because such couplings are stable to perturbations, they encode geometric information about the underlying cost-function, and they exist in smooth as well as non-smooth settings. Given that the existence of this transport map can not be guaranteed, a Kantorovich relaxation amounts to a convex relaxation of Monge's formulation wherein we seek a coupling $\gamma \in P(\Psi, \Lambda)$,

$$
\gamma_0 = \arg\min_{\gamma \in P(\Psi, \Lambda)} \int_{\Psi \times \Lambda} c\left(x^\psi, x^\lambda\right) d\gamma \quad (11)
$$

The joint probability measure with the marginals ψ_m and λ_m allow us to define a Wasserstein distance of order p between ψ_m and λ_m,

$$\mathcal{W}(\psi_m, \lambda_m) = \inf \left(\left\{ \mathbb{E} \, D(x^\psi, x^\lambda)^p \right\}^{1/p} \right) \quad (12)$$

D is a distance with a corresponding cost of $c(x^\psi, x^\lambda) = d(x^\psi, x^\lambda)^p$. This Earth-Mover or the Monge-Kantorovich distance provides us with a metric over the space of squared integrable probability measures. For two Normal distribution, the L_2-Wasserstein distance [27], [30] reads,

$$D_\mathcal{W} = \|\mu_a - \mu_b\|_2^2 + tr\left(\Sigma_a + \Sigma_b - 2\left(\Sigma_a^{1/2} \Sigma_b \Sigma_a^{1/2} \right)^{1/2} \right) \quad (13)$$

Once the GMMs have been approximated via Eqn. 1 or Eqn. 6, it is fairly simple to compute distances using Eqn. 13.

2.2.3. Ranking *via* Affine Invariant Riemannian Metric.
Let us again consider two feature matrices (query and database), $V_a \sim \mathcal{N}(0, \Sigma_a)$ and $V_b \sim \mathcal{N}(0, \Sigma_b)$. These positive-definite matrices are elements of $S_{++}^{f \times f}$, a space with a defined Riemannian metric [9], [19]. Under such a geometry, the distance $D_R(V_a, V_b)$ between these two matrices is,

$$D_R(\Sigma_a, \Sigma_b) = \left\| \log(\Sigma_a^{-1/2} \Sigma_b \Sigma_a^{-1/2}) \right\|_F = \left[\sum_{c=1}^{C} \log^2 \lambda_c \right]^{1/2} \quad (14)$$

C is the dimension of the co-variance matrix, λ_c are the eigenvalues, and F represents the Frobenius norm. A useful property of such a distance is that regardless of how the images are manipulated – be it re-scaling, normalisation, whitening, filtering, etc. – the distance between the two sources as captured by Eqn. 14 remains invariant.

To compute Eqn. 14 one can use Eqn. 1 to approximate both GMMs as Normal distributions; alternatively, the covariance estimated using Eqn. 6 can be used.

3. Experiments

In this section, deep feature geometric retrieval methods are evaluated on the DTD dataset. For each image, a set of deep local features is extracted from last convolutional layer of a pre-trained VGG-M. The dimension of each local feature vector is 512. A GMM with 64 components is subsequently generated from the set of local deep features. Normal approximation by GMM and sparse covariance estimation by SMT are used to represent the feature matrix; information-theoretic (Normal and Variational approximation KL) and geometric (Wasserstein and Riemannian) measures to gauge the similarity of two images. In this experiment, Normal approximation KL, Variational approximation KL and Wasserstein metric is applied on GMM model respectively and represented by GMM-Normal KL, GMM-Variational KL and GMM-Wasserstein. Normal approximation KL, Wasserstein and Riemannian metric are applied on sparse covariance generated by SMT respectively

MAP	Top-1	Top-5	Top-10	Time
GMM-Normal KL	0.53	0.46	0.42	0.375s
GMM-Variational KL	0.45	0.42	0.38	0.016s
GMM-Wasserstein	0.62	0.52	0.46	5.147s
SMT-Riemannian	0.50	0.44	0.39	0.754s
SMT-Normal KL	0.53	0.44	0.39	0.125s
SMT-Wasserstein	0.59	0.51	0.46	9.04s
SIFT-BoW-Euclidean	0.43	0.37	0.32	0.0007s

TABLE 1: Retrieval results on the DTD dataset. Note that VGG-M has been pre-trained on Imagenet.
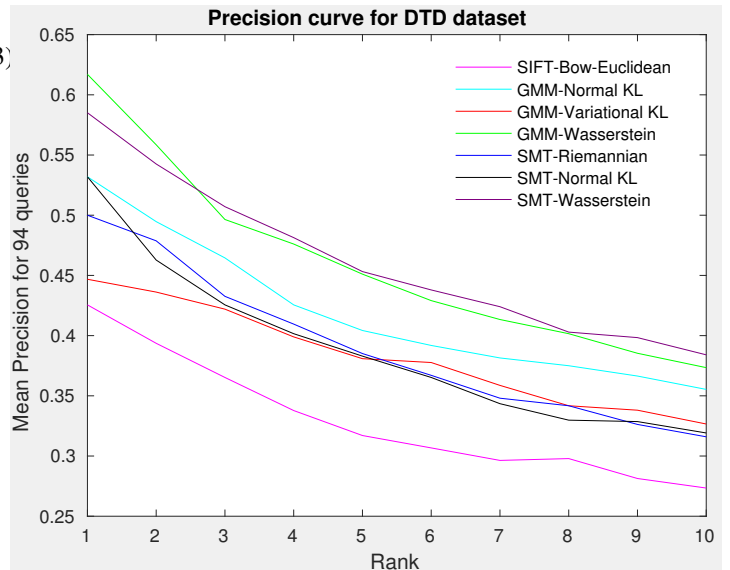

Figure 1: Precision on the DTD dataset

and denoted by SMT-Normal KL, SMT-Wasserstein and SMT-Riemannian.

To evaluate similarity metric for image retrieval, Mean Average Precision (MAP) on top 10 rankings are calculated. 2 images per category, i.e., a total of 94 images are selected as queries from the test dataset. The dataset retrieved includes 3760 images from DTD training and validation datasets. MAP on DTD is listed in Table 1.

Figure 2: Retrieved results on DTD: a) GMM-Wasserstein b) SMT-Wasserstein c) GMM-Normal KL d) SMT-Normal KL e) SMT-Riemannian f) GMM-Variational KL g) SIFT-BoW Euclidean

Average precision on top 10 ranking is displayed in Figure 1. An example of the retrieval obtained with each method is shown in Figure 2. On each case, ten images are displayed. Top left image is the query used. The rest of images are arranged by similarity to query image as obtained with each method.

As is apparent from Table 1, computing Wasserstein distances (using GMM or shrank co-variances) prove to be superior when compared to the other methods evaluated, including the baseline that uses a bag of words based SIFT feature generation and Euclidean distances. Surprisingly, variational KL is the least precise distance metric even in contrast to the case where we approximate the feature matrix with a univariate Normal distribution. A possible cause for such a lower precision might be due to the inferior conditioning of the co-variance matrix.

Query times are shown in Table 1; they have been obtained as the average time to calculate the similarity between two images. The code was implemented in Matlab 2015a under Linux with Intel(R) Xeon(R) CPU E5-2640 @ 2.00GHz and 125G RAM.

4. Conclusion

Using hand-crafted features such as scale-invariant feature transform (SIFT), a histogram of oriented gradients (HoG), etc. with Euclidean distances has had a long standing history in computer vision, especially before the advent of deep-learning based feature extractors. Hand-crafted features have poor generalisation capabilities along with being non-robust to non-linear transformations. The same goes for Euclidean distance, which is often not the ambient geometry for the objects being compared. For probability measures, the notion of an ambient geometry is clear due to the Riemann manifold inherited by these measures. In this paper, we have touched upon the 0th order problem that may lead to understanding 'perceptual similarity'. More specifically, we have used a convolution neural network (CNN) to obtain feature matrices; utilising either Gaussian Mixture Models (GMMs) or shrunk covariance estimators to obtain a probabilistic representation of the features. Subsequently, using information theoretic divergences and Riemann geometric metrics, we compare (dis) similarities between images.

Based on evaluation for DTD dataset, Wasserstein distances show increased retrieval fidelity based on both probabilistic representations, albeit they are more expensive to evaluate. We believe that the increased accuracy of the Wasserstein distance is due to two properties – first, the metric does not include calculating the inverse of covariance matrices, thereby enclosing the cases with singularity; in contrast, the KL-divergence between two distributions could easily reach infinity if the covariance of the second distribution becomes singular. The second property, which we hypothesise, is the increased statistical robustness of the Wasserstein distance, i.e., the metric might have small variance when comparing distributions that are closely situated in the parametric manifold.

Although, we have utilised the final convolutional layer of a CNN to distinguish images; much empirical work has shown that there are many general features of an image or a video that are captured by the initial layer of a CNN [31]. By visualising different layers in [18], it is apparent that the lower layer of CNN can capture more colour information, the higher layers, on the other hand, are more objective. The retrieval result in Figure 2 demonstrates that colour is not adequately captured due to deep local features extracted from the last convolutional layer, which keeps less colour related information. Hence, the fidelity to distinguish images using any of our retrieval criteria should undoubtedly increase with additional 'independent' feature vector that can be computed via the initial or the middle (general to a more specific characterisation of the image) layers of a CNN. Bayesian model averaging or multi-kernel learning, as has been utilised for video-based action recognition might be a way forward [23], [24].

Factors that affect the successful deployment

For a commercial system, speed is an essential ingredient. In fact, computing Wasserstein and Riemann distance have their issues. For example, Wasserstein distance in computer vision was proposed more than a decade ago [22]. The cost of computing optimal transport between two distributions of dimension d is at least $\mathcal{O}(d^3 \log d)$. This is especially not plausible to compute in a commercial environment when feature vectors are generated by deep convolutional neural networks, which are by construction high dimensional. In our study, even after approximating the GMMs as multivariate Normal distributions, the computational inefficiency is inherent, as computing Eqn. 13 proves to be most expensive amongst all the metrics that we compare. A solution emerges in the form of low dimensional embedding of the metric space [13], [16]; such solutions introduce distortions in addition to an increase in computational cost when the embedding dimension becomes larger than four [8]. Additionally, they are not designed to be scalable to take advantages of large-scale GPU resources. [8] has suggested improving the scalability of the distance calculation by using an iterative diagonal scaling algorithm, known as Sinkhorn's algorithm or iterative proportional fitting. We leave this scalability issue for future work.

Similarly, computing the geodesic distance between two co-variance matrices is equally time inefficient – $\mathcal{O}(4d^3)$. The main component of this inefficiency emerges from the generalised eigenvalue equation, particularly for calculating multiple Cholesky factorisations each time a query is initiated. One way forward may be to use Stein's distance [26] while preserving affine invariance and geometric properties inherited by the covariance matrices. Another way ahead is to perform the factorisation on a GPU [17]. This becomes increasingly important if our framework were to be used for indexing of videos (instead of images). This future application relies on returning a set of similar videos in response to a query video. This could replace the current

text based tagged video framework, like that used by several online video platforms, with feature based tagged videos.

References

[1] S. Amari, H. Nagaoka, and D. Harada. *Methods of Information Geometry*. Translations of Mathematical Monographs. American Mathematical Society, 2007. 1

[2] G. Cao, L. R. Bachega, and C. A. Bouman. The sparse matrix transform for covariance estimation and analysis of high dimensional signals. *IEEE Trans. Image Processing*, 20(3):625–640, 2011. 2, 3

[3] G. Cao, C. A. Bouman, and K. J. Webb. Noniterative MAP reconstruction using sparse matrix representations. *IEEE Trans. Image Processing*, 18(9):2085–2099, 2009. 2, 3

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. 2

[5] H.-T. Cheng, L. Koc, J. Harmsen, V. Jain, X. Liu, and H. Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, Deep Learning for Recommender Systems 2016, pages 7–10, New York, NY, USA, 2016. ACM. 1

[6] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[7] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013. 5

[9] W. Förstner and B. Moonen. *A Metric for Covariance Matrices*, pages 299–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. 4

[10] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. 2

[11] K. Friston, B. Sengupta, and G. Auletta. Cognitive dynamics: From attractors to active inference. *Proceedings of the IEEE*, 102(4):427–445, 2014. 1

[12] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, pages 392–407, 2014. 2

[13] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proceedings of the IEEE Computer Society Conference on CVPR*, volume 1, 2004. 5

[14] J. R. Hershey and P. A. Olsen. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, 2007. 3

[15] W. James and J. Stein. Estimation with quadratic loss. In J. Neyman, editor, *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 361–379, 1961. 2

[16] H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007. 5

[17] G. Macindoe. *Hybrid algorithms for efficient Cholesky decomposition and matrix inverse using multicore CPUs with GPU accelerators*. PhD thesis, University College London, 2013. 5

[18] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Computer Society Conference on CVPR*, 2015. 5

[19] M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, 26:35–747, 2005. 4

[20] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, MA, 2012. 2

[21] F. Nielsen and K. Sun. Guaranteed bounds on the Kullback-Leibler divergence of univariate mixtures. *IEEE Signal Processing Letters*, 23(11):1543–1546, Nov 2016. 3

[22] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998. 5

[23] B. Sengupta and Y. Qian. Pillar networks++: Distributed non-parametric deep and wide networks. *arXiv preprint arXiv:1708.06250*, 2017. 5

[24] B. Sengupta and Y. Qian. Pillar networks for action recognition. *arXiv preprint arXiv:1707.06923*, 2017. 5

[25] B. Sengupta, E. Vazquez, V. Simaiaki, M. Sasdelli, Y. Qian, M. Peniak, L. Netherton, and G. Delfino. Large-scale image analysis using docker sandboxing. *CoRR*, abs/1703.02898, 2017. 1, 2

[26] S. Sra. Positive definite matrices and the s-divergence, 2015. 5

[27] A. Takatsu. On Wasserstein geometry of the space of Gaussian measures. *arXiv preprint arXiv:0801.2250*, 2008. 4

[28] H. Tamura, S. Mori, and T. Yamawak. Textural features corresponding to visual perception. *IEEE Transactions on systems, Man and Cybernetics*, page 460 473, 1978. 2

[29] A. Van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, pages 2643–2651, 2013. 1

[30] C. Villani. *Optimal Transport: Old and New*. Springer-Verlag, 2009. 1, 3, 4

[31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in NIPS 27*, pages 3320–3328. 2014. 5