A TREE-BASED DISTANCE BETWEEN DISTRIBUTIONS: APPLICATION TO CLASSIFICATION OF NEURONS

Riwal Lefort, François Fleuret

IDIAP research institute, Switzerland

ABSTRACT

The usual strategy for computing a distance between two distributions consists of modeling the distributions in feature space, and of computing the distance between the models. We propose here to model the distributions of points by using unsupervised trees. Our main contribution is the definition of a tree-based approximation of the Kullback-Leibler divergence for very large feature spaces, from which we derive a symmetric distance.

Our tree-based KL divergence consists first of building for each set of samples a balanced tree. Then, for any pair of sets of samples, we effectively compute the KL divergence between the empirical distributions at the leaves for the set used to build the tree, and the empirical distribution at the leaves for the other set.

We show experimentally on synthetic data the consistency between this quantity and the exact KL divergence, and demonstrate its efficiency for both unsupervised and supervised classification on multiple standard real-world data-sets. Our main application is the characterization of abnormal neuron development.[1]

Index Terms**—** Distance measurement, Tree data structures, Biological cells

1. INTRODUCTION

Computing the distance between two distributions of points in a high dimensional space remains an important challenge. The usual strategyconsists of first modeling the distributions using a continuous density function such as a mixture of Gaussians, beta law, etc. In such cases, parameters of probability density functions are estimated using maximum likelihood [1], often through an alternating procedure such as expectation maximization [2]. One drawback of this method is its difficulty in processing high-dimensional data-sets. Real-world data-sets often do not fit to a known continuous function, or a single continuous function, and such models are not efficient enough to model both discrete and continuous features.

Recently, specifically in speech recognition [3] and computer vision [4, 5, 6, 7], it has been shown that discretization of the feature space leads to a better approximation of distributions. The corresponding method, dubbed "bag-of-feature" (bof) [4, 5, 6, 7] or "bag-of-words" [3] consists of partitioning the feature space, and, of estimating the empirical distributions over the component of the partition. These techniques usual rely on a Voronoi partitioning obtained by using k-means [4, 5, 6, 7], or on unsupervised trees [8]. The advantage with this latter approach is its computational efficiency and good behavior in high dimensional spaces.

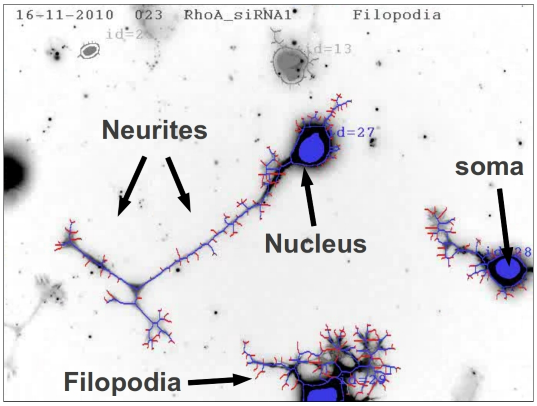

Fig. 1. The genotype of neurons determines their morphodynamics. Neurons can be slow or fast, neurites can grow or withdraw with different frequencies, etc. By classifying the neurons, we predict if classes of neurons are close or not, depending on their morphodynamical signature (speed, acceleration, etc.).

Once distributions are modeled, distances between models can be computed. For instance with the Kullback-Leibler-based distances for continuous densities [9], or – in the discrete case – with the simple Euclidean distance and the Earth mover distance [4, 5, 6, 7].

In this paper, we focus on the discrete case. We propose to model distributions using unsupervised oblique kd-tree [10]. This choice has several advantages [11, 12]: they are adapted to high dimensional feature space, they are easy to learn, there is no optimization problem, no choice of continuous function, and they can mix categorical features and continuous features.

The main contribution of this paper is the definition of a tree-based approximation of the KL-divergence, from which

[1] This work was supported by the Swiss National Science Foundation

we derive a symmetric distance.

In section 2, we describe our framework, and define notation and objectives. Section 3 presents the proposed distance, and section 4 gives experimental results. In the experiments, we propose an application to supervised classification of neurons, the objective being to characterize the morphodynamics of neurons in videos (Figure 1).

2. FRAMEWORK

Let $\mathbf{X}_i = \{X_{i,1} \dots X_{i,N_i}\}$ be a set of N_i points in a feature space. Let Q_i denotes the distribution of $\mathbf{X}_i$.

In the case of videos for instance, $\mathbf{X}_i$ could stand for a set of SIFT key points [4, 5, 6, 7] in the video #i. In natural language processing, $\mathbf{X}_i$ could represent a set of words [3] in document #i. Our objective is to learn a classifier which can classify any $\mathbf{X}_i$.

To achieve this goal, we want to define a distance d between $\mathbf{X}_i$s, from which we will derive a kernel

$$K_{i,j} = \exp\left(-\frac{d^2(\mathbf{X_i}, \mathbf{X_j})}{2\sigma^2}\right) \tag{1}$$

where σ is a scale parameter [4, 5, 6, 7].

As stated in the introduction, a standard strategy consists of modeling the distributions Q_i by first discretizing the space using the k-means algorithm on the aggregated data. Then, the empirical distribution over the resulting clusters are computed for each instance i. In contrast, we model each distribution Q_i by using unsupervised kd-trees $\mathcal{T}_{X_i}$ [10].

We finally formulate the problem as follows: given two sets of points $\mathbf{X}_i$ and $\mathbf{X}_j$, and given the two unsupervised trees $\mathcal{T}_{X_i}$ and $\mathcal{T}_{X_j}$ built from these sets respectively, what is the distance $d(\mathbf{X_i}, \mathbf{X_j})$ between the two instances indexed by i and j ?

3. TREE-BASED KL DIVERGENCE DISTANCE

3.1. Method

Given two sets $\mathbf{X}_i$ and $\mathbf{X}_j$, we propose to compare the structures of the trees $\mathcal{T}_{X_i}$ and $\mathcal{T}_{X_j}$ built from them, with the distribution of points $\mathbf{X}_i$ and $\mathbf{X}_j$. More specifically, points $\mathbf{X}_i$ are passed through the tree $\mathcal{T}_{X_j}$ and points $\mathbf{X}_j$ are passed through the tree $\mathcal{T}_{X_i}$. We then consider the distributions of $\mathbf{X}_i$ in the leaves of the tree $\mathcal{T}_{X_j}$ and the distributions of $\mathbf{X}_j$ in the leaves of the tree $\mathcal{T}_{X_i}$. Intuitively, if Q_i and Q_j are similar, the points $\mathbf{X}_i$ should reach almost all the leaves of the tree $\mathcal{T}_{X_j}$ and the points $\mathbf{X}_j$ should reach almost all the leaves of the tree $\mathcal{T}_{X_i}$. Also, if Q_i and Q_j are not similar, and supposing that Q_i and Q_j are very distant, only one leaf of $\mathcal{T}_{X_j}$ is reached by the points $\mathbf{X}_i$, and only one leaf of $\mathcal{T}_{X_i}$ is expected to be reached by the points $\mathbf{X}_j$.

Formally, let $\mathcal{T}_{X_i}(\mathbf{X}_j)$ be a vector standing for the distribution of the points $\mathbf{X}_j$ over the leaves of $\mathcal{T}_{X_i}$. A N_i-leaf tree $\mathcal{T}_{X_i}$ is built such that each component of the vector $\mathcal{T}_{X_i}(\mathbf{X}_i)$ equals $\frac{1}{N_i}$. This means that the distributions $\mathcal{T}_{X_i}(\mathbf{X}_i)$ and $\mathcal{T}_{X_j}(\mathbf{X}_j)$ are uniform.

In order to measure the distance between instances i and j, we consider the distance between the distributions $\mathcal{T}_{X_i}(\mathbf{X}_j)$ and $\mathcal{T}_{X_i}(\mathbf{X}_i)$ as well as the distance between the distributions $\mathcal{T}_{X_j}(\mathbf{X}_i)$ and $\mathcal{T}_{X_j}(\mathbf{X}_j)$. For that, we propose the Kullback-Leibler-based divergence:

$$\begin{aligned} d(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{2} KL\left[\mathcal{T}_{X_i}(\mathbf{X}_j), \mathcal{T}_{X_i}(\mathbf{X}_i)\right] + \dots \\ \frac{1}{2} KL\left[\mathcal{T}_{X_j}(\mathbf{X}_i), \mathcal{T}_{X_j}(\mathbf{X}_j)\right] \end{aligned} \tag{2}$$

where $KL\left[\mathcal{T}_{X_i}(\mathbf{X}_j), \mathcal{T}_{X_i}(\mathbf{X}_i)\right]$ is the Kullback-Leibler divergence:

$$KL\left[\mathcal{T}_{X_i}(\mathbf{X}_j), \mathcal{T}_{X_i}(\mathbf{X}_i)\right] = \sum_{n=1}^{N_i} \mathcal{T}^n_{X_i}(\mathbf{X}_j) \log \frac{\mathcal{T}^n_{X_i}(\mathbf{X}_j)}{\mathcal{T}^n_{X_i}(\mathbf{X}_i)} \tag{3}$$

where $\mathcal{T}^n_{X_i}(\mathbf{X}_j)$ is the nth component of the vector $\mathcal{T}_{X_i}(\mathbf{X}_j)$. In other words, $\mathcal{T}^n_{X_i}(\mathbf{X}_j)$ is related to the number of points $X_{j,k}$ that reach the nth leaf of the tree $\mathcal{T}_{X_i}$.

Then, if Q_i and Q_j are similar, the distributions $\mathcal{T}_{X_i}(\mathbf{X}_j)$ and $\mathcal{T}_{X_i}(\mathbf{X}_i)$ are equal, and the divergence measure reaches its minimum value: $KL(\mathcal{T}_{X_i}(\mathbf{X}_j), \mathcal{T}_{X_i}(\mathbf{X}_i)) = 0$. On the contrary, if Q_i and Q_j are widely separated in the feature space, the points $\mathbf{X}_i$ should fill only one leaf of the tree $\mathcal{T}_{X_j}$ and the points $\mathbf{X}_j$ should fill only one leaf of the tree $\mathcal{T}_{X_i}$. In this case, $\mathcal{T}_{X_i}(\mathbf{X}_j)$ and $\mathcal{T}_{X_j}(\mathbf{X}_i)$ have a binary form, i.e. only one component equals one and the others equal zero, and then, the distance $d(\mathbf{X}_i, \mathbf{X}_j)$ must reach its maximum value: $d(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{2}(\log N_i + \log N_j)$.

3.2. Computational cost

For Parzen-windows [13], Gaussian kernels [14], and mixture of Gaussians [9], the complexity for computing the distance between two sets of points is $\mathcal{O}(N_i N_j)$ where N_i and N_j denote the number of points for instance i and j respectively. This may be computationally difficult if N_i and N_j are high or if the number of features is high.

In comparison, our method achieves a significant lower complexity bound. The complexities for building the two trees are $\mathcal{O}(N_i \log(N_i))$ and $\mathcal{O}(N_j \log(N_j))$ respectively. The complexities for passing the samples $\mathbf{X_i}$ in the tree $\mathcal{T}_{X_j}$ is $\mathcal{O}(N_i \log(N_j))$ and the complexities for passing the samples $\mathbf{X_j}$ in the tree $\mathcal{T}_{X_i}$ is $\mathcal{O}(N_j \log(N_i))$. Thus, the final complexity is $\mathcal{O}((N_i + N_j)\log(N_i N_j))$.

4. EXPERIMENTS

4.1. Simulated data-set

Two sets of points $\mathbf{X_i}$ and $\mathbf{X_j}$ are generated such that their corresponding distributions Q_i and Q_j are Gaussians with means μ_i and μ_j respectively and diagonal covariance matrices $\Sigma_i = \Sigma_j$. In this case, the Kullback-Leibler divergence takes the value: $\frac{1}{2}(\mu_i - \mu_j)^T(\mu_i - \mu_j)$. In Figure 2, we plot this value, the proposed approximation of the Kullback-Leibler-based divergence of Eq. (2) for different number of points, as a function of the Euclidean distance between μ_i and μ_j.

We note that the proposed Kullback-Leibler-based divergence of Eq. (2) nearly fit with the theoretical value of the Kullback-Leibler divergence.

For an intuitive understanding, we discuss the extreme values that are reached. If $\mu_i = \mu_j$, we observe that our tree-based distance $d(\mathbf{X}_i, \mathbf{X}_j) \neq 0$ when the theoretical value equals 0. This is due to the fact that different subsequent realizations of a given Gaussian distribution are not exactly the same. If the Euclidean distance between μ_i and μ_j tends towards infinity, the tree-based distance (2) never tends towards infinity. This is due to the finite number of points (N_i and N_j) for each realization. For instance, considering that only one leaf of the tree is reached and the Euclidean distance between μ_i and μ_j tends to infinity, we can easily show that the proposed distance equals $d(\mathbf{X}_i, \mathbf{X}_j) = \log(N)$ where $N = N_i = N_j$ is the number of leaves. Thus, if $N = 10$, $d(\mathbf{X}_i, \mathbf{X}_j) = 2.3$, if $N = 100$, $d(\mathbf{X}_i, \mathbf{X}_j) = 4.6$, if $N = 1,000$, $d(\mathbf{X}_i, \mathbf{X}_j) = 6.9$, and if $N = 10,000$, $d(\mathbf{X}_i, \mathbf{X}_j) = 9.2$, which correspond to the rightmost values displayed.

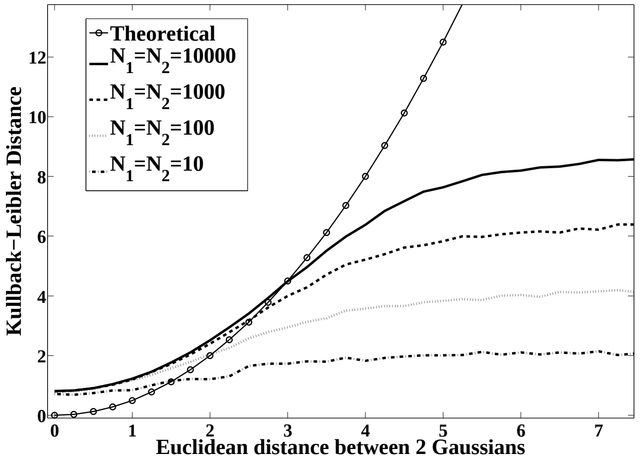

Fig. 2. Comparison between the theoretical Kullback-Leibler distance and the proposed tree-based distance (Eq. (2)). The distance value is plotted as a function of the Euclidean distance between the means of the two generated Gaussians.

4.2. Unsupervised classification

In this section, we compare the tree-based distance (Eq. (2)) and the bof-based distance for an unsupervised image classification task. We use the k-means technique to group similar data together. A fifty-fold cross validation is used to estimate the mean error rate and standard deviation. At each iteration, for each class, 20 objects are sampled from the database. The error e is defined as a function of the pair-wise error: $e = 1 - \frac{TP+TN}{TP+FP+FN+TN}$ where TP denotes a true positive decision, TN denotes a true negative decision, FP denotes a false positive decision, and FN denotes a false negative decision.

In Table 1, we report errors for three data-sets: the CBCL face and car data-set[2], the Amsterdam Library of Object Images [15] (ALOI), and the Head Pose Image Database [16] (HPID). Results show that our method (treeKL) outperforms the baseline (bof) two times out of three in terms of average error, and is more stable on all three. Also, our method does not rely on the setting of sensitive parameters.

	treeKL	bof
CBCL	**0.27±0.06**	0.28±0.12
ALOI	**0.06±0.04**	0.19±0.13
HPID	0.24±0.08	**0.18±0.08**

Table 1. The mean error rates and the standard deviations are reported for the three data-sets (CBCL, ALOI, and HPID). Two distances are used for the k-means: the tree-based distance (Eq. (2)) (treeKL) and the bag-of-feature-based distance (bof) that considers the Euclidean distance between histograms.

4.3. Application to supervised classification of neurons

The characterization of cell morphology and dynamics is an important research topic in modern human biology [17]. Recent work in oncology in particular has shown that cell dynamics provides important information about the cell genotype [18]. We are interested here in the morphodynamics of neurons as captured in videos during their development.

In each video, neurons are segmented and tracked [19], and each neuron is then associated to morphodynamical features such as speed information, acceleration information, information about evolution of the shape of each element of the neuron (see Figure 1), etc. A total of 95 features are extracted.

One experiment corresponds to the comparison between “normal” neurons, and neurons that have been genetically modified. Five genes have been knocked down along four days, which leads to a total number of 47 experiments. In Table 2, we give statistics regarding the classification accuracy over the 47 experiments.

The accuracy is obtained as follows. We use a 100-fold cross validation procedure to compute the average classification rate. Four classifiers are compared: (a) bof+RF: each instance $\mathbf{X_i}$ is modeled using bof and Random Forest [12] (RF)

[2] http://cbcl.mit.edu/projects/cbcl/software-data-sets/

is trained from the histograms of the bof, (b) bof+linSVM: similar to bof+RF but a linear SVM is used, (c) bof+rbfSVM: similar to bof+linSVM but SVM is used with a Gaussian kernel (1) where d stands for the Euclidean distance between histograms of bof, (d) treeKL+rbfSVM: SVM is used with a Gaussian kernel (1) where d stands for the proposed distance (Eq. (2)). At each iteration of the cross validation, we test sets of parameters for the kernel and the bof, and we chose the best parameters.

Analysis of the experiments are given in Table 2. These results show that the proposed method outperforms the bof-based methods on average. This is not surprising, given that tree structures are more efficient than k-means at modelling point distributions in high dimensional space. We also notice that the number of times for which the proposed method outperforms is higher than the bof-based methods.

Regarding the biological analysis, the accuracy reaching 60% in average, we can conclude that a modification of the genotype leads to a modification of the morphodynamics of the neurons.

	mean accuracy	number of outperforming times
bof+RF	0.58±0.21	11
bof+linSVM	0.57±0.21	7
bof+rbfSVM	0.59±0.19	8
treeKL+rbfSVM	0.62±0.19	21

Table 2. Results of the classification experiments. Four classifiers are compared (bof+RF, bof+linSVM, bof+rbfSVM, and treeKL+rbfSVM). The mean accuracy is given as well as the number of times where a given classifier outperforms over all the 47 biological experiments.

5. CONCLUSION

We have proposed a new method for computing the distance between two sets of points in a high dimensional space. Regarding applications, these distributions of points could be key points in images, or words in documents. Our method is based on the discretization of the feature space using unsupervised trees. Once trees are built, we compute a tree-based Kullback-Leibler distance by passing the points of one set in the tree which is associated with another set. The major advantages of the method are the absence of parameters, the high dimensional tolerance, and its ability to mix continuous features with categorical features.

Experimentally, we showed first that the proposed approximation has a behavior consistent with the exact Kullback-Leibler divergence on synthetic data, and we also demonstrated on real-world data-sets that this method can outperform the baseline, i.e. bag-of-feature, in unsupervised learning as well as in supervised learning. We showed in particular how our method can be applied to the characterization of abnormal neuron morphodynamics.

6. REFERENCES

[1] J. Aldrich, “R.a. fisher and the making of maximlum likelihood,” in *Statistical Science*, 1997, vol. 12(3), pp. 162–176.

[2] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” in *Journal of the Royal Statistic Siciety*, 1977, vol. 39, pp. 1–38.

[3] D. Lewis, “Naive (bayes) at forty: The independence assumption in information retrieval,” *ECML*, pp. 4–15, 1998.

[4] G. Csurka, C. Dance, and L. Fan, “Visual categorization with bags of keypoints,” *ECCV Workshop Statistical Learning in Computer Vision*, pp. 59–74, 2004.

[5] L. Fei-Fei, R. Fergus, and A. Torralba, “Recognizing and learning object categories,” *CVPR*, 2007.

[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” *CVPR*, 2008.

[7] L. Duan, D. Xu, I.W. Tsang, and J. Luo, “Visual event recognition in videos by learning from web data,” *CVPR*, 2010.

[8] F. Moosman, E. Nowak, and F. Jurie, “Randomized clustering forests for image classification,” *TPAMI*, vol. 30, no. 9, 2008.

[9] J. Goldberger, S. Gordon, and H. Greenspan, “An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures,” *ICCV*, vol. 2, 2003.

[10] I. Wald and V. Havran, “On building fast kd-trees for ray tracing, and on doing that in o(n log n),” *Symposium on Interactive Ray Tracing*, pp. 61–69, 2006.

[11] J. Quinlan, “C4.5: Programs for machine learning,” *Morgan Kaufmann Publisher*, 1993.

[12] L. Breiman, “Random forest,” *Machine Learning*, vol. 45, pp. 5–32, 2001.

[13] E. Parzen, “On estimation of a probability density function and mode,” *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[14] B. Schölkopf and A. Smola, “Learning with kernel,” *The MIT Press*, 2002.

[15] J.M. Geusebroek, G.J. Burghouts, and A.W.M. Smeulders, “The amsterdam library of object images,” *IJCV*, vol. 61, no. 1, pp. 103–112, 2005.

[16] N. Gourier, D. Hall, and J.L. Crowley, “Estimating face orientation from robust detection of salient facial features,” *International Workshop on Visual Observation of Deictic Gestures*, 2004.

[17] C. Bakal, J. Aach, G. Church, and N. Perrimon, “Quantitative morphological signatures define local signaling networks regulating cell morphology,” *Science*, vol. 316, no. 5832, pp. 1753–1756, 2007.

[18] B. Neumann and all., “Phenotypic profiling of the human genome by time-lapse micorscopy reveals cell division genes,” *Nature*, vol. 464, pp. 721–727, 2010.

[19] G. Gonzáles and all., “Steerable features for statistical 3d dentrite detection,” *MICCAI*, 2009.