Similarity Sensitive Nonlinear Embeddings

Abstract*

Dhruv Batra **Gregory Shakhnarovich**
Toyota Technological Institute–Chicago
Chicago, IL 60637
{dbatra,greg}@ttic.edu

We introduce a parametric, nonlinear embedding scheme for supervised learning of similarity. Given a set of pairs of examples with the corresponding similarity values, our algorithm learns a nonlinear parametric mapping onto a sphere in target space, possibly of dimension higher than that of the original data. The similarity between two points is estimated by the cosine distance (dot product) between their images. The embedding is learned through stochastic optimization of a differentiable albeit nonconvex, objective. Our method achieves results superior to previously proposed metric learning methods (linear and nonlinear) on ten UCI data sets, outdoor scene recognition data set and CIFAR-10 object classification data set.

Background There is a large body of literature on learning similarity, much of it from the last decade. To place our work in this context we consider a few axes along which one can place different methods.

- *Supervision* Our method allows direct supervision, full or partial, with respect to target similarity values, without making assumptions about distance in the original space.
- *Out-of-sample Extension* We propose to learn a functional embedding for data, computable explicitly on novel data.
- *Linearity* The proposed embedding is natively non-linear, and does not require kernel matrix computation in order to compute the embedding.
- *Learning similarity vs. learning to hash* In contrast to many recently published efforts to learn a hashing scheme whose goal is to essentially compress data without sacrificing accuracy w.r.t. *known* distance measure, we directly learn a distance measure that captures similarity as conveyed by examples.

Embedding scheme We would like to construct an M-dimensional embedding $H : \mathcal{X} \rightarrow [-1,1]^M$. The m-th dimension is computed by a (generally nonlinear) parametric function $h_m(\mathbf{x};\mathbf{w}_m) : \mathbb{R}^{d+1} \rightarrow [0,1]$, i.e. the value of each dimension m is defined by the settings of its own parameter vector $\mathbf{w}_m$, applied to the feature vector $\boldsymbol{\phi}(\mathbf{x})$. As a choice of the parametric family of h, in this paper we use the hyperbolic tangent $h_m(\mathbf{x};\mathbf{w}_m) = \frac{2}{1+\exp(\mathbf{w}_m^T\boldsymbol{\phi}(\mathbf{x}))} - 1$, motivated primarily by numerical considerations For notational convenience we can collect the parameters into a $(d+1) \times M$ matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$.

Given $\mathbf{W}$, we can string the resulting embedding values in an unnormalized vector

$$\tilde{H}(\mathbf{x};\mathbf{W}) = [h_1(\mathbf{x};\mathbf{w}_1), \ldots, h_M(\mathbf{x};\mathbf{w}_M)]^T. \quad (1)$$

Finally, the normalized embedding of $\mathbf{x}$ is obtained by $H(\mathbf{x};\mathbf{W}) = \tilde{H}(\mathbf{x};\mathbf{W})/\|\tilde{H}(\mathbf{x};\mathbf{W})\|_2$, i.e., $H(\mathbf{x})$ is a point on the M-dimensional unit sphere. Similarity in the embedding space is measured by the dot product

$$\hat{S}(\mathbf{x},\mathbf{z};\mathbf{W}) \triangleq H(\mathbf{x};\mathbf{W})^T H(\mathbf{z};\mathbf{W}). \quad (2)$$

*Full paper is under review

	Euc.	Euc. (Scaled)	ITML	LMNN	mLMNN	NCA	SSNE
ionosphere	84.23 (±0.68)	83.56 (±0.56)	87.40 (±0.79)	88.65 (±0.77)	**90.58** (±0.83)	88.08 (±1.25)	89.33 (±0.63)
balance	82.58 (±0.40)	82.85 (±0.66)	86.51 (±0.63)	89.30 (±0.58)	88.60 (±0.84)	92.15 (±0.97)	**92.90** (±0.49)
wdbc	63.29 (±0.29)	96.71 (±0.31)	95.53 (±0.38)	96.18 (±0.29)	96.94 (±0.27)	96.00 (±0.36)	**97.12** (±0.33)
protein	**72.58** (±1.46)	65.48 (±1.80)	69.03 (±1.46)	69.03 (±1.46)	68.06 (±2.82)	64.84 (±2.07)	71.29 (±2.07)
pima	67.74 (±0.81)	74.04 (±0.88)	73.13 (±0.80)	74.22 (±0.71)	74.22 (±0.83)	73.35 (±0.65)	**74.83** (±0.70)
wine	57.88 (±1.63)	96.15 (±0.76)	97.31 (±0.82)	98.08 (±0.41)	98.46 (±0.48)	97.12 (±0.72)	**98.46** (±0.38)
iris	95.78 (±0.40)	96.22 (±0.81)	**97.11** (±0.94)	96.22 (±0.88)	96.00 (±1.04)	95.56 (±1.15)	96.22 (±1.00)
heart	62.96 (±1.09)	**83.21** (±0.99)	80.86 (±0.91)	81.98 (±1.38)	81.98 (±1.30)	80.62 (±1.31)	83.09 (±1.12)
sonar	74.52 (±2.17)	77.10 (±1.61)	72.10 (±2.04)	78.87 (±1.24)	80.48 (±1.14)	**81.61** (±1.69)	78.71 (±1.54)
glass	63.11 (±1.53)	66.07 (±1.69)	65.57 (±2.25)	66.23 (±1.75)	**67.38** (±1.99)	64.75 (±1.72)	67.21 (±0.73)

Table 1: Classification accuracies for the UCI datasets, mean and std. deviation of accuracy for 5 times repeated 2-fold cross validation. Bold indicates top performer. See text for details.

One reason for this normalization is the natural expectation that any point be maximally similar to itself; absent the normalization by the norm, this may not be the case. The other reason is that of numerical stability; normalized embedding dampens the influence of outliers. Finally, this ensures that most similar pairs w.r.t. the dot product similarity are also nearest neighbors w.r.t. Euclidean distance.

Learning We learn the embedding by direct optimization of the regularized objective:

$$J(\mathbf{W}) \;=\; \lambda \sum_{q \in C} \left(s_q - H(\mathbf{x}_{i_{q,1}}; \mathbf{W})^T H(\mathbf{x}_{i_{q,2}}; \mathbf{W}) \right)^2 \;+\; R(\mathbf{W}), \tag{3}$$

where $(\mathbf{x}_{i_{q,1}}, \mathbf{x}_{i_{q,2}})$ is the q-th pair training examples for which similarity value s_q is given, and $R(\mathbf{W})$ is a regularization term, penalizing the $L_{1,2}$ group norm of $\mathbf{W}$. Such regularization is expected to drive entire columns of $\mathbf{W}$ to zero if they do not contribute sufficiently to squared loss minimization. Parameter λ controls the tradeoff between loss and norm penalty.

While the objective in (3) is non-convex it's differentiable and we optimize it by batch-stochastic block-coordinate descent: presenting small batches of training pairs and iteratively updating columns of $\mathbf{W}$.

Results We compared our method to Euclidean distance (with or without coordinate scaling), ITML, LMNN, mLMNN, and NCA. On ten UCI data sets our method achieves best results overall (Table 1).

We also compared these methods to ours on the task of learning similarity to be used in NN classification of images, in two data sets: CIFAR-10 and Outdoor Scene Classification (by Oliva and Torralba). Here we obtained accuracy better than that of competing similarity learning methods, shown in Table 2 (LMNN, not shown, was consistently dominated by mLMNN). The experiments are based on a single partition of each data set into train and test, and reporting test accuracy (10 classes in each case).

Data set	Train	Test	SSNE	Euclidean	ITML	mLMNN	NCA
CIFAR-10	50000	10000	**61.49**	60.47	49.13	51.89	50.46
OSC	800	1886	**81.94**	70.82	60.33	80.56	76.43

Table 2: Results on vision data sets. Numbers are test accuracy in % for k-NN classification.

The results reported above demonstrate that SSNE is quite effective in learning an embedding of the data suitable for k-NN prediction. While none of the methods we compared is superior across the board, SSNE comes the closest. However there is a big gap between the accuracies obtained with k-NN classifier on the vision data sets and the state-of-the-art figures reported in literature. Methods that achieve state-of-the-art results use much more sophisticated techniques, typically relying on spatial pooling of features and use of pyramid kernels. We are interested in extending our approach to benefit from these ideas, and will describe at the workshop our ongoing work in this direction.