

Learning from Heterogeneous Data Sources by Combining Dissimilarities

Brent Castle ^{a,b}

Michael Trosset ^{b,a}

This work was supported by a grant from the Office of Naval Research.

a - School of Informatics and Computing, Indiana University

b - Department of Statistics, Indiana University

Heterogeneous Data

Our goal is to combine information from heterogeneous sources of data for the purpose of classification.

Objects with heterogeneous features:

Websites have content, hierarchical structure, design elements, ...

People have social networks, physical characteristics, travel patterns, ...

Proteins have sequence information, function, three-dimensional structure, ...

Domain experts have constructed specialized measures of pairwise (dis)similarity for particular types of data. These pairwise measures are used for tasks such as content-based retrieval.

Images: Pyramid match kernels.

Graphs: Shortest path distance.

Protein Alignment: HSP scores returned by BLAST.

Example 1: Images & Text

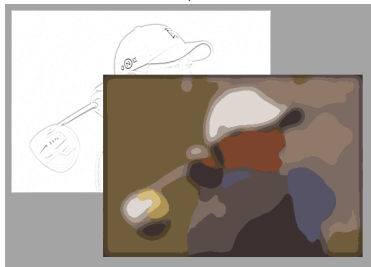
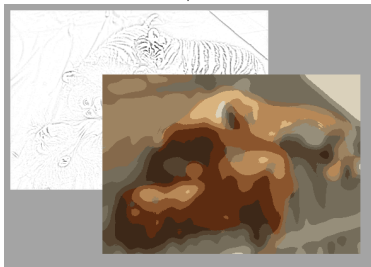


Two Sumatran tiger cub sleep with two baby orangutans in a nursery room at the Taman Safari zoo Wednesday Feb. 28, 2007, in Bogor, Indonesia. The tiger and orangutan babies, which would never be together in the wild, have become inseparable playmates after they were abandoned by their mothers.



Executive chairman Kyi Hla Han said the Asian Tour is set to massively expand with corporate interest higher than ever as golf becomes cool thanks to Tiger Woods, pictured at the WGC-Accenture Match Play Championships last month (AFP/Getty Images/File/Scott Halleran)

Example 2: Shape & Color



Combining Proximities

Let $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^N$ be a set of iid learning instances from a joint distribution D where $x_i \in \Xi$ and $y \in \{1, \dots, G\}$. The feature space Ξ may be the product of several feature spaces, i.e., $\Xi = \Xi_1 \times \dots \times \Xi_q$.

We seek a function g that assigns a class label y to an unlabeled x .

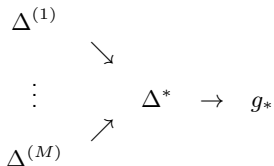
Nearest neighbor classification requires a notion of (dis)similarity. We suppose that individual dissimilarity functions, $\{\delta^{(\ell)} : \Xi \times \Xi \rightarrow \mathfrak{R}_{\geq}\}$, are provided for each feature space.

A dissimilarity measure is a function $\delta : \Xi \times \Xi \rightarrow \mathfrak{R}^+ \cup \{0\}$ that satisfies nonnegativity ($\delta(x_1, x_2) \geq 0$), symmetry ($\delta(x_1, x_2) = \delta(x_2, x_1)$), and reflexivity ($\delta(x, x) = 0$).

We do not require the dissimilarities to satisfy identifiability ($\delta(x_i, x_j) = 0$ iff $i = j$) or the triangle inequality ($\delta(x_1, x_3) \leq \delta(x_1, x_2) + \delta(x_2, x_3)$).

Combining Proximities

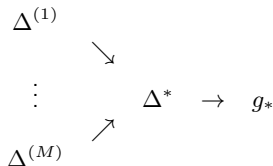
“Combining Dissimilarities”



$\Delta^{(i)}$ are not necessarily metric and g_* is typically a nearest neighbor classifier.

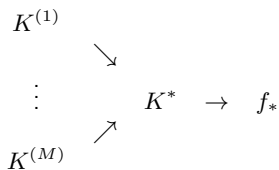
Combining Proximities

“Combining Dissimilarities”



$\Delta^{(i)}$ are not necessarily metric and g_* is typically a nearest neighbor classifier.

“Multiple Kernel Learning” [Lanckriet et al. 2004, Chen and Gupta 2009]



$K^{(i)}$ are typically positive definite kernels, f_* is typically learned by a SVM, and typically $K^* = \sum_{i=1}^M \mu_i K^{(i)}$: $\sum_i \mu_i = 1$ and $\mu_i \geq 0$.

Combining Dissimilarities for Nearest Neighbors (CDNN)

We define the set of triples

$$\mathcal{T} = \{(i, j, k) : x_j \in \mathcal{N}_p(x_i), \text{ and } y_i \neq y_k\}$$

where $\mathcal{N}_p(x_i)$ is the set of the p nearest within-class neighbors.

We seek an effective combination of dissimilarities for k nearest neighbor classification. Toward that end, we attempt to choose δ^* to minimize

$$\frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \mathbb{I}(\delta^*(x_i, x_j) > \delta^*(x_i, x_k))$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Combining Dissimilarities

If $\Delta^{(1)}, \dots, \Delta^{(M)}$ are Euclidean distance matrices one may naturally consider adding the squared dissimilarities, i.e.,

$$\delta_I^*(x_i, x_j) = \sum_{\ell=1}^M \left(\delta^{(\ell)}(x_i, x_j) \right)^2.$$

Adding squared distances \Leftrightarrow Concatenating feature spaces

We can rewrite the above as

$$\delta_I^*(x_i, x_j) = \pi_{ij}^T I \pi_{ij}$$

where $\pi_{ij} = (\delta^{(1)}(x_i, x_j), \delta^{(2)}(x_i, x_j), \dots, \delta^{(M)}(x_i, x_j))$.

Combining Dissimilarities

Let $\mathbf{w} = (w_1, w_2, \dots, w_M)^T$ lie in the unit simplex. A natural extension of the previous formulation is to weight the squared dissimilarities, i.e.,

$$\delta_w^*(x_i, x_j) = \sum_{\ell=1}^M w_\ell \left(\delta^{(\ell)}(x_i, x_j) \right)^2.$$

Adding weighted
squared distances

\Leftrightarrow

Concatenating weighted
feature spaces

We can rewrite the above as

$$\delta_w^*(x_i, x_j) = \pi_{ij}^T W \pi_{ij}$$

where $W = \text{diag}(\mathbf{w})$.

Combining Dissimilarities

We are interested in learning nonnegative dissimilarities. Therefore, one natural extension is to replace W with any positive semidefinite matrix A .

$$\delta_A^*(x_i, x_j) = \pi_{ij}^T A \pi_{ij}$$

This formulation is intimately related to *Distance Metric Learning* (i.e., learning a Mahalanobis distance function) [Weinberger et al., 2006], [Weinberger and Saul, 2009], and [Jin et al., 2009].

Combining Dissimilarities

We are interested in learning nonnegative dissimilarities. Therefore, one natural extension is to replace W with any positive semidefinite matrix A .

$$\delta_A^*(x_i, x_j) = \pi_{ij}^T A \pi_{ij}$$

This formulation is intimately related to *Distance Metric Learning* (i.e., learning a Mahalanobis distance function) [Weinberger et al., 2006], [Weinberger and Saul, 2009], and [Jin et al., 2009].

However, we can exploit the fact that $\pi_{ij} \succeq 0$.

Combining Dissimilarities

The $M \times M$ copositive cone is defined as

$$\mathcal{C}_M = \{A \in \mathcal{S}_M : v^T A v \geq 0, \forall v \succeq 0\}.$$

If $A \in \mathcal{C}_M$, then $\delta_A^*(x_i, x_j) = \pi_{ij}^T A \pi_{ij}$ is nonnegative.

Notice that the copositive cone contains the positive definite cone.

Combining Dissimilarities for Nearest Neighbors (CDNN)

We define the set of triples

$$\mathcal{T} = \{(i, j, k) : x_j \in \mathcal{N}_p(x_i), \text{ and } y_i \neq y_k\}$$

where $\mathcal{N}_p(x_i)$ is the set of the p nearest within-class neighbors.

We seek an effective combination of dissimilarities for k nearest neighbor classification. Toward that end, we attempt to choose A to minimize

$$\frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \mathbb{I}(\delta_A^*(x_i, x_j) > \delta_A^*(x_i, x_k)) \quad (1)$$

subject to $A \in \mathcal{C}_M$ where $\mathbb{I}(\cdot)$ is the indicator function.

Combining Dissimilarities for Nearest Neighbors (CDNN)

We transform the problem into one with a continuous loss function:

$$\arg \min_{A \in \mathcal{C}_M} f(A; \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} (\delta_A^*(x_i, x_j) - \delta_A^*(x_i, x_k) + 1)_+ + \frac{\lambda}{2} \|A\|_F^2.$$

This ℓ_2 -regularized copositive programming problem is closely related to formulations used for *Distance Metric Learning* [Weinberger et al., 2006], [Weinberger and Saul, 2009], and [Jin et al., 2009].

Combining Dissimilarities for Nearest Neighbors (CDNN)

Unfortunately, copositive programming is NP-Hard! In fact, determining whether A is in \mathcal{C}_M is even co-NP-Hard!! We propose a relaxation of the copositive cone that ensures nonnegativity over a finite set of dissimilarities

$$\tilde{\mathcal{C}}_M \equiv \{A : \pi_{ij}^T A \pi_{ij} \geq 0, \forall \pi_{ij} \in \Pi\}$$

where Π is the set of π_{ij} for (i, j) pairs in which nonnegative dissimilarity is required.

The resulting constraint is a finite set of linear inequalities that we call the *polyhedral copositive cone relaxation*. We can now rewrite our optimization problem

$$\min_{A \in \tilde{\mathcal{C}}_m} f(A; \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} (\delta_A^*(x_i, x_j) - \delta_A^*(x_i, x_k) + 1)_+ + \frac{\lambda}{2} \|A\|_F^2. \quad (2)$$

Solving (2)

We use a projected quasi-gradient method (a modification of the Pegasos algorithm proposed in Shalev-Schwartz et al. 2007 for solving the Support Vector Machine).

Algorithm

Choose A_0 such that $A_0 \in B \cap \tilde{\mathcal{C}}_M$

Choose r

For $t = 1, 2, \dots, T$:

Choose $\mathcal{T}_t \subseteq \mathcal{T}$ where $|\mathcal{T}_t| = r$

Set $\mathcal{T}_t^+ = \{\tau \in \mathcal{T}_t : (\pi_{ij}^T A_t \pi_{ij} - \pi_{ik}^T A_t \pi_{ik} + 1) > 0\}$

Set $\eta_t = \frac{1}{\lambda t}$

Set $A_{t+\frac{1}{2}} = (1 - \eta_t \lambda) A_t - \frac{\eta_t}{r} \sum_{\tau \in \mathcal{T}_t^+} (\pi_{ij} \pi_{ij}^T - \pi_{ik} \pi_{ik}^T)$

Set A_{t+1} to the projection of $A_{t+\frac{1}{2}}$ onto $B \cap \tilde{\mathcal{C}}_M$. (QCQP)

Output: A_T .

Experiment #1 - 5-Dimensional Cylinder

Let $x_i \in \mathcal{N}(0, I_5)$ where $y_i = 1$ if $x_{i,1}^2 + x_{i,2}^2 \leq r$ and $y_i = 2$ otherwise.

There are 5 dissimilarities measures. Each one is the Euclidean distance in one dimension of the feature space.

We used 100 instances per training and test set, λ chosen by 5-fold CV.

Error rates (over 20 repetitions):

	1-NN
Δ_1	0.256 ± 0.061
Δ_2	0.259 ± 0.054
Δ_3	0.391 ± 0.038
Δ_4	0.405 ± 0.064
Δ_5	0.402 ± 0.045
Pythagorean CDNN	0.198 ± 0.034 0.125 ± 0.033

Experiment #2 - Protein Fold Prediction

A problem described in [Damoulas and Girolami. *Bioinformatics*, 2008] has 27 classes, 694 proteins, and 12 measures of similarity. The parameter λ was chosen by 5-fold CV. We performed 20 replications with training and test sets selected at random.

Average performance (1-error rate):

	1-NN
Composition	0.370 \pm 0.009
Secondary	0.258 \pm 0.020
Hydrophobicity	0.240 \pm 0.017
Volume	0.289 \pm 0.009
Polarity	0.317 \pm 0.018
Polarizability	0.189 \pm 0.022
L1	0.249 \pm 0.029
L4	0.216 \pm 0.008
L14	0.361 \pm 0.015
L30	0.245 \pm 0.009
SWblosum62	0.441 \pm 0.033
SWpam50	0.425 \pm 0.021
Pythagorean	0.416 \pm 0.016
CDNN	0.457 \pm 0.019

Generalization Bound for CDNN

We are interested in understanding the expected loss, or risk, i.e.,

$$\mathcal{R}(A; \mathcal{S}) = \mathbb{E} \left[\widehat{\mathcal{R}}(A; \mathcal{S}) | \mathcal{S} \right] = \mathbb{E} \left[\frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} (1 + \pi_{ij}^T A \pi_{ij} - \pi_{ik}^T A \pi_{ik})_+ | \mathcal{S} \right]$$

Let γ be an upper bound on the number of times an object is a nearest neighbor. For 1-NN and a constant γ , we show that for any $N \geq 1$ and any $\epsilon \in \{0, 1\}$ the following holds with probability $1 - \epsilon$ for any random draw from D^N of the training set \mathcal{S}

$$\mathcal{R}(A; \mathcal{S}) \leq \widehat{\mathcal{R}}(A; \mathcal{S}) + \frac{2\kappa}{\rho N} + \left(\frac{2\kappa}{\rho} + \frac{1}{\sqrt{\lambda \rho N}} \right) \sqrt{\frac{\ln 1/\epsilon}{2N}}$$

where κ is a constant. The proof relies on the uniform stability of CDNN and McDiarmid's Inequality.

Comments / Future Work

- ▶ We are interested in understanding performance when one measure significantly outperforms the others.
- ▶ We are interested in performing an intermediate feature selection prior to combining the dissimilarities. [Castle, Tang, and Trosset 2009 [†]] addresses this issue for Multiple Kernel Learning and found promising results.

[†] - IU Department of Statistics Technical Report

Thank you!