

# LATENT COINCIDENCE ANALYSIS: A HIDDEN VARIABLE MODEL FOR DISTANCE METRIC LEARNING

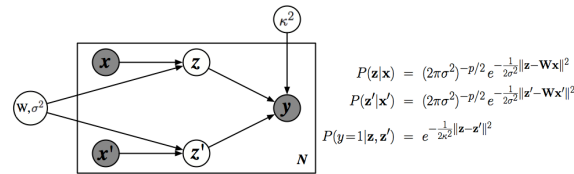
Matthew Der and Lawrence K. Saul, University of California, San Diego

## 1 INTRODUCTION

Latent Coincidence Analysis (LCA) is a latent variable model for supervised dimensionality reduction and distance metric learning. The model discovers linear projections of data that shrink the distance between similarly labeled inputs and expand the distance between differently labeled ones. Inference is completely tractable, and we derive an Expectation-Maximization (EM) algorithm for parameter estimation. The model's main advantage is its simplicity: at each iteration of the EM algorithm, the distance metric is re-estimated by solving an unconstrained least-squares problem.

## 2 MODEL

This is LCA's representation as a Bayesian network. There are three observed variables: the inputs  $\mathbf{x}, \mathbf{x}' \in \mathbf{R}^d$ , which we always imagine to be observed in pairs, and the binary label  $y \in \{0, 1\}$ , which indicates if the inputs should map to nearby locations in the latent space. These locations are represented by the Gaussian latent variables  $\mathbf{z}, \mathbf{z}' \in \mathbf{R}^p$ .



### 2.1 INFERENCE

Inference requires averaging over  $\mathbf{z}, \mathbf{z}'$ . The required integrals take the form of simple Gaussian convolutions:

$$P(y=1|\mathbf{x}, \mathbf{x}') = \int d\mathbf{z} d\mathbf{z}' P(y=1|\mathbf{z}, \mathbf{z}') P(\mathbf{z}|\mathbf{x}) P(\mathbf{z}'|\mathbf{x}')$$

$$= \left( \frac{\kappa^2}{\kappa^2 + 2\sigma^2} \right)^{p/2} \exp\left( -\frac{\|\mathbf{W}(\mathbf{x}-\mathbf{x}')\|^2}{2(\kappa^2 + 2\sigma^2)} \right)$$

Of special importance for learning are the statistics of the posterior distribution obtained using Bayes rule:

$$P(\mathbf{z}, \mathbf{z}'|\mathbf{x}, \mathbf{x}', y) = \frac{P(y|\mathbf{z}, \mathbf{z}') P(\mathbf{z}|\mathbf{x}) P(\mathbf{z}'|\mathbf{x}')}{P(y|\mathbf{x}, \mathbf{x}')}$$

#### Posterior means:

$$E[\mathbf{z}|\mathbf{x}, \mathbf{x}', y=0] = \mathbf{W} \left[ \mathbf{x} - \left( \frac{\nu\sigma^2}{\kappa^2 + 2\sigma^2} \right) (\mathbf{x}' - \mathbf{x}) \right] \quad \nu = \frac{P(y=1|\mathbf{x}, \mathbf{x}')}{P(y=0|\mathbf{x}, \mathbf{x}')}$$

$$E[\mathbf{z}|\mathbf{x}, \mathbf{x}', y=1] = \mathbf{W} \left[ \mathbf{x} + \left( \frac{\sigma^2}{\kappa^2 + 2\sigma^2} \right) (\mathbf{x}' - \mathbf{x}) \right]$$

#### Posterior variances:

$$E\left[ \|\mathbf{z} - \bar{\mathbf{z}}\|^2 \mid \mathbf{x}, \mathbf{x}', y=0 \right] = p\sigma^2 \left[ 1 + \frac{\nu\sigma^2}{\kappa^2 + 2\sigma^2} \right]$$

$$E\left[ \|\mathbf{z} - \bar{\mathbf{z}}\|^2 \mid \mathbf{x}, \mathbf{x}', y=1 \right] = p\sigma^2 \left[ 1 - \frac{\sigma^2}{\kappa^2 + 2\sigma^2} \right]$$

### 2.2 LEARNING

We use the EM algorithm to learn the parameters that maximize the conditional log-likelihood of the observed data:

$$\mathcal{L}(\mathbf{W}, \sigma^2, \kappa^2) = \sum_{i=1}^N \log P(y_i|\mathbf{x}_i, \mathbf{x}'_i)$$

#### E-step:

Compute statistics of posterior distribution  $P(\mathbf{z}, \mathbf{z}'|\mathbf{x}, \mathbf{x}', y)$

$$\text{means: } \bar{\mathbf{z}}_i, \bar{\mathbf{z}}'_i \quad \text{variance: } \varepsilon_i^2$$

#### M-step:

Update model parameters using those statistics. Minimize sum of squared errors:

$$\mathcal{E}(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^N \left[ \|\bar{\mathbf{z}}_i - \mathbf{W}\mathbf{x}_i\|^2 + \|\bar{\mathbf{z}}'_i - \mathbf{W}\mathbf{x}'_i\|^2 \right]$$

Update rules:

$$\mathbf{W} \leftarrow \left[ \sum_{i=1}^N (\bar{\mathbf{z}}_i \mathbf{x}_i^\top + \bar{\mathbf{z}}'_i \mathbf{x}'_i{}^\top) \right] \left[ \sum_{i=1}^N (\mathbf{x}_i \mathbf{x}_i^\top + \mathbf{x}'_i \mathbf{x}'_i{}^\top) \right]^{-1}$$

$$\sigma^2 \leftarrow \frac{1}{pN} \left[ \min_{\mathbf{W}} \mathcal{E}(\mathbf{W}) + \sum_{i=1}^N \varepsilon_i^2 \right]$$

When  $\kappa^2$  is necessary, we re-estimate it by simple line search.

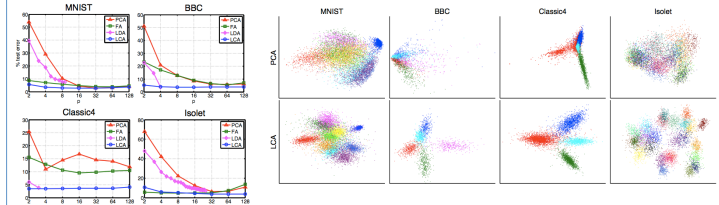
## 3 APPLICATIONS

### 3.1 GAUSSIAN MIXTURE MODELING

We learn one model of LCA for each class  $c$  of the data. We construct a training set of labeled pairs  $\{\mu_c, \mathbf{x}_i, y_{ic}\}$  over all examples  $\mathbf{x}_i$  where  $y_{ic} = 1$  if  $y_i = c$  and  $y_{ic} = 0$  if  $y_i \neq c$ . We use EM to learn a linear projection  $\mathbf{W}_c$  and noise level  $\sigma_c^2$ . To classify an unlabeled example  $\mathbf{x}$ , we compute the probabilities

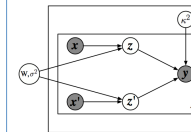
$$P(y_c=1|\mathbf{x}) = \left( \frac{1}{1+2\sigma_c^2} \right)^{p/2} \exp\left\{ -\frac{\|\mathbf{W}_c(\mathbf{x}-\mu_c)\|^2}{2(1+2\sigma_c^2)} \right\}$$

and label  $\mathbf{x}$  by the class  $c$  that maximizes this probability.



### 3.2 DISTANCE METRIC LEARNING

We apply LCA to learn a distance metric that improves kNN classification. We consider an example  $\mathbf{x}_i$  with its *target neighbors* – the  $k$  nearest neighbors in Euclidean space with the same label – as coincident pairs ( $y = 1$ ), and  $\mathbf{x}_i$  with its *impostors* – points with a different label that lie closer than the  $k^{\text{th}}$  target neighbor – as non-coincident pairs ( $y = 0$ ). We include  $\mathbf{x}_i$  paired with its target neighbors and impostors in the training set for LCA if  $\mathbf{x}_i$  has any impostors.



For each  $\mathbf{x}_i$ , we learn a local *length scale* parameter  $\kappa^2$ , which is needed to account for the fact that different inputs may reside at very different distances from their target neighbors.

After training, we perform kNN classification using the Mahalanobis distance metric parameterized by the linear transformation  $\mathbf{W}$ .

