Deep Supervised t-Distributed Embedding

Renqiang (Martin) Min

Joint work with

Laurens van der Maaten, Zineng Yuan, Anthony Bonner, and Zhaolei Zhang

Department of Computer Science

University of Toronto

June 2010

Why Deep Non-linear Embedding

- Embedding is useful for high-dimensional data visualization and data classification

- Deep neural networks pre-trained with RBMs are capable of generating powerful non-linear embeddings
  - Linear mapping is often incapable of capturing higher-order statistics hidden in input feature vector components

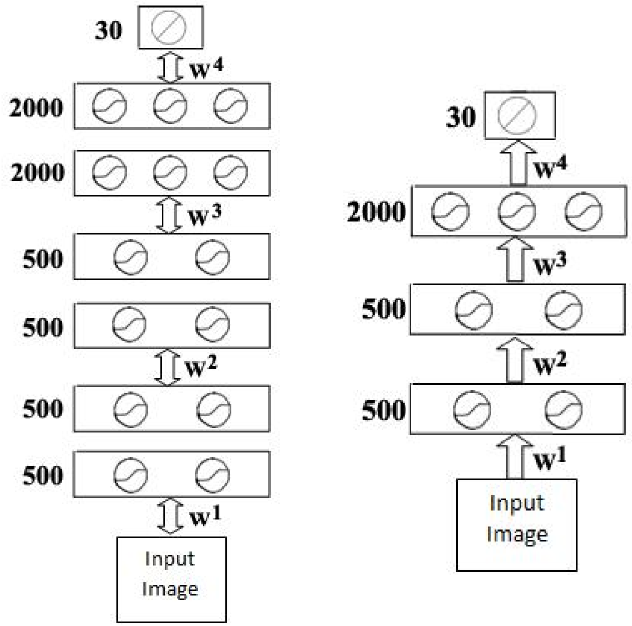
  - Deep learning methods are good at extracting meaningful structure from high-dimensional input features

Extend Supervised Linear Embedding Methods with Deep Neural Networks

- Maximally Collapsing Metric Learning (MCML) learns a linear mapping to collapse all the points in the same class to one point

- Neighborhood Component Analysis (NCA) learns a linear mapping by maximizing the expected number of points correctly classified

- We can use a deep neural network pre-trained with RBMs to learn a deep supervised non-linear embedding by optimizing the cost of MCML and NCA for both high-dimensional data visualization and classification

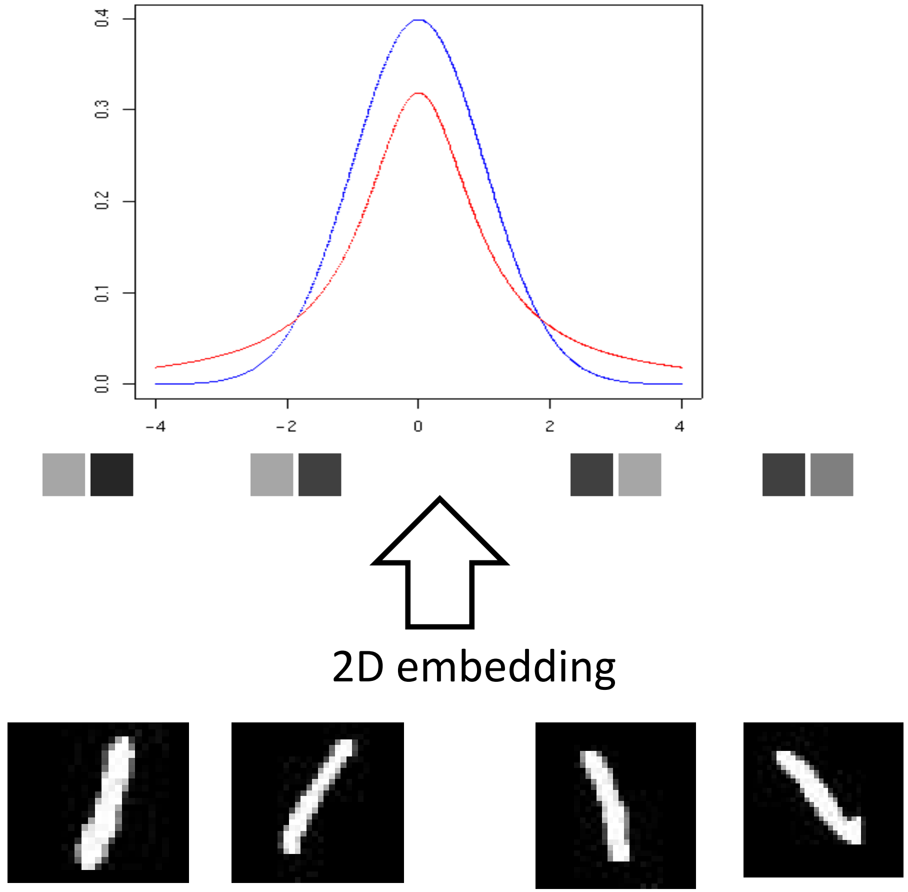


Supervised Peaky and Multimodal Class Collapsing

- Make similar data points in the same class stay close together

- Allow dissimilar data points in the same class to be put far apart in the embedding space

- Different classes of data should be put even further apart



dt-MCML and dt-NCA

- Using a t-distribution for modeling conditional probabilities in the embedded space, dt-MCML collapses classes while dt-NCA maximizes the expected number of points correctly classified

- Collapsing classes works well for very low-dimensional embedding such as two-d embedding, but is unnecessary and might cause overfitting when the dimensionality of the embedded space is large

- dt-NCA is more suitable for higher-dimensional embedding than dt-MCML

dt-MCML

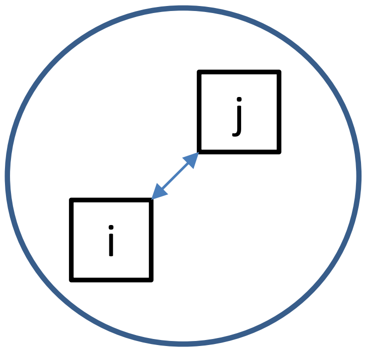
- Unlike in MCML, we use symmetric q distribution to simplify gradient computation:

$$\ell_{dt-MCML} = KL(P||Q) = \sum_i \sum_{j:j\neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{ij} \propto 1 \text{ iff } y^{(i)} = y^{(j)},\ p_{ij} = 0 \text{ iff } y^{(i)} \neq y^{(j)} \qquad \textstyle\sum_{ij} p_{ij} = 1$$

$$q_{ij} = \frac{(1 + d_{ij}^2/\alpha)^{-\frac{1+\alpha}{2}}}{\sum_{kl:k\neq l}(1 + d_{kl}^2/\alpha)^{-\frac{1+\alpha}{2}}}, \quad q_{ii} = 0 \qquad d_{ij}^2 = ||f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(j)})||^2$$

- This objective function is equivalent to the negative log product of q_{ij}s



- Prevent data points in the same class spreadout

dt-NCA

$$\ell_{dt-NCA} = -\sum_{ij:i\neq j} \delta_{ij} q_{j|i},$$

δ_{ij} equals 1 if $y^{(i)} = y^{(j)}$ and 0 otherwise

$$q_{j|i} = \frac{(1 + d_{ij}^2/\alpha)^{-\frac{1+\alpha}{2}}}{\sum_{k:k\neq i}(1 + d_{ik}^2/\alpha)^{-\frac{1+\alpha}{2}}}, \quad q_{i|i} = 0.$$

- dt-NCA uses asymmetric q distribution while dt-MCML uses symmetric q distribution

- dt-NCA maximizes the sum of the probabilities q_ij while dt-MCML maximizes the product of the probabilities q_ij

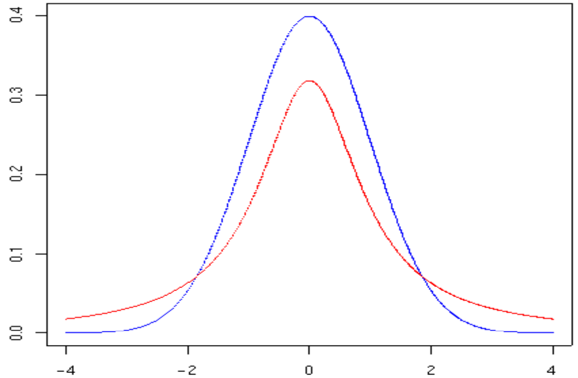
the advantages of using a t-distribution

- In t-SNE, there are no supervision signals, and t-distribution helps to avoid “crowding problem”

- In dt-MCML and dt-NCA:

- allow one class of data to be embedded to different modes

- result in tighter clusters in the embedding



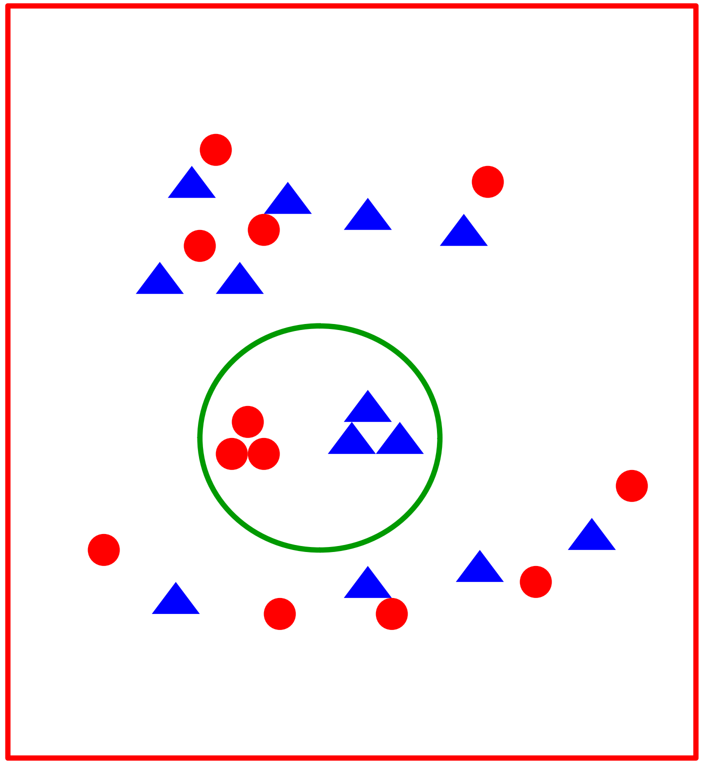
- allow larger separations between classes

- make gradient-based optimization easier: the gradient of the tail of a t-distribution is much deeper than that of a Gaussian

Symmetric / Asymmetric dt-MCML and dt-NCA

dt-MCML
- Symmetric 
- Asymmetric 

dt-NCA
- Asymmetric 
- Symmetric 

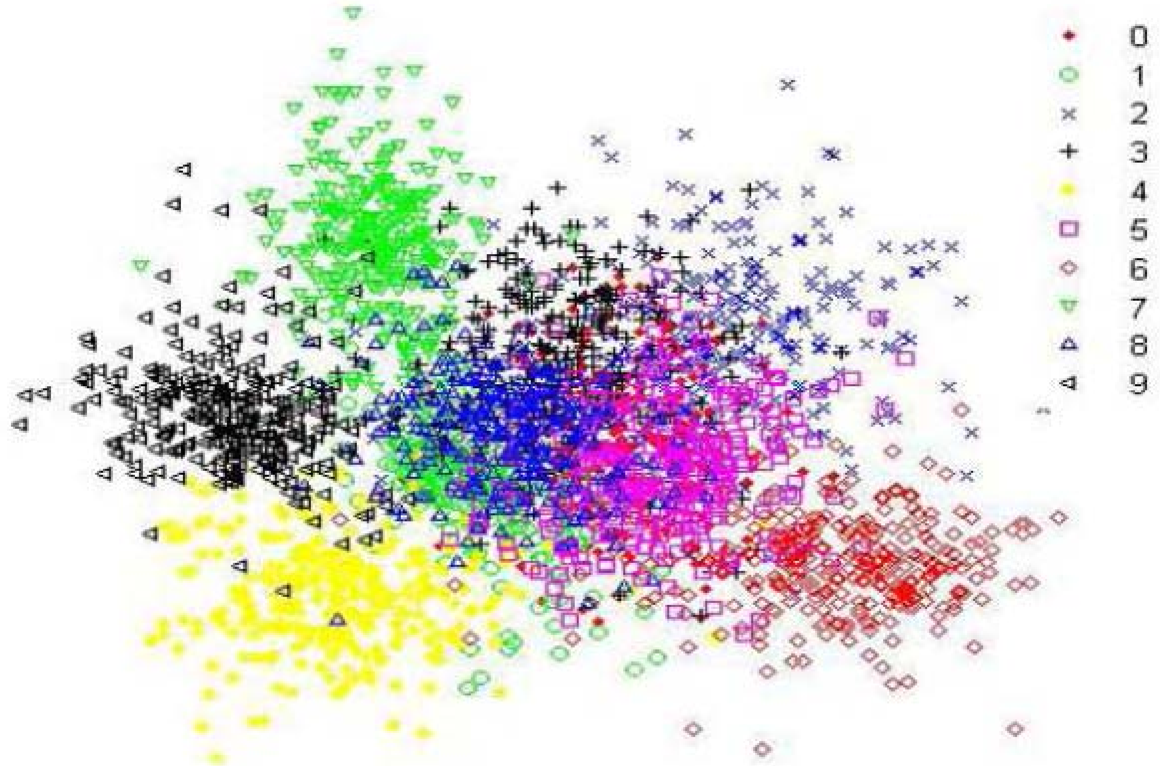


Embedding Results on USPS Digits

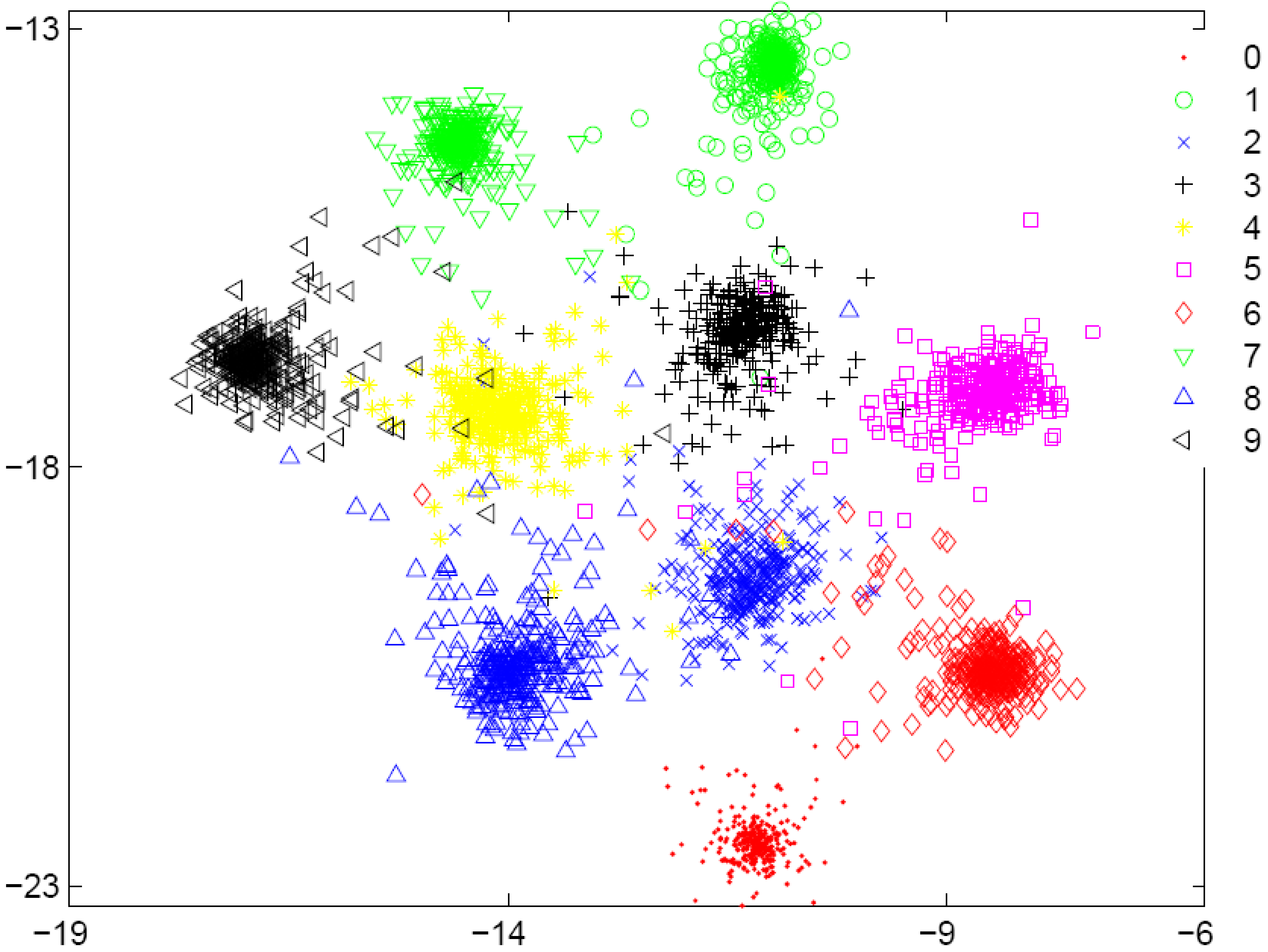
Table 1. Mean and standard deviation of test error (in %) on 2-dimensional and 30-dimensional embedding for various techniques on the 6 splits of USPS data set.

Dimensionality d	**2D**	**30D**
MCML	35.63 ± 0.44	5.53 ± 0.39
dG-MCML	3.37 ± 0.18	1.67 ± 0.21
dt-MCML ($\alpha = d - 1$)	$\mathbf{2.46 \pm 0.35}$	1.73 ± 0.47
dt-MCML (learned α)	2.80 ± 0.36	1.61 ± 0.36
dG-NCA	10.22 ± 0.76	1.91 ± 0.22
dt-NCA ($\alpha = d - 1$)	5.11 ± 0.28	$\mathbf{1.15 \pm 0.21}$
dt-NCA (learned α)	6.69 ± 0.92	1.17 ± 0.07

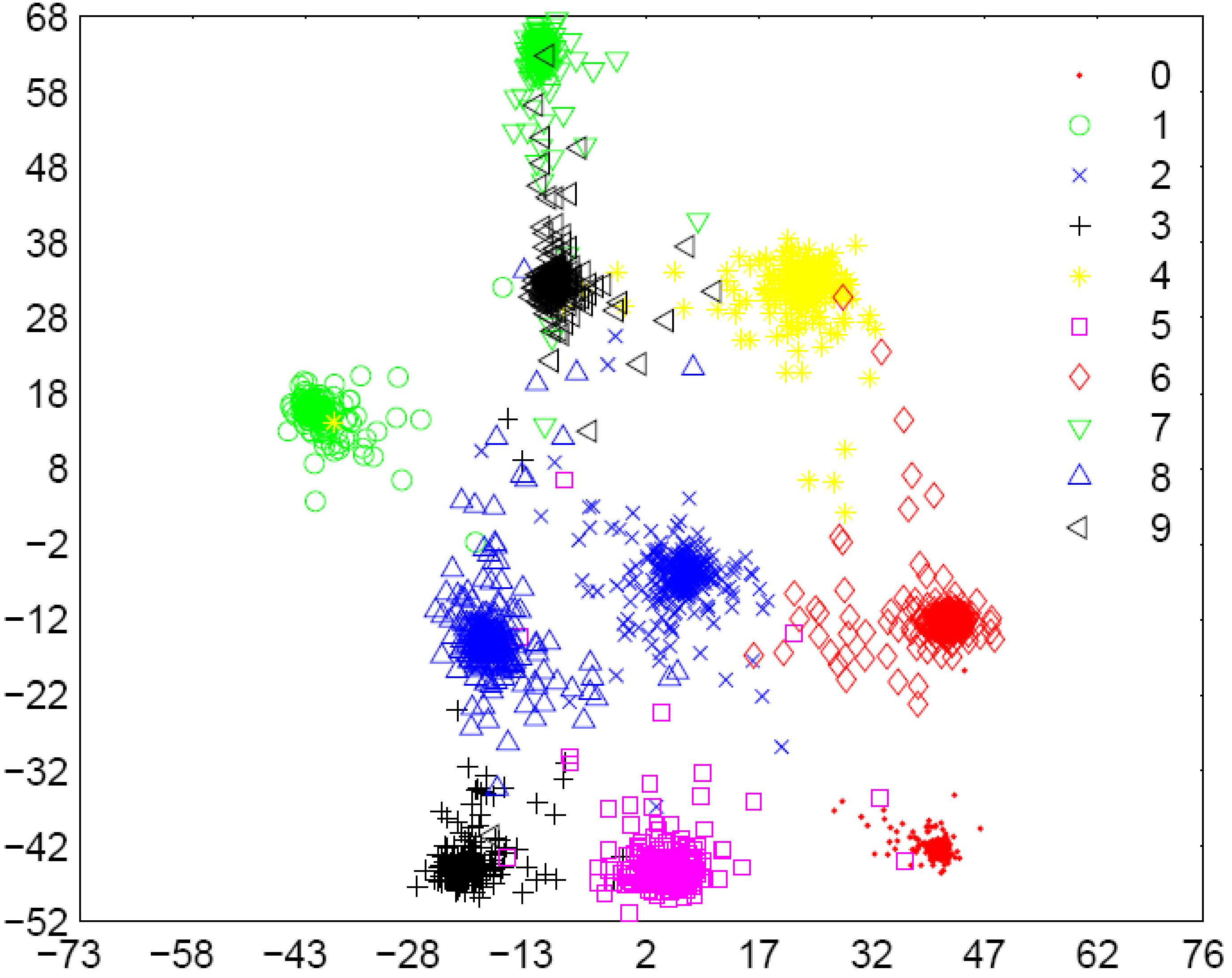
Embedding Results on USPS Digits (MCML)



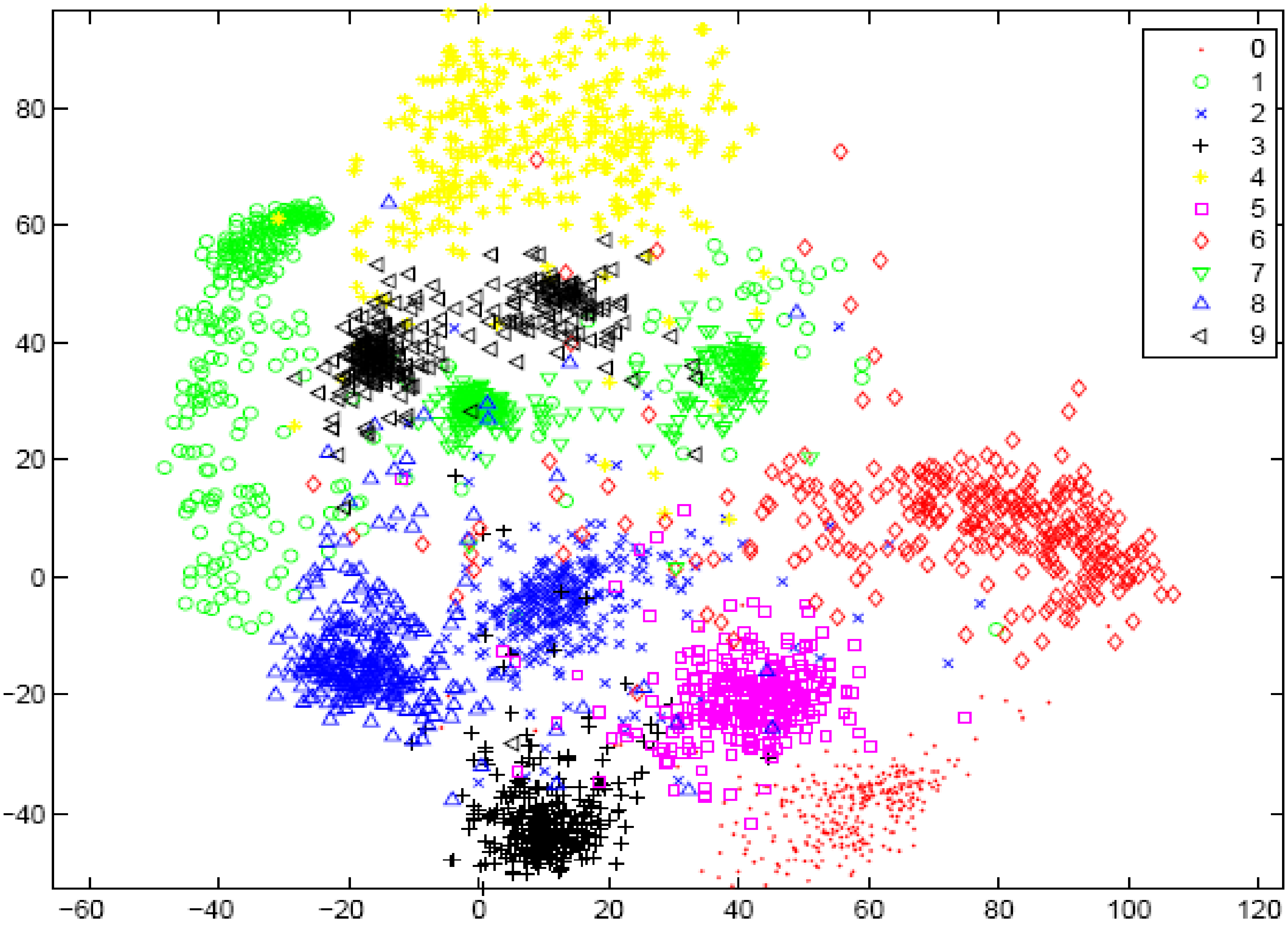
Embedding Results on USPS Digits (dG-MCML)



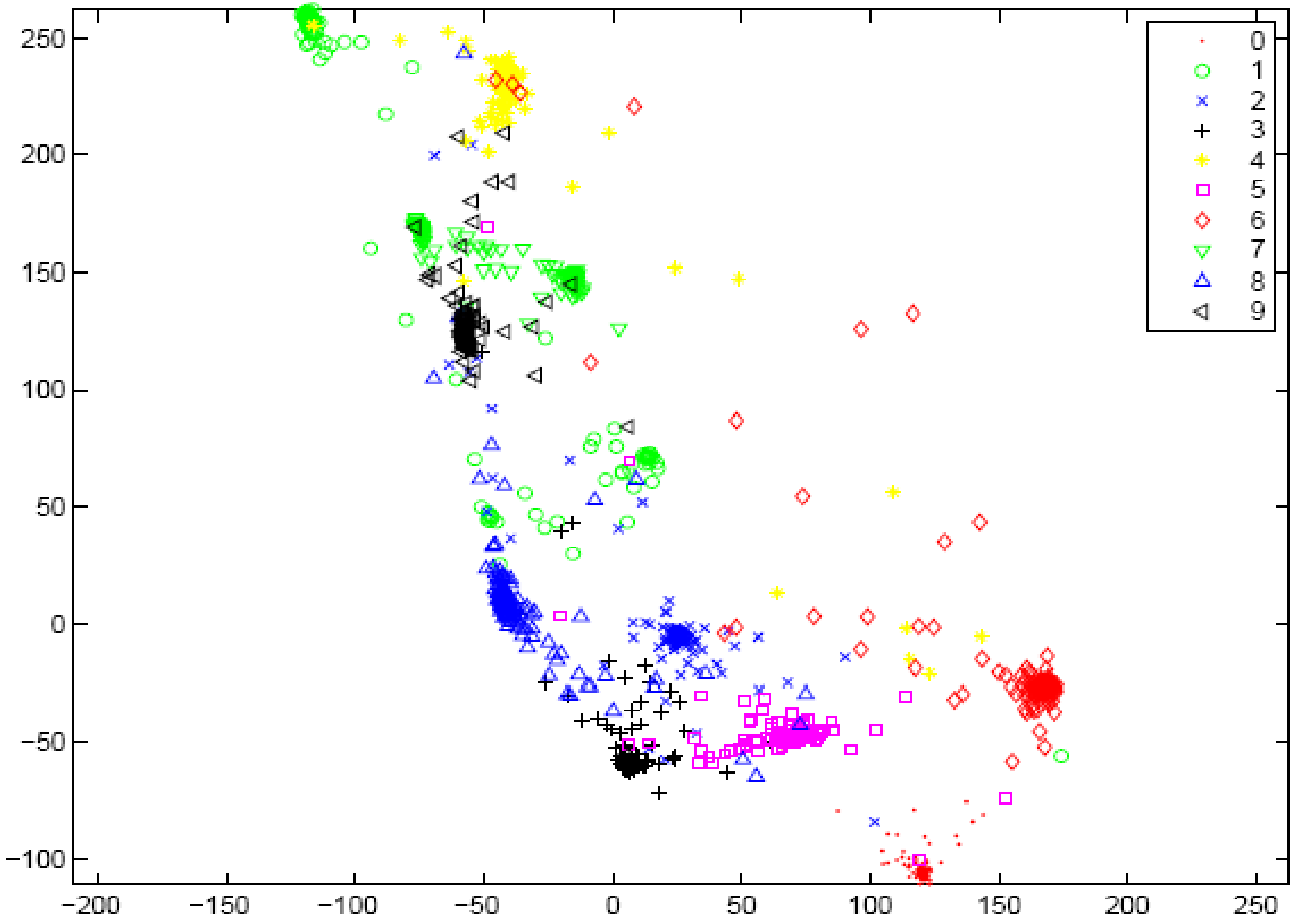
Embedding Results on USPS Digits (dt-MCML)



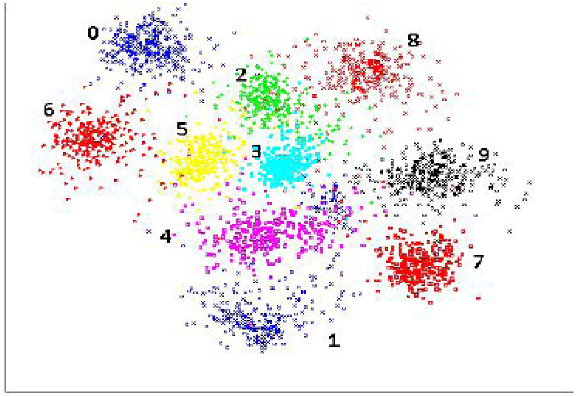
Embedding Results on USPS Digits (dG-NCA)



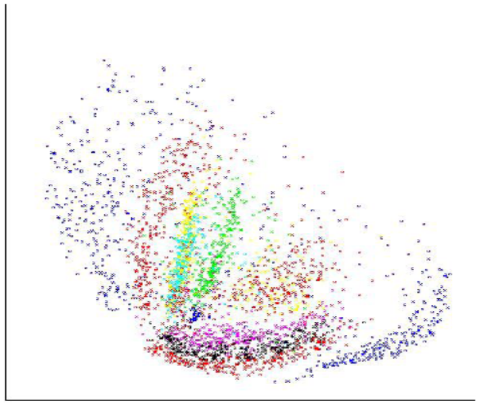
Embedding Results on USPS Digits (dt-NCA)



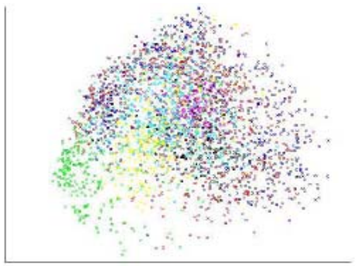
Embedding Results on USPS Handwritten Digits



Two-dimensional embedding of 3000 USPS-fixed test data using the Deep Neural Network kNN classifier (DNet-kNN).



Two-dimensional embedding of 3000 USPS-fixed test data using the Deep Autoencoder (DA).



Two-dimensional embedding of 3000 USPS-fixed test data using PCA.

2D and 30D Embedding Results on MNIST Handwritten Digits

Table 2. Test error (in %) on 2-dimensional and 30-dimensional embedding for various techniques on the MNIST data set.

Dimensionality d	2D	30D
dG-MCML	$\boxed{2.13}$	1.49
dt-MCML ($\alpha = d - 1$)	$\boxed{\mathbf{2.03}}$	1.63
dt-MCML (learned α)	2.14	1.49
dG-NCA	7.95	1.11
dt-NCA ($\alpha = d - 1$)	3.48	$\boxed{\mathbf{0.92}}$
dt-NCA (learned α)	3.79	$\boxed{0.93}$

DNet-kNN (dim = 30, batch size=1.0e4)	**0.94**
DNet-kNN-E (dim = 30, batch size=1.0e4)	0.95
Deep Autoencoder (dim = 30, batch size=1.0e4)	2.13
Non-linear NCA based on a Deep Autoencoder ([16]	1.03
Deep Belief Net [11]	1.25
SVM: degree 9 [4]	1.4
kNN (pixel space)	3.05
LMNN	2.62
LMNN-E	1.58
DNet-kNN (dim = 2, batch size=1.0e4)	2.65
DNet-kNN-E (dim = 2, batch size=1.0e4)	2.65
Deep Autoencoder (dim = 2, batch size=1.0e4)	24.7

Conclusion and Future Work

- Deep neural networks produce better mappings than their linear counterparts, and scale well to massive data sets with batch training
- Heavy-tailed distributions are more suitable for modeling probabilities in low-d space than Gaussian in embedding
- dt-MCML favors 2D embedding for visualization while dt-NCA favors higher-dimensional embedding for classification
- collapsing classes causes overfitting in higher-dimensional embedding
- Approaches here are easily extended to semi-supervised learning settings by combining the supervision signals of dt-MCML or dt-NCA, t-SNE, and an auto-encoder learned with unlabeled data

Acknowledgement

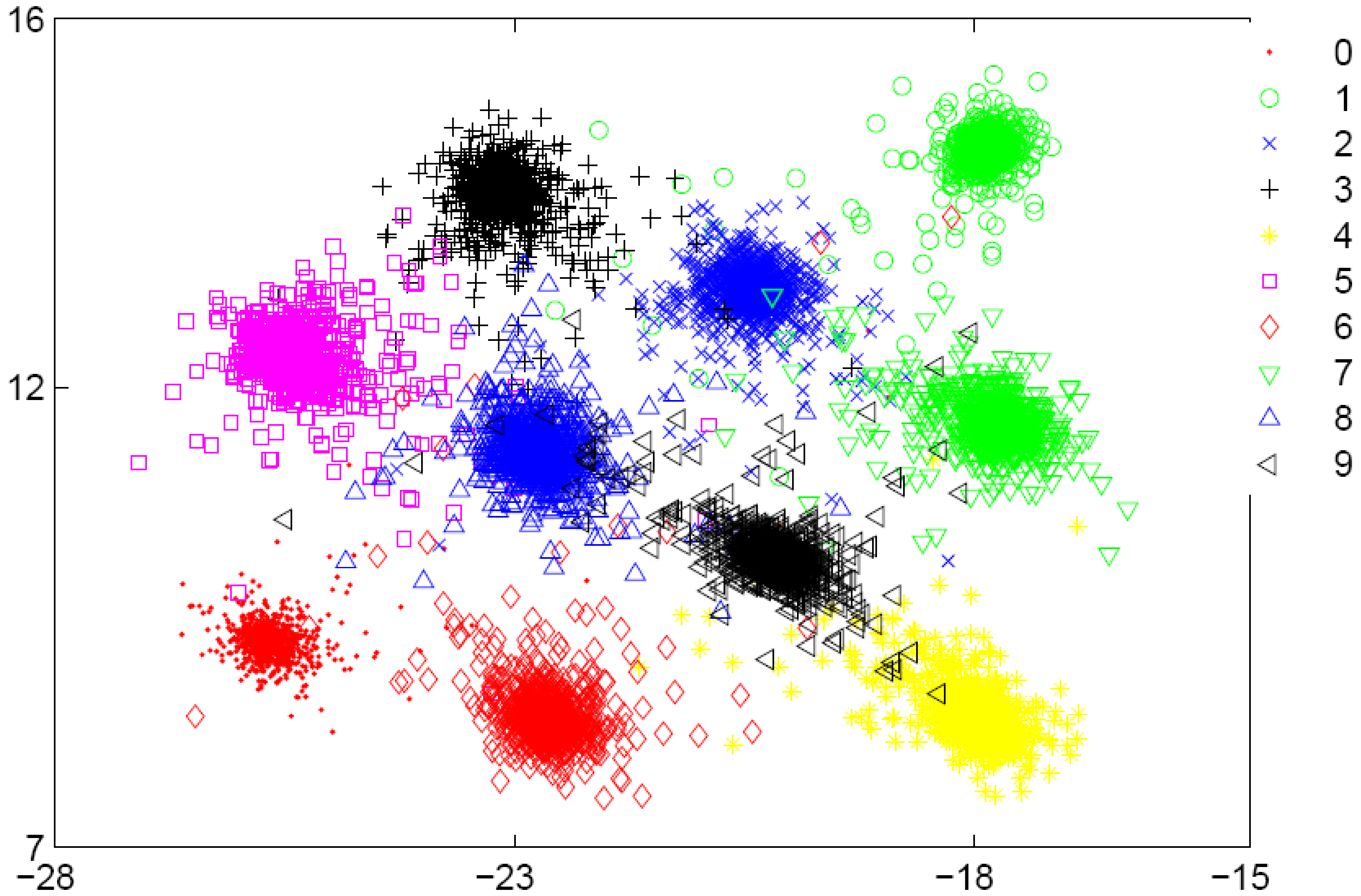
University of Toronto
Geoff Hinton

Hebrew University
Amir Globerson

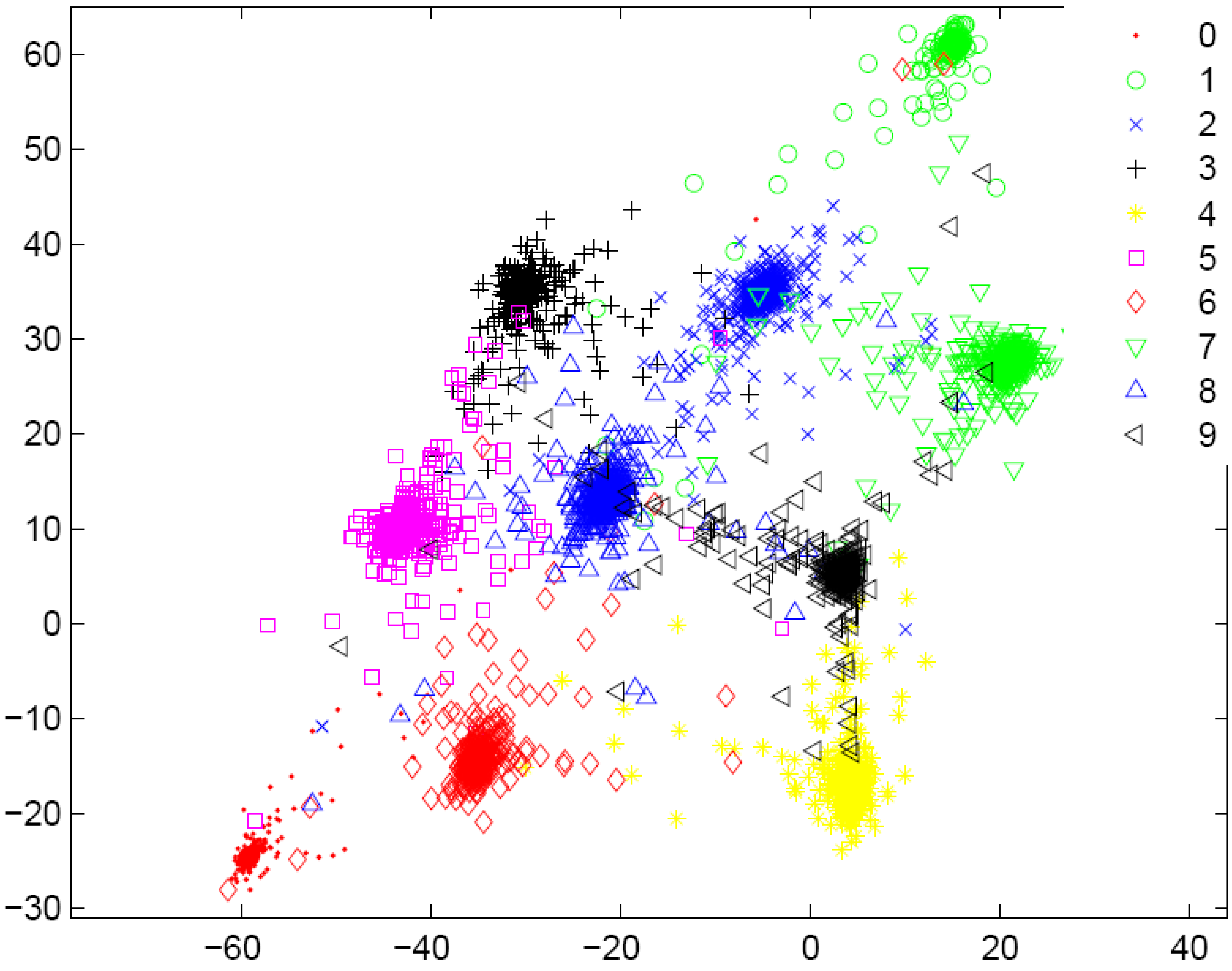
Questions

Thank You

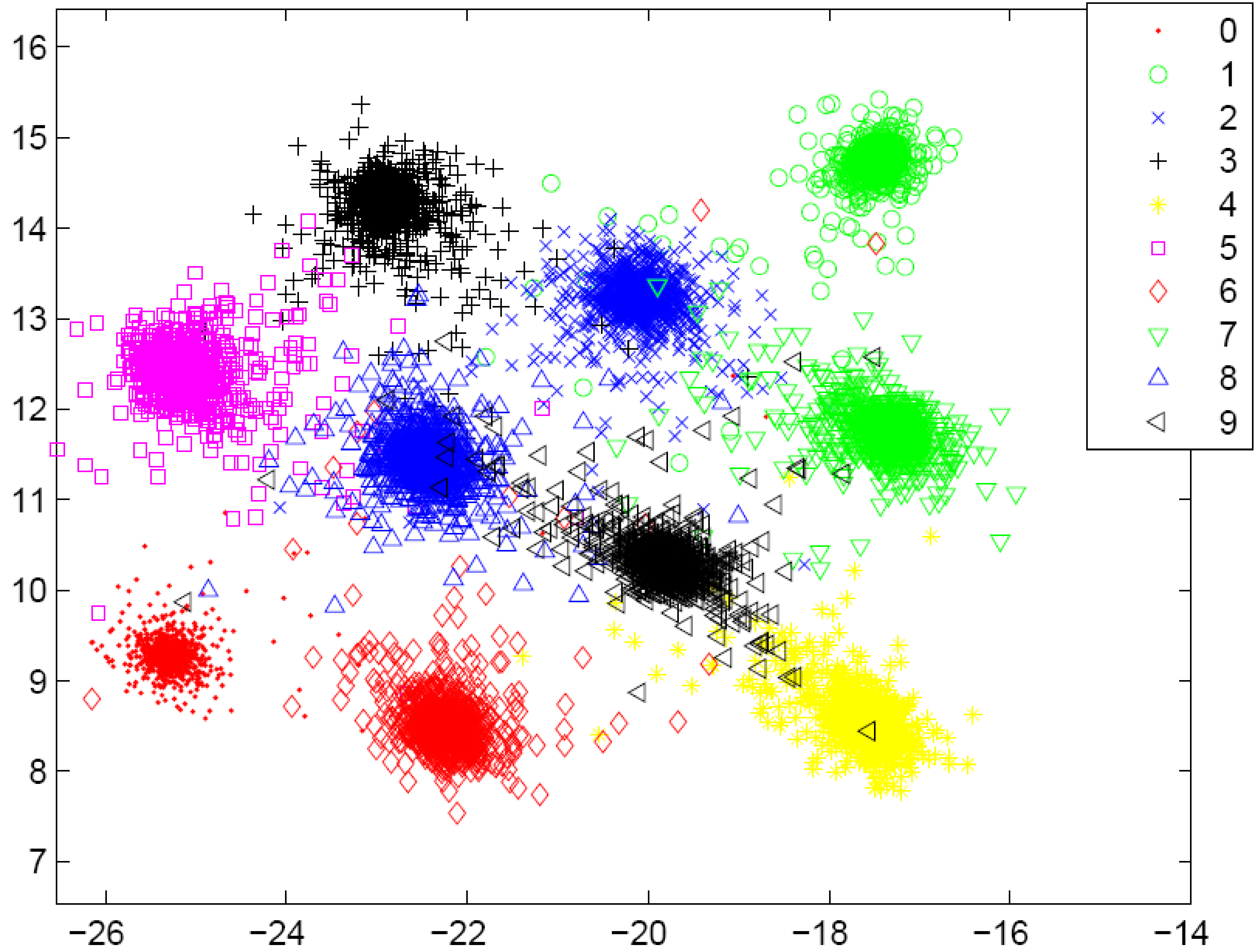
Embedding Results on MNIST Digits (dG-MCML)



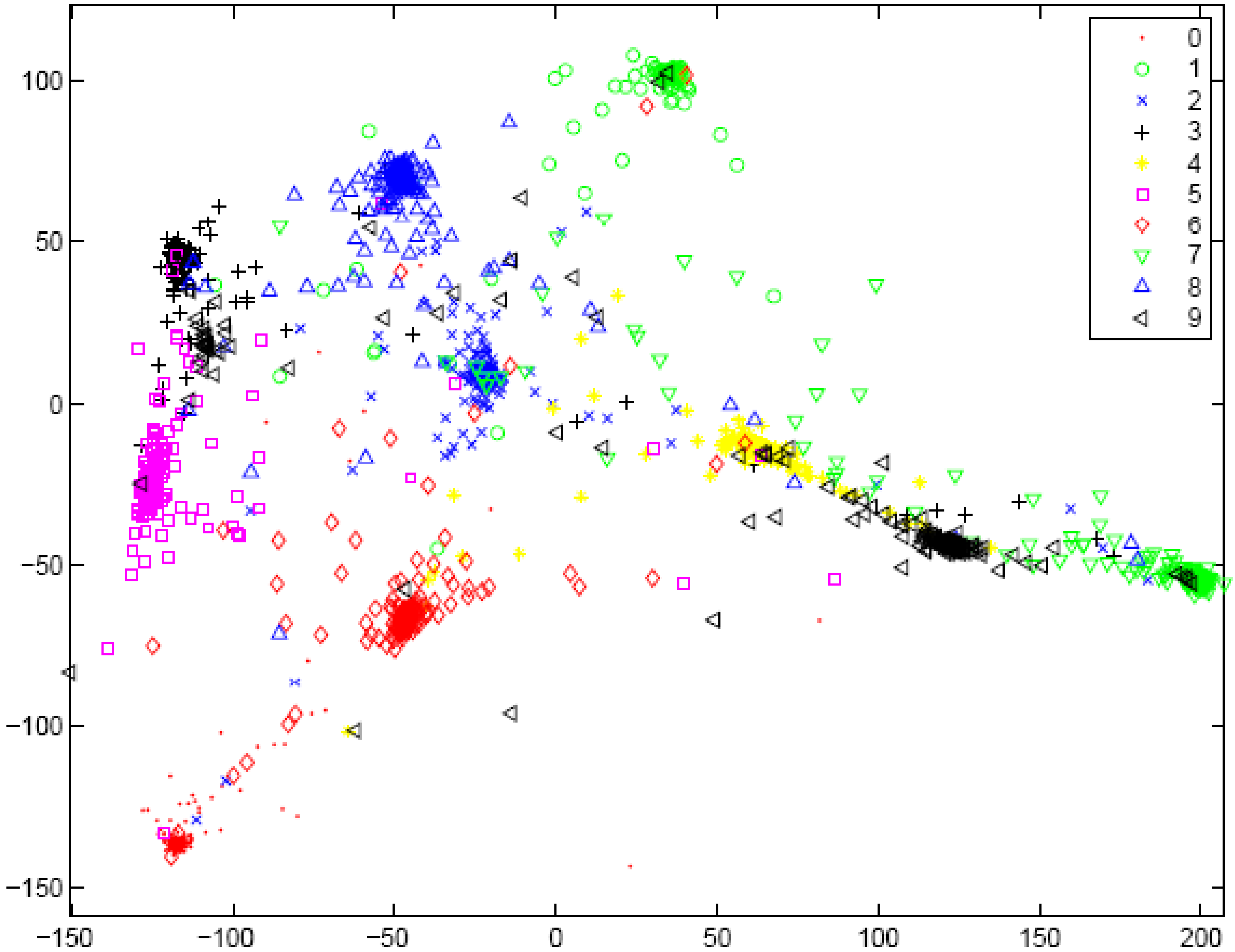
Embedding Results on MNIST Digits (dt-MCML)



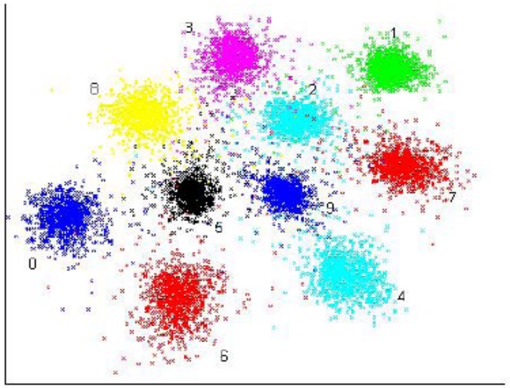
Embedding Results on MNIST Digits (dG-NCA)



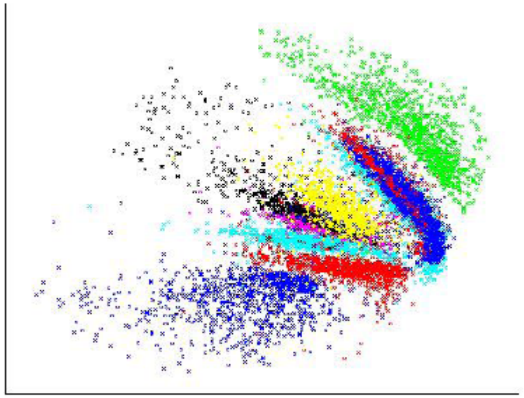
Embedding Results on MNIST Digits (dt-NCA)



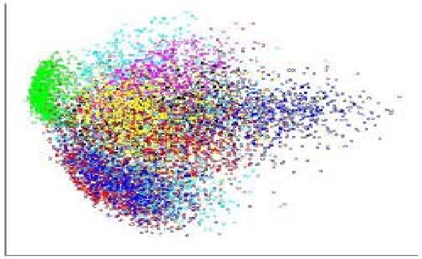
Embedding Results on MNIST Digits (Other Methods)



Two-dimensional embedding of 10,000 MNIST test data using the Deep Neural Network kNN classifier (DNet-kNN).



Two-dimensional embedding of 10,000 MNIST test data using the Deep Autoencoder (DA).



Two-dimensional embedding of 10,000 MNIST test data using PCA.