Aalto University
School of Science

Multiplicative Update For Fast Optimization of Information Retrieval Based Neighbor Embedding

Jaakko Peltonen, Ziyuan Lin

Department of Information and Computer Science
Aalto University

September 22, 2013

Outline

Neighbor retrieval visualizer

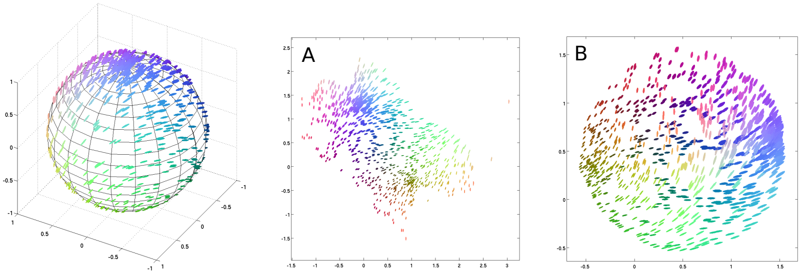
How to make it fast

Experiments

Summary

1. *Neighbor Retrieval Visualizer* (NeRV) is the first non-linear dimensional reduction method that formulates dimensional reduction as an information retrieval task. It directly optimizes information retrieval measures which makes it outperform other non-linear dimension reduction methods;
2. We propose a multiplicative update rule for the NeRV which makes it faster and preserve the good quality as the original NeRV method;

An Example of NeRV



A 3D sphere "Tearing-open" "Squashing"

Figure : Two ways showing 2D visualizations of a sphere: **A**) to tear it open, which emphasizes precision; or **B**) to squash it flat, which emphasizes recall.

Errors in visual information retrieval

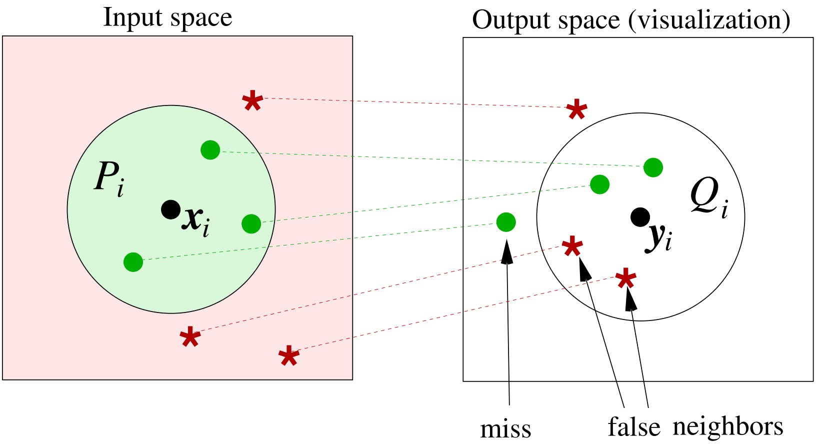


Figure : *Misses* are true neighbors that are not neighbors on the display; *false neighbors* are neighbors on the display that are not true neighbors.

Technical detail of NeRV

The resulting total cost of errors turns out to be KL-divergences between neighborhood distributions (Venna et al. 2010):

$$\{y_{ij}^*\} = \arg\min_{\{y_{ij}\}} \sum_i \sum_{j \neq i} \lambda p_{ij} \log \frac{p_{ij}}{q_{ij}} + (1-\lambda) q_{ij} \log \frac{q_{ij}}{p_{ij}}$$

$$\triangleq \arg\min_{\{y_{ij}\}} \sum_i \lambda KL(P_i \| Q_i) + (1-\lambda) KL(Q_i \| P_i)$$

where

$$p_{ij} = \frac{\exp(-d(x_i, x_j)^2/(2\sigma_i^2))}{\sum_{k \neq i} \exp(-d(x_i, x_k)^2)/(2\sigma_i^2)}, \quad q_{ij} = \frac{\exp(-\|y_i - y_j\|^2/(2\sigma_i^2))}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)/(2\sigma_i^2)}$$

$$KL(P_i \| Q_i) \propto \ \#\text{misses}, \ KL(Q_i \| P_i) \propto \ \#\text{false neighbors}$$

NeRV is slow

NeRV is based on the traditional gradient descent method, which uses the additive update rule that needs a line search for the step size α:

$$\mathbf{z}_{id}^{(t+1)} = \mathbf{z}_{id}^{(t)} - \alpha_{id} \nabla_{id} \mathcal{J}\left(\mathbf{z}^{(t)}\right) ,$$

where

$$\begin{aligned} z_{id}^{(t)} &: \text{the } (i,d)\text{-th parameter at the } t\text{-th iteration;} \\ \mathcal{J} &: \text{the cost function;} \\ \nabla_{id} &: \text{the } (i,d)\text{-th component of the gradient;} \end{aligned}$$

NeRV is slow

- This additive rule is widely used, but it is slow in cases that need fast response. For example the user interaction application (Peltonen, Sandholm, and Kaski 2013) of NeRV that we have explored recently;
- The Quad-tree based accelerations (Yang, Peltonen, and Kaski 2013 or Maaten 2013) are comparable methods, but they have to assume the neighborhood distribution to be sparse, while what we present here has no such requirements.

Our method

To avoid the tuning or line search in the additive update, we derive the *multiplicative* update rule as below. The work in (Yang, Wang, and Oja 2010) is a special case for t-SNE.

$$\mathbf{z}_{id}^{(t+1)} = \mathbf{z}_{id}^{(t)} \frac{\nabla_{id}^{-} \mathcal{J}\left(\mathbf{z}^{(t)}\right)}{\nabla_{id}^{+} \mathcal{J}\left(\mathbf{z}^{(t)}\right)} ,$$

where ∇_{id}^{+} and ∇_{id}^{-} is a decomposition of the gradient that satisfies

$$\begin{aligned}
\nabla_{id}\mathcal{J} &= \nabla_{id}^{+}\mathcal{J} - \nabla_{id}^{-}\mathcal{J} \\
\nabla_{id}^{+}\mathcal{J} &\geq 0 \\
\nabla_{id}^{-}\mathcal{J} &\geq 0
\end{aligned}$$

Connections with additive update

1. They have the same extreme points:

$$
\begin{aligned}
z_{id}^{(t+1)} &= z_{id}^{(t)} \text{ (in additive update)} \\
\Longrightarrow & \nabla_{id}\mathcal{J}(z_{id}) = 0 \\
\Longrightarrow & \nabla_{id}^{+}\mathcal{J}(z_{id}) = \nabla_{id}^{-}\mathcal{J}(z_{id}) \\
\Longrightarrow & z_{id}^{(t+1)} = z_{id}^{(t)} \text{ (in multiplicative update)}
\end{aligned}
$$

2.

$$
\left.
\begin{aligned}
z_{id}^{(t+1)} =& z_{id}^{(t)} - \alpha_{id}\nabla_{id}\mathcal{J}(z^{(t)}) \\
\alpha_{id} =& \frac{z_{id}^{(t)}}{\nabla_{id}^{+}\mathcal{J}(z_{id}^{(t)})}
\end{aligned}
\right\} \Longrightarrow z_{id}^{(t+1)} = z_{id}^{(t)} \frac{\nabla_{id}^{-}\mathcal{J}(z^{(t)})}{\nabla_{id}^{+}\mathcal{J}(z^{(t)})}
$$

The pipeline of multiplicative NeRV

1. The multiplicative update rule is sign preserving. So we transform the coordinates with $z_{id}^{(0)} = \exp(y_{id}^{(0)})$;
2. Update multiplicatively

$$z_{id}^{(t+1)} = z_{id}^{(t)} \frac{\nabla_{id}^{-} \mathcal{J}(z^{(t)})}{\nabla_{id}^{+} \mathcal{J}(z^{(t)})}$$

   for a sufficient number of iterations. In our experiments, 300 iterations works empirically well;
3. Transform back to the display space:

$$y_{id}^{*} = \log z_{id}^{*}$$

Information retrieval perspective interpretation

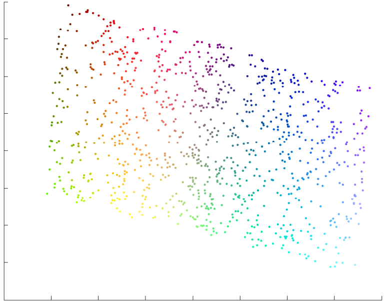
- The multiplicative rule is that it is intuitive understandable, because terms in update rule have information retrieval interpretation;
- It pushes the coordinates y_{id} away from the worst false neighbors of i and towards missed neighbors;
- More details are available in the paper.

Experiments

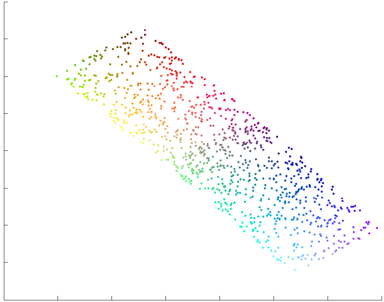
Our experiments show three things:

1. Our update rule produces visually similar results as the original NeRV;
2. It improves much faster;
3. In quantitative comparison, it outperforms many earlier methods.

Experiments: results on Plain S-curve



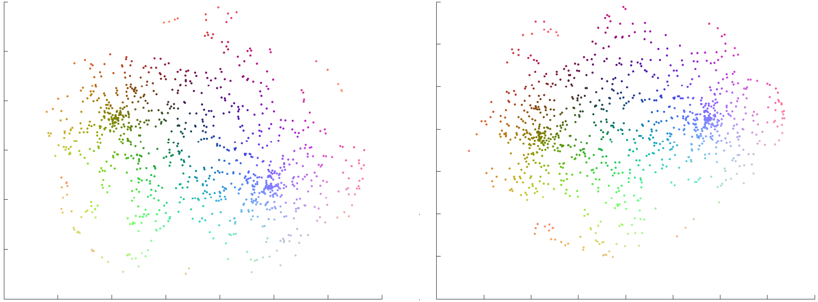
Multiplicative



Additive

Figure : Colors correspond to original 3D coordinates.

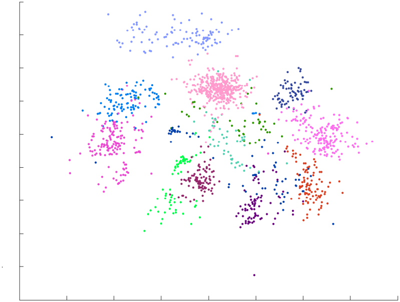
Experiments: results on Sphere



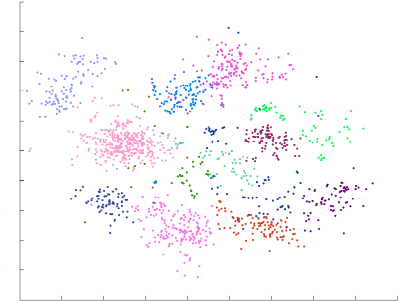
Multiplicative Additive

Figure : Colors correspond to original 3D coordinates.

Experiments: results on Phoneme



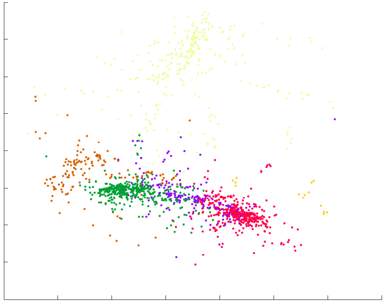
Multiplicative



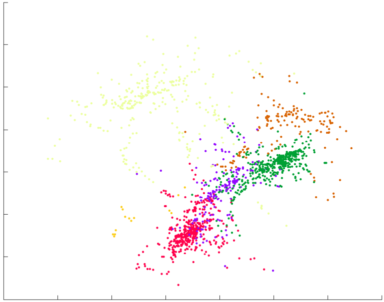
Additive

Figure : A data set of phonemes (very short audio samples for speech). Colors correspond to different phonemes.

Experiments: results on Landsat



Multiplicative



Additive

Figure : A data set of images of land taken by satellite. Colors correspond to terrain types.

Experiments: results on MNIST

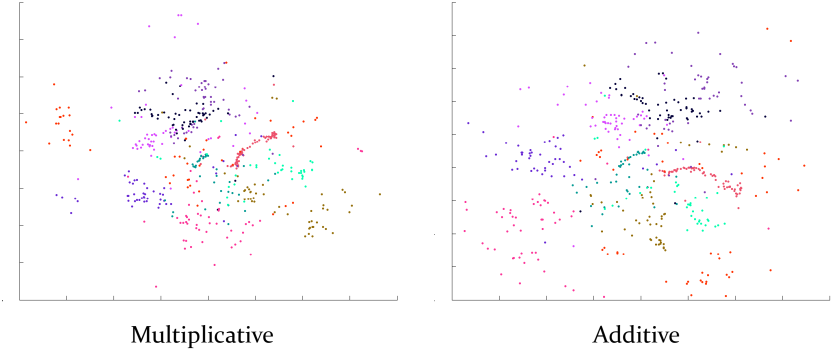
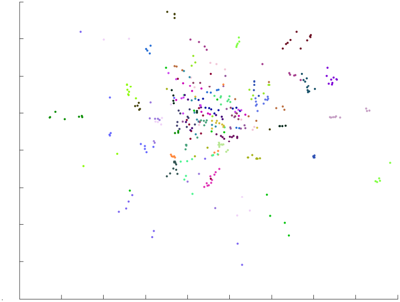
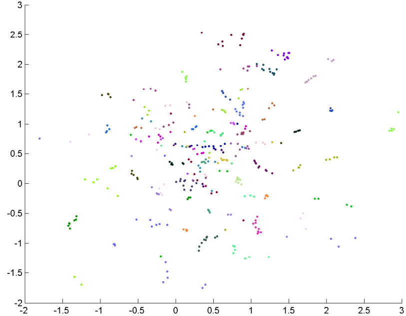


Figure : A data set for hand-written digits. Colors correspond to different digits.

Experiments: results on Olivetti Faces



Multiplicative



Additive

Figure : A data set of faces. Colors correspond to different people.

Experiments: performance

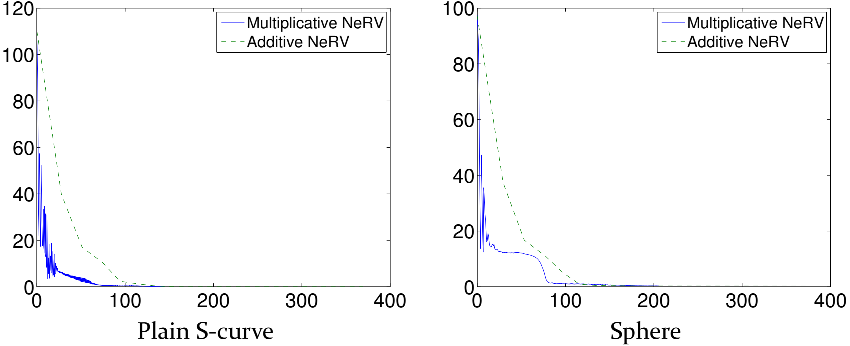


Figure : Graphs of running time vs. information retrieval performance (NeRV's cost, the lower the better).

Experiments: performance

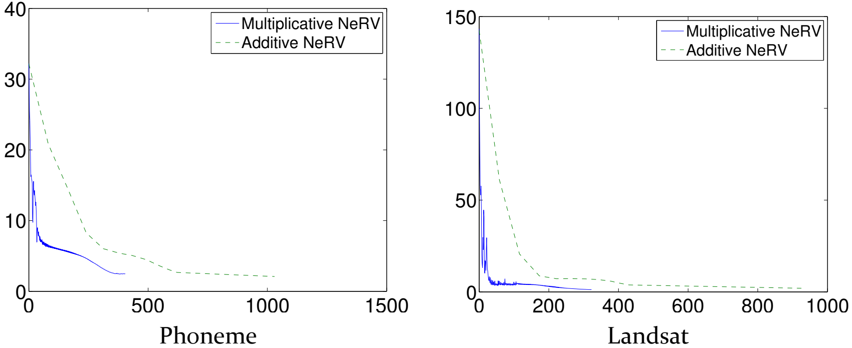


Figure : Graphs of running time vs. information retrieval performance (NeRV's cost, the lower the better).

Experiments: performance

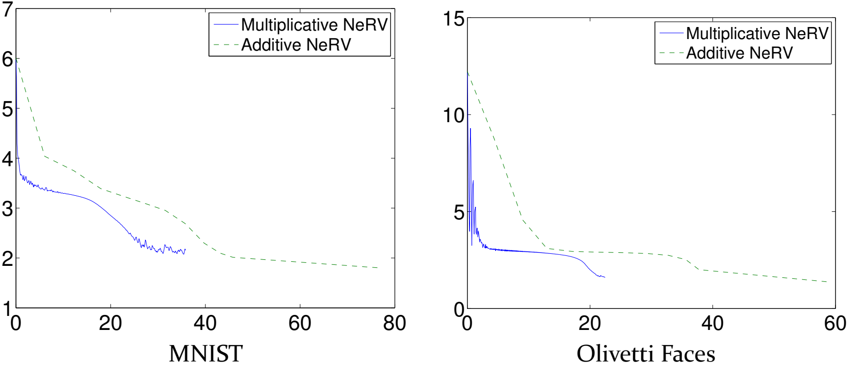


Figure : Graphs of running time vs. information retrieval performance (NeRV's cost, the lower the better).

Experiments: comparison with other NLDR methods

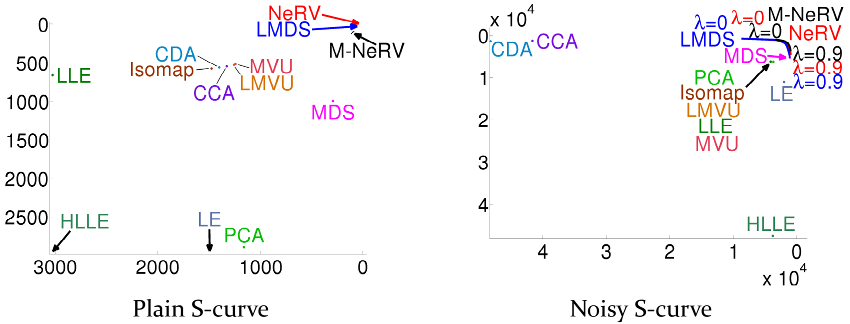
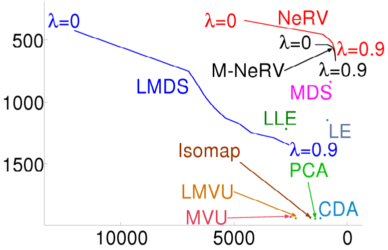
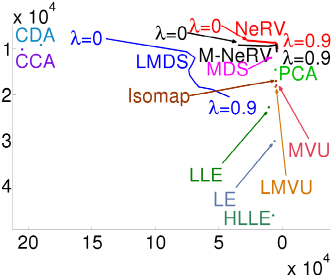


Figure : Mean smoothed recall vs. mean smoothed precision for different methods. Best methods are located top-right.

Experiments: comparison with other NLDR methods



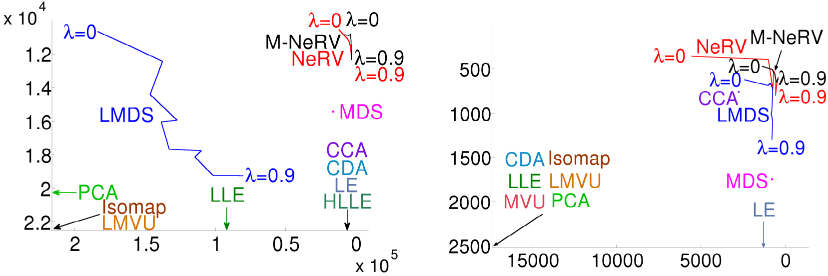
Olivetti Faces



Mouse Gene Expression

Figure : Mean smoothed recall vs. mean smoothed precision for different methods. Best methods are located top-right. Mouse Gene Expression data set: gene expression profiles from different mouse tissues.

Experiments: comparison with other NLDR methods



Gene Expression Compendium   Sea-water Temperature Time Series

Figure : Gene Expression Compendium data set: human gene expression arrays; Sea-water Temperature Time Teries data set: a time series of weekly temperature measurements of sea water over years.

Conclusion

1. NeRV is a method that is already known to perform very well in experiments of comparison with other methods. In this paper we address the speed of learning;
2. We introduced a multiplicative update rule for the information retrieval based visualization method Neighbor Retrieval Visualizer (NeRV);
3. It needs no user-assigned learning rate parameter or line search, but yields strong speedup over original NeRV and maintains state of the art performance, in terms of qualitative appearance of visualizations and quantitative information retrieval performance measures.