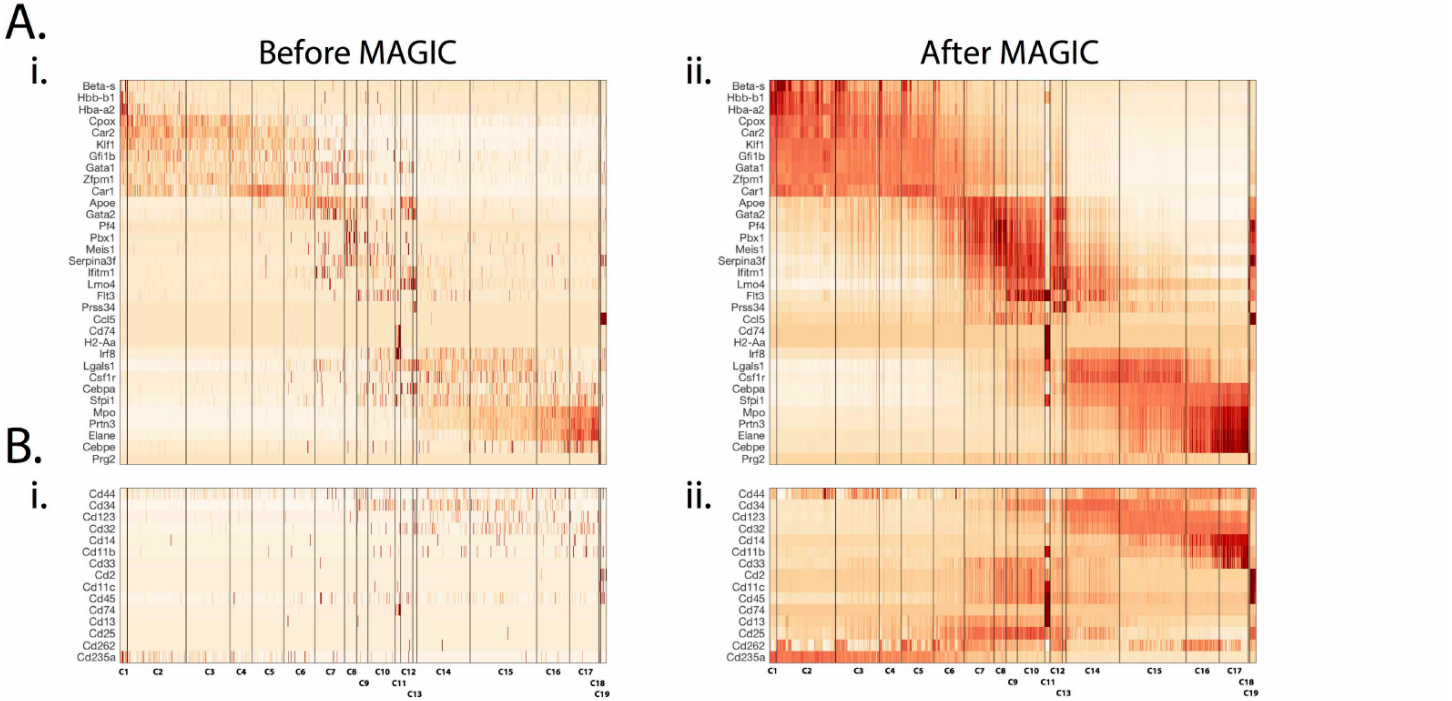
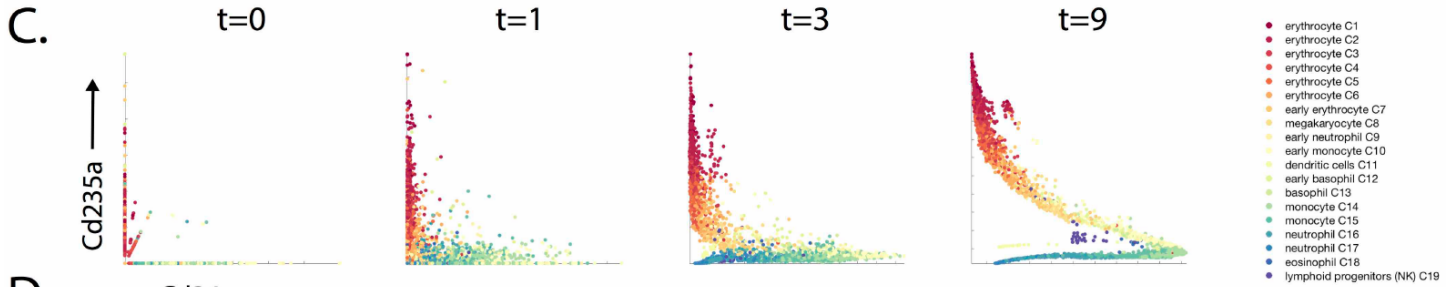
Imputation reveals gene-gene correlation patterns





A: model-based imputation

bayNorm	[47]	Binomial model, empirical Bayes prior
BISCUIT	[48]	Gaussian model of log counts, cell- and cluster-specific parameters
CIDR	[49]	Decreasing logistic model (DO), non-linear least-squares regression (imp)
SAVER	[50]	NB model, Poisson LASSO regression prior
ScImpute	[51]	Mixture model (DO), non-negative least squares regression (imp)
scRecover	[52]	ZINB model (DO identification only)
VIPER	[53]	Sparse non-negative regression model

B: data smoothing

DrImpute	[54]	*k*-means clustering of PCs of correlation matrix
knn-smooth	[55]	*k*-nearest neighbor smoothing
LSImpute	[56]	Locality sensitive imputation
MAGIC	[57]	Diffusion across nearest neighbor graph
netSmooth	[58]	Diffusion across PPI network

C: data reconstruction, matrix factorizatio

ALRA	[59]	SVD with adaptive thresholding
ENHANCE	[60]	Denoising PCA with aggregation step
scRMD	[61]	Robust matrix decomposition
consensus NMF	[62]	Meta-analysis approach to NMF
f-scLVM	[63]	Sparse Bayesian latent variable model
GPLVM	[64]	Gaussian process latent variable model
pCMF	[65]	Probab. count matrix factorization with Poisson model
scCoGAPS	[66]	Extension of NMF
SDA	[67]	Sparse decomposition of arrays (Bayesian)
ZIFA	[68]	ZI factor analysis
ZINB-WaVE	[69]	ZINB factor model

C: data reconstruction, machine learning

AutoImpute	[70]	AE, no error back-propagation for zero counts
BERMUDA	[71]	AE for cluster batch correction (MMD and MSE loss function)
DeepImpute	[72]	AE, parallelized on gene subsets
DCA	[73]	Deep count AE (ZINB / NB model)
DUSC / DAWN	[74]	Denoising AE (PCA determines hidden layer size)
EnImpute	[75]	Ensemble learning consensus of other tools
Expression Saliency	[76]	AE (Poisson negative log-likelihood loss function)
LATE	[77]	Non-zero value AE (MSE loss function)
Lin_DAE	[78]	Denoising AE (imputation across *k*-nearest neighbor genes)
SAUCIE	[79]	AE (MMD loss function)
scScope	[80]	Iterative AE
scVAE	[81]	Gaussian-mixture VAE (NB / ZINB / ZIP model)
scVI	[82]	VAE (ZINB model)
scvis	[83]	VAE (objective function based on latent variable model and t-SNE)
VASC	[84]	VAE (denoising layer; ZI layer, double-exponential and Gumbel distribution)
Zhang_VAE	[85]	VAE (MMD loss function)

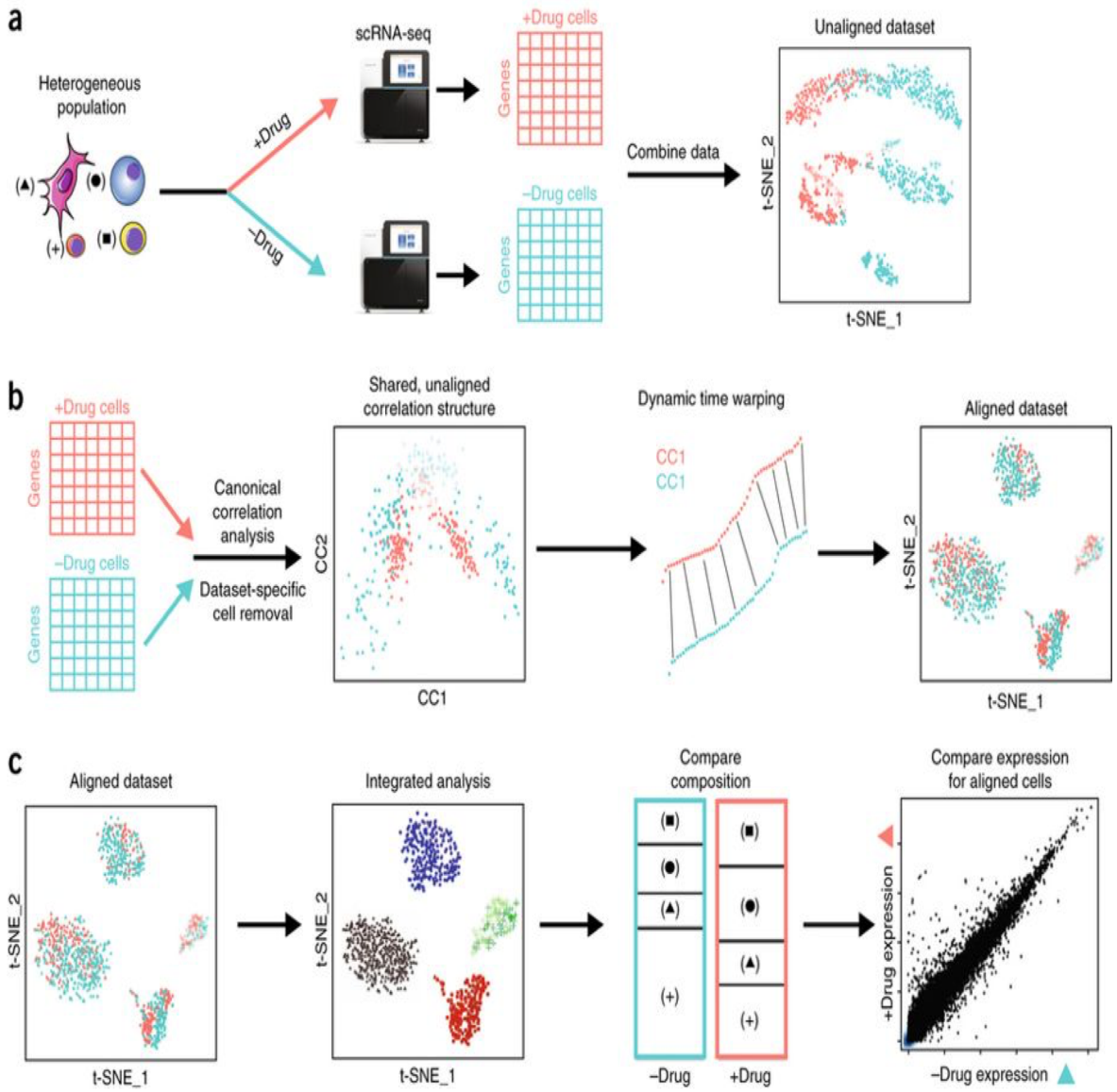
T: using external information

ADImpute	[86]	Gene regulatory network information
netSmooth	[58]	PPI network information
SAVER-X	[87]	Transfer learning with atlas-type resources
SCRABBLE	[88]	Matched bulk RNA-seq data
TRANSLATE	[77]	Transfer learning with atlas-type resources
URSM	[89]	Matched bulk RNA-seq data

47. Tang W, Bertaux F, Thomas P, Stefanelli C, Saint M, et al. bayNorm: Bayesian gene expression recovery, imputation and normalisation for single cell RNA-sequencing data. bioRxiv. 2018. https://www.biorxiv.org/content/10.1101/384586v2.abstract.
48. Azizi E, Prabhakaran S, Carr A, Pe'er D. Bayesian inference for single-cell clustering and imputing. Genomics Comput Biol. 2017;3(1):46. https://doi.org/10.18547/gcb.2017.vol3.iss1.e46. Accessed 27 Mar 2019.
49. Lin P, Troup M, Ho JWK. CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. Genome Biol. 2017;18(1):59. https://doi.org/10.1186/s13059-017-1188-0. Accessed 27 Mar 2019.
50. Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, Murray JI, Raj A, Li M, Zhang NR. SAVER: gene expression recovery for single-cell RNA sequencing. Nat Methods. 2018;15(7):539. https://doi.org/10.1038/s41592-018-0033-z. Accessed 27 Mar 2019.
51. Li WV, Li JJ. An accurate and robust imputation method scimpute for single-cell RNA-seq data. Nat Commun. 2018;9(1):997.
52. Miao Z, Li J, Zhang X. scRecover: discriminating true and false zeros in single-cell RNA-seq data for imputation. bioRxiv. 2019665323. https://doi.org/10.1101/665323. Accessed 15 Oct 2019.
53. Chen M, Zhou X. VIPER: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. Genome Biol. 2018;19(1):196.
54. Gong W, Kwak I-Y, Pota P, Koyano-Nakagawa N, Garry DJ. DrImpute: imputing dropout events in single cell RNA sequencing data. BMC Bioinformatics. 2018;19(1):220. https://doi.org/10.1186/s12859-018-2226-y. Accessed 27 Mar 2019.
55. Wagner F, Yan Y, Yanai I. K-nearest neighbor smoothing for high-throughput single-cell RNA-Seq data. bioRxiv. 2018217737. https://doi.org/10.1101/217737. Accessed 15 Oct 2019.
56. Moussa M, Măndoiu II. Locality sensitive imputation for single cell RNA-Seq data. J Comput Biol. 2019. https://doi.org/10.1089/cmb.2018.0236. Accessed 27 July 2019.
57. Dijk DV, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, Burdziak C, Moon KR, Chaffer CL, Pattabiraman D, Bierie B, Mazutis L, Wolf G, Krishnaswamy S, Pe'er D. Recovering gene interactions from single-cell data using data diffusion. Cell. 2018;174(3):716–72927. https://doi.org/10.1016/j.cell.2018.05.061. Accessed 27 Mar 2019.
58. Jonathan Rohen AA. netsmooth: network-smoothing based imputation for single cell RNA-seq. F1000Res. 2018;7:. https://github.com/BIMSBbioinfo/netSmooth.
59. Linderman GC, Zhao J, Kluger Y. Zero-preserving imputation of scRNA-seq data using low-rank approximation. bioRxiv. 2018. https://www.biorxiv.org/content/10.1101/397588v1.abstract.
60. Wagner F, Barkley D, Yanai I. Accurate denoising of single-cell RNA-Seq data using unbiased principal component analysis. bioRxiv. 2019655365. URL https://doi.org/10.1101/655365. Accessed 15 Nov 2019.
61. Chen C, Wu C, Wu L, Wang Y, Deng M, Xi R. scRMD: imputation for single cell RNA-seq data via robust matrix decomposition. bioRxiv. 2018459404. https://doi.org/10.1101/459404. Accessed 15 Oct 2019.
62. Kotliar D, Veres A, Nagy MA, Tabrizi S, Hodis E, Melton DA, Sabeti PC. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. Elife. 2019;8:43803.
63. Buettner F, Pratanwanich N, McCarthy DJ, Marioni JC, Stegle O. f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. Genome Biol. 2017;18(1):212. https://doi.org/10.1186/s13059-017-1334-8.
64. Verma A, Engelhardt BE. A robust nonlinear low-dimensional manifold for single cell RNA-seq data. bioRxiv. 2018443044. https://doi.org/10.1101/443044. Accessed 15 Nov 2019.
65. Durif G, Modolo L, Mold JE, Lambert-Lacroix S, Picard F. Probabilistic count matrix factorization for single cell expression data analysis. Bioinformatics. 2019. https://doi.org/10.1093/bioinformatics/btz177.
66. Stein-O'Brien GL, Clark BS, Sherman T, Zibetti C, Hu Q, Sealfon R, Liu S, Qian J, Colantuoni C, Blackshaw S, Goff LA, Fertig EJ. Decomposing cell identity for transfer learning across cellular measurements, platforms, tissues, and species. Cell Syst. 2019;8(5):395–4118. https://doi.org/10.1016/j.cels.2019.04.004.
67. Jung M, Wells D, Rusch J, Ahmad S, Marchini J, Myers SR, Conrad DF. Unified single-cell analysis of testis gene regulation and pathology in five mouse strains. eLife. 2019;8:. URL https://doi.org/10.7554/eLife.43966.
68. Pierson E, Yau C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. 2015;16(1):241. https://doi.org/10.1186/s13059-015-0805-z. Accessed 27 Mar 2019.
69. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert J-P. A general and flexible method for signal extraction from single-cell RNA-seq data. Nature Commun. 2018;9(1):284. https://doi.org/10.1038/s41467-017-02554-5.
70. Talwar D, Mongia A, Sengupta D, Majumdar A. AutoImpute: autoencoder based imputation of single-cell RNA-seq data. Sci Rep. 2018;8(1):16329. https://doi.org/10.1038/s41598-018-34688-x.
71. Wang T, Johnson TS, Shao W, Lu Z, Helm BR, Zhang J, Huang K. BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes. Genome Biol. 2019;20(1):165. https://doi.org/10.1186/s13059-019-1764-6. Accessed 15 Nov 2019.
72. Arisdakessian C, Poirion O, Yunits B, Zhu X, Garmire L. DeepImpute: an accurate, fast and scalable deep neural network method to impute single-cell RNA-Seq data. bioRxiv. 2018. https://www.biorxiv.org/content/10.1101/353607v1.abstract.
73. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun. 2019;10(1):390. https://doi.org/10.1038/s41467-018-07931-2. Accessed 27 Mar 2019.
74. Srinivasan S, Johnson NT, Korkin D. A hybrid deep clustering approach for robust cell type profiling using single-cell RNA-seq data. bioRxiv. 2019. https://www.biorxiv.org/content/10.1101/511626v1.abstract.
75. Zhang X-F, Ou-Yang L, Yang S, Zhao X-M, Hu X, Yan H. EnImpute: imputing dropout events in single cell RNA sequencing data via ensemble learning. Bioinformatics. 2019. https://doi.org/10.1093/bioinformatics/btz435.
76. Kinalis S, Nielsen FC, Winther O, Bagger FO. Deconvolution of autoencoders to learn biological regulatory modules from single cell mRNA sequencing data. BMC Bioinformatics. 2019;20(1):379. https://doi.org/10.1186/s12859-019-2952-9.
77. Badsha MB, Li R, Liu B, Li YI, Xian M, Banovich NE, Fu AQ. Imputation of single-cell gene expression with an autoencoder neural network. bioRxiv. 2018504977. https://doi.org/10.1101/504977. Accessed 15 Oct 2019.
78. Lin C, Jain S, Kim H, Bar-Joseph Z. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. Nucleic Acids Res. 2017;45(17):156.
79. Amodio M, Dijk DV, Srinivasan K, Chen WS, Mohsen H, Moon KR, Campbell A, Zhao Y, Wang X, Venkataswamy M, Desai A, Ravi V, Kumar P, Montgomery R, Wolf G, Krishnaswamy S. Exploring single-cell data with deep multitasking neural networks. bioRxiv. 2019237065. https://doi.org/10.1101/237065. Accessed 15 Oct 2019.
80. Deng Y, Bao F, Dai Q, Wu LF, Altschuler SJ. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. Nat Methods. 2019. https://doi.org/10.1038/s41592-019-0353-7.
81. Grønbech CH, Vording MF, Timshel P, Sønderby CK, Pers TH, Winther O. scVAE: Variational auto-encoders for single-cell gene expression data. bioRxiv. 2019318295. https://doi.org/10.1101/318295. Accessed 15 Oct 2019.
82. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15(12):1053–8. https://doi.org/10.1038/s41592-018-0229-2.
83. Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat Commun. 2018;9(1):2002.
84. Wang D, Gu J. VASC: Dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. Genomics Proteomics Bioinforma. 2018;16(5):320–31.
85. Zhang C. Single-cell data analysis using mmd variational autoencoder for a more informative latent representation. bioRxiv. 2019613414. https://doi.org/10.1101/613414. Accessed 15 Oct 2019.
86. Leote AC, Wu X, Beyer A. Network-based imputation of dropouts in single-cell RNA sequencing data. bioRxiv. 2019611517. URL https://doi.org/10.1101/611517. Accessed 23 Apr 2019.
87. Wang J, Agarwal D, Huang M, Hu G, Zhou Z, Ye C, Zhang NR. Data denoising with transfer learning in single-cell transcriptomics. Nat Methods. 2019;16(9):875–8. https://doi.org/10.1038/s41592-019-0537-1. Accessed 15 Oct 2019.
88. Peng T, Zhu Q, Yin P, Tan K. SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data. Genome Biol. 2019;20(1):88. https://doi.org/10.1186/s13059-019-1681-8.
89. Zhu L, Lei J, Devlin B, Roeder K. A unified statistical framework for single cell and bulk RNA sequencing data. Ann Appl Stat. 2018;12(1):609–32. https://doi.org/10.1214/17-AOAS1110. Accessed 15 Nov 2019.

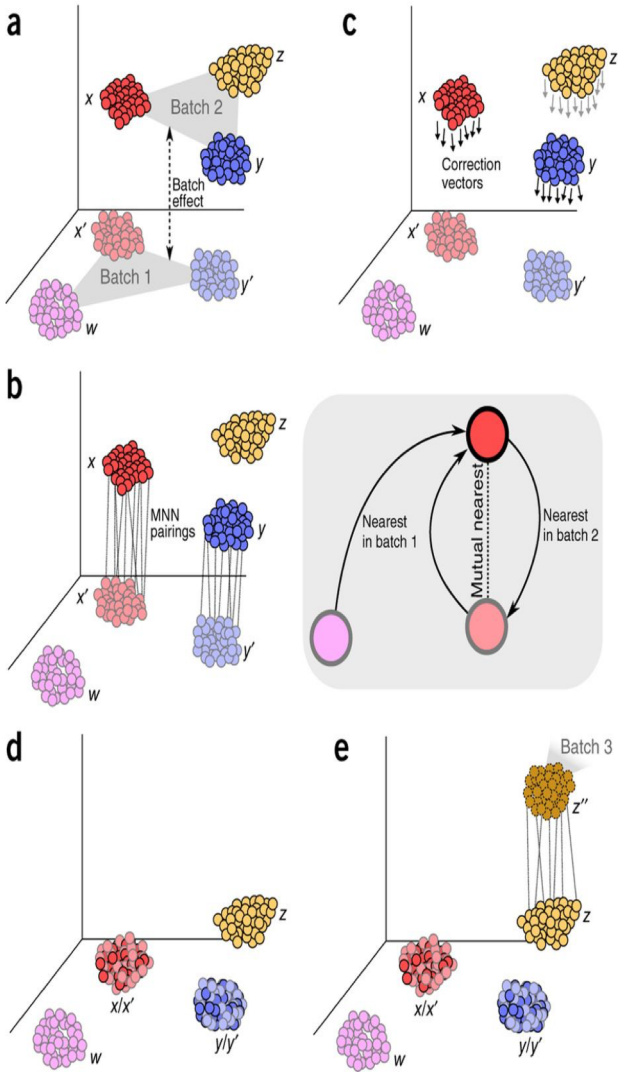
Integrating multiple single-cell datasets

Canonical Correlation Analysis (CCA)



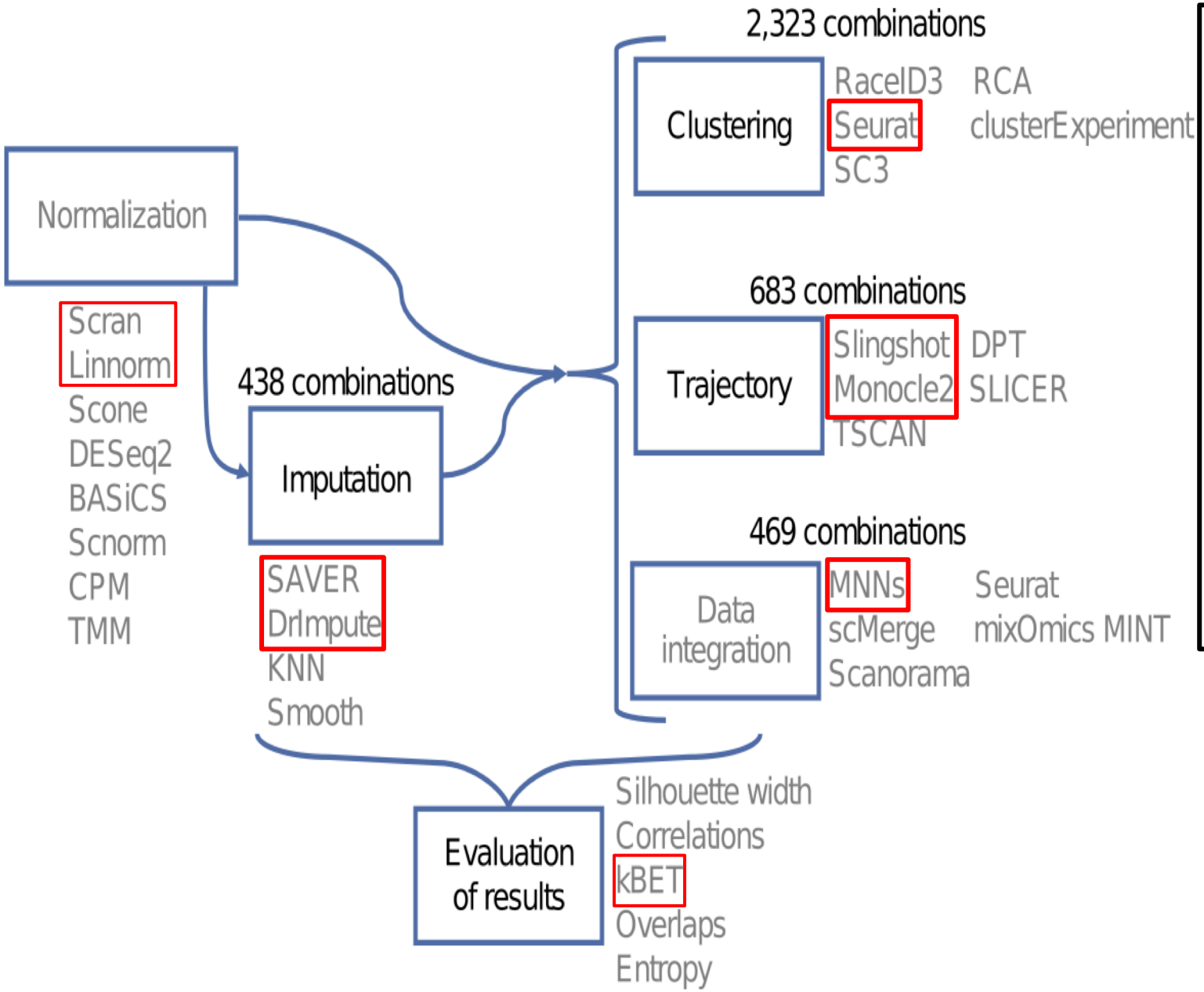
From: Butlet *et al.,* 2018

Mutual nearest neighbors (MNN) correction



From: Haghverdi *et al., 2018*

Summary and Method comparison



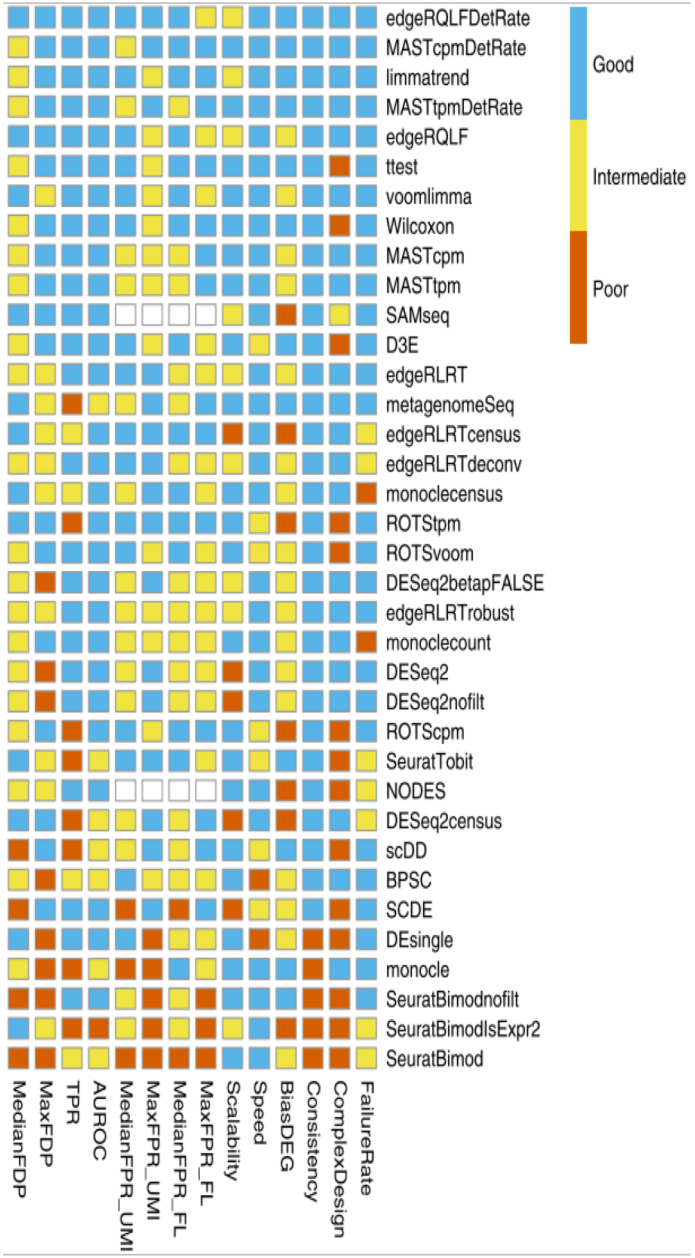
ACTIONet

Monocle3

Harmony

Recent Methods

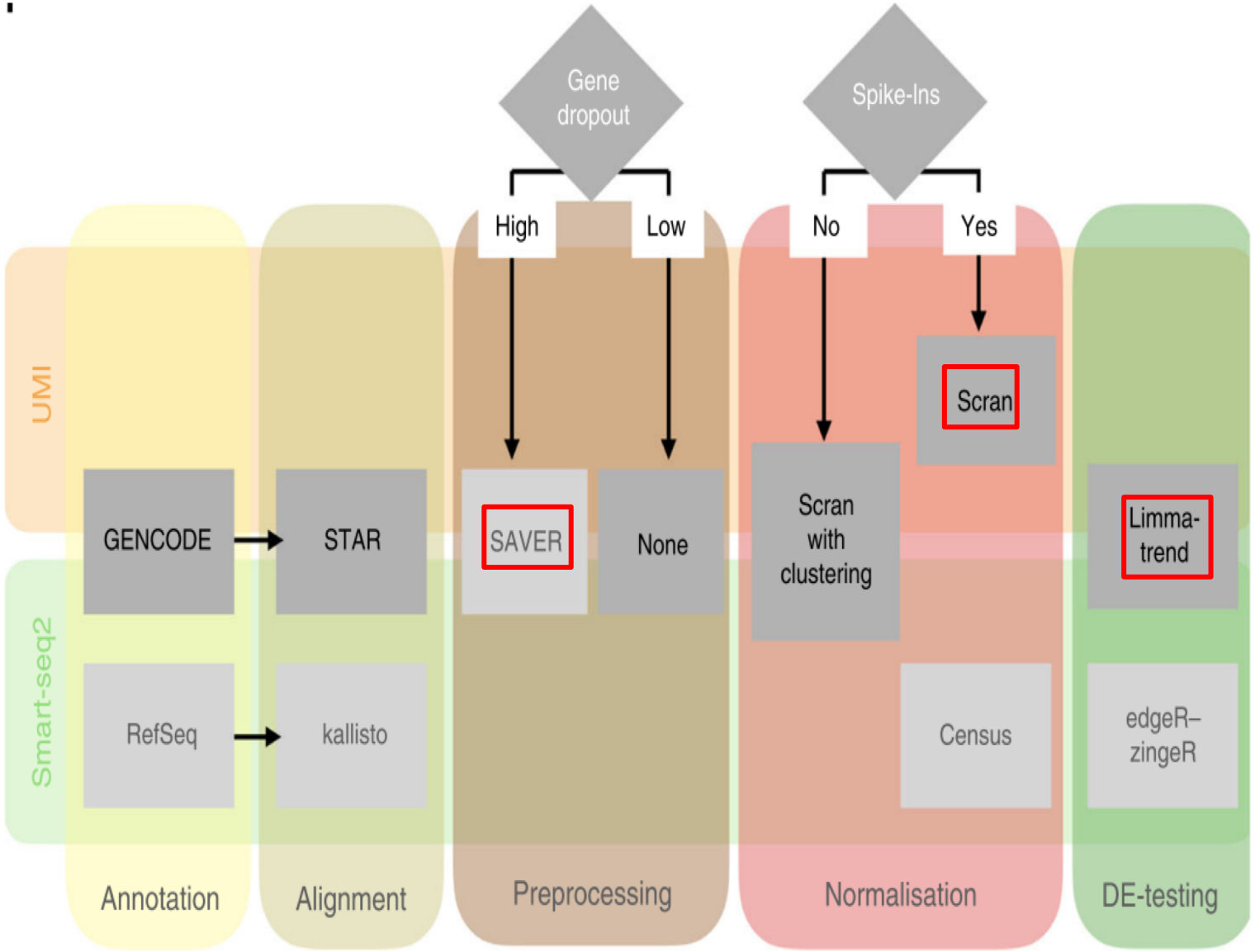
Tian, L., Dong, X., Freytag, S., Lê Cao, K.-A., Su, S., JalalAbadi, A., … Ritchie, M. E. (2019). Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. Nature Methods, 16(6), 479–487.



Comparison of differential expression methods

Found Limma-trend, MAST, edgeR, also t-test and Wilcoxon to perform well

Soneson, C., & Robinson, M. D. (2018). Bias, robustness and scalability in single-cell differential expression analysis. Nature Methods, 15(4), 255–261.



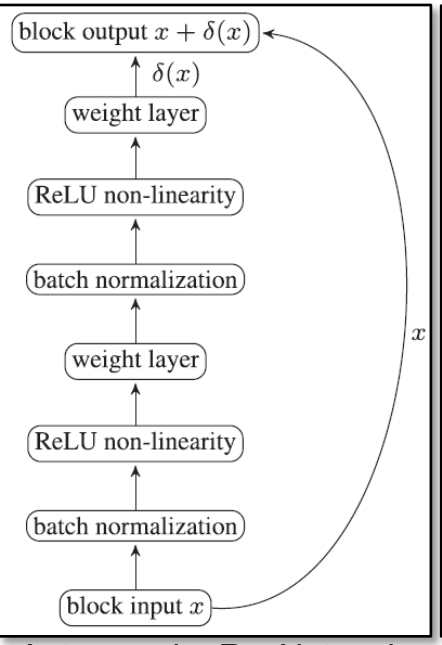
Vieth, B., Parekh, S., Ziegenhain, C., Enard, W., & Hellmann, I. (2019). A systematic evaluation of single cell RNA-seq analysis pipelines. Nature Communications, 10(1), 4667.

Summary

- Normalization
  - Scran and Linnorm
- Imputation
  - SAVER
- Batch-correction
  - [fast]MNN and Harmony
- Clustering
  - ACTIONet and Seurat
- Trajectory detection
  - Monocle3 and Slingshot
- Differential expression
  - Limma-trend

Deep Learning methods for scRNA-seq

MMD-ResNet: Autoencoder for batch correction



Autoencoder ResNet arch.

MMD (Gretton *et al.*, 2006, 2012) is a measure for distance between two probability distributions p, q. It is defined with respect to a function class $\mathcal{F}$ by

$$\mathrm{MMD}(\mathcal{F}, p, q) \equiv \sup_{f \in \mathcal{F}} (\mathbb{E}_{x \sim p} f(x) - \mathbb{E}_{x \sim q} f(x)).$$

When $\mathcal{F}$ is a reproducing kernel Hilbert space with kernel k, the MMD can be written as the distance between the mean embeddings of p and q

$$\mathrm{MMD}^2(\mathcal{F}, p, q) = ||\mu_p - \mu_q||^2_{\mathcal{F}}, \quad (1)$$

where $\mu_p(t) = \mathbb{E}_{x \sim p} k(x, t)$. Equation (1) can be written as

$$\mathrm{MMD}^2(\mathcal{F}, p, q) = \mathbb{E}_{x,x' \sim p} k(x, x') - 2\mathbb{E}_{x \sim p, y \sim q} k(x, y) + \mathbb{E}_{y,y' \sim q} k(y, y'), \quad (2)$$

where x and x' are independent, and so are y and y'. Importantly, if k is a universal kernel, then $\mathrm{MMD}(\mathcal{F}, p, q) = 0$ iff $p = q$. In practice, the distributions p, q are unknown, and instead we are given observations $X = \{x_1, \ldots x_n\}, Y = \{y_1, \ldots y_m\}$, so that the (biased) sample version of (2) becomes

$$\mathrm{MMD}^2(\mathcal{F}, X, Y) = \frac{1}{n^2} \sum_{x_i, x_j \in X} k(x_i, x_j)$$

$$- \frac{2}{nm} \sum_{x_i \in X, y_j \in Y} k(x_i, y_j)$$

$$+ \frac{1}{m^2} \sum_{y_i, y_j \in Y} k(y_i, y_j).$$

Maximum Mean Discrepancy (MMD) loss function

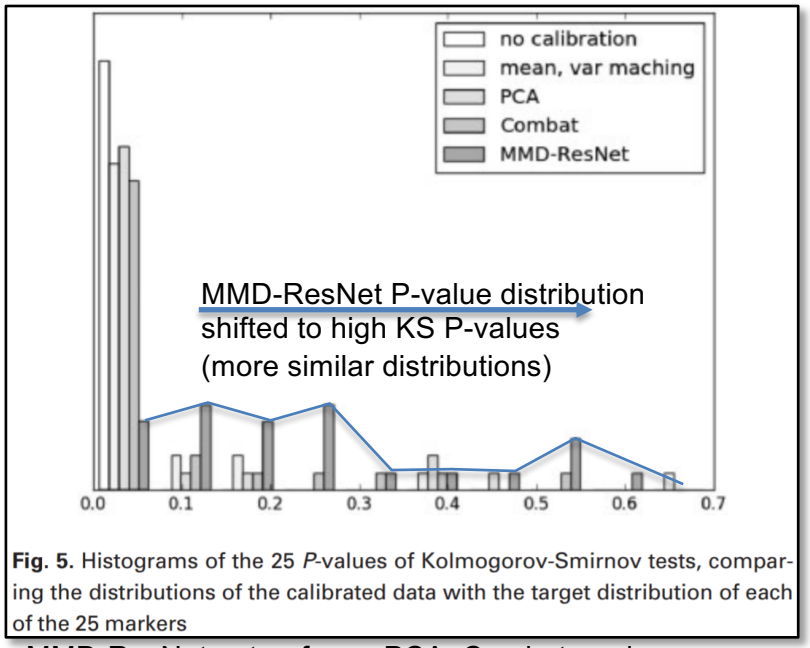


Fig. 5. Histograms of the 25 *P*-values of Kolmogorov-Smirnov tests, comparing the distributions of the calibrated data with the target distribution of each of the 25 markers

MMD-ResNet outperforms PCA, Combat, and

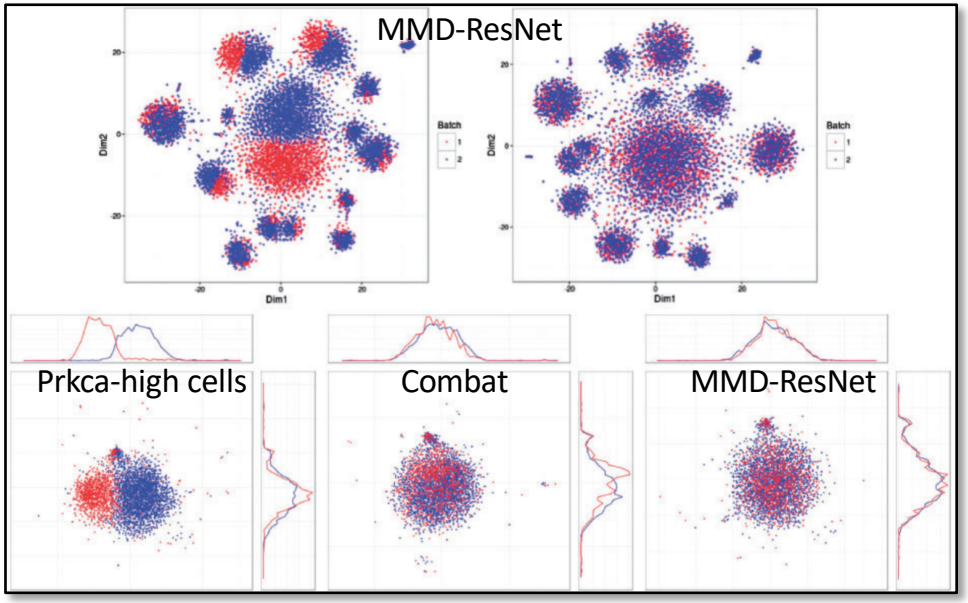
$$L(w) = \sqrt{\mathrm{MMD}^2(\{\widehat{\psi}(x_1), \ldots \widehat{\psi}(x_n)\}, \{y_1, \ldots y_m\})},$$

Train ResNet with loss MMD score function

Removal of batch effects using distribution-matching residual networks

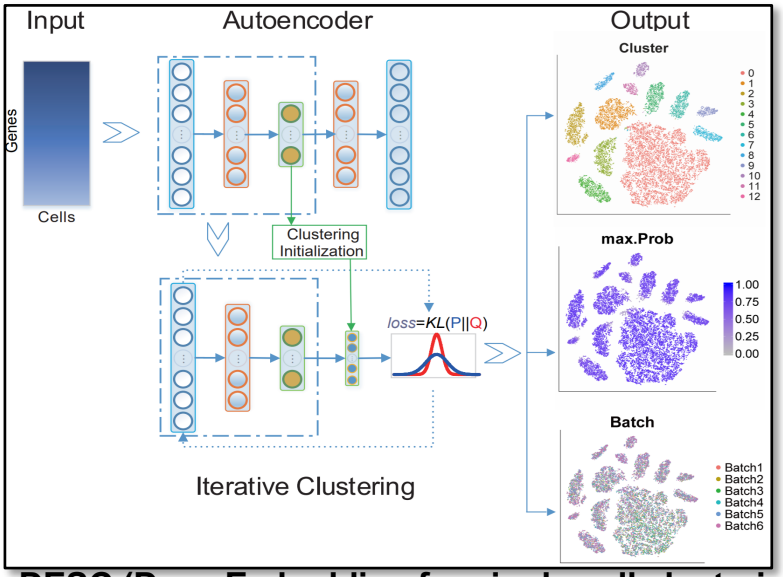
Uri Shaham[1,†], Kelly P. Stanton[2,3,†], Jun Zhao[3], Huamin Li[4], Khadir Raddassi[5], Ruth Montgomery[6] and Yuval Kluger[2,3,4,*]

Shaham et al., Bioinf, 2017



t-SNE plots before (left) and after (right) calibration

DESC: Deep embedding for cell-type-specific batch correction



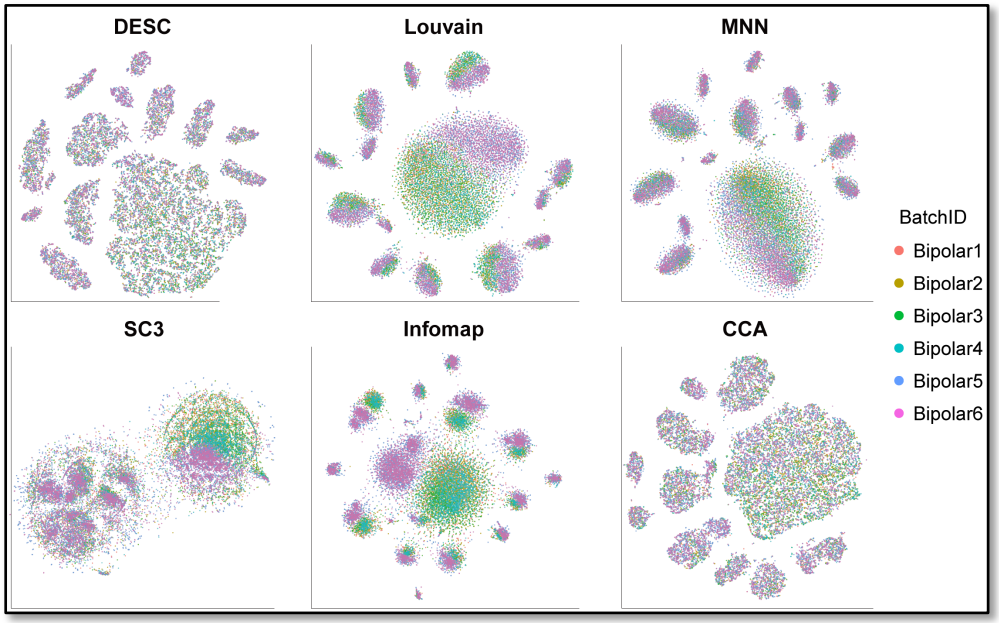
DESC (Deep Embedding for single-cell clustering):

- Stacked auto-encoder learns cluster-specific gene expression representation and cluster assignments for scRNA-seq data clustering
- Initialize clustering obtained from autoencoder
- Learn non-linear mapping from original space to a low-dimensional space
- iteratively optimize clustering objective function
  - Move each cell to nearest cluster
  - balance biological and technical differences between clusters
  - reduce influence of batch effect
- Enables soft clustering by assigning cluster-specific probabilities to each cell
- Facilitates clustering of cells with high confidence

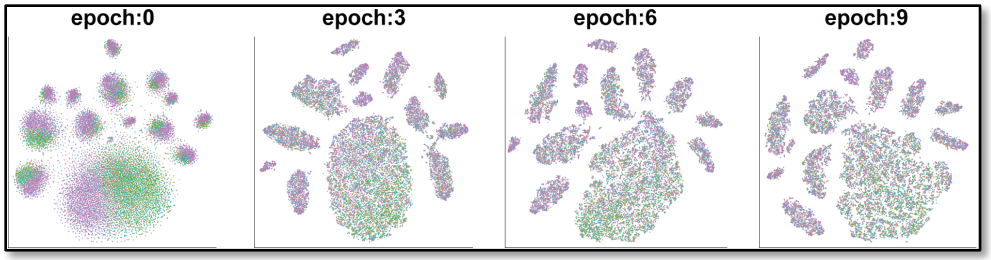
Deep learning enables accurate clustering and batch effect removal in single-cell RNA-seq analysis

Xiangjie Li, Yafei Lyu, Jihwan Park, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Gang Hu, Mingyao Li

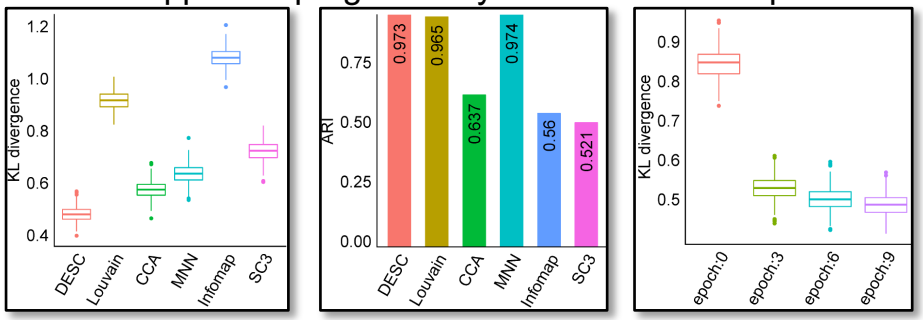
doi: https://doi.org/10.1101/530378



DESC avoids cluster-specific batch effects found in other methods

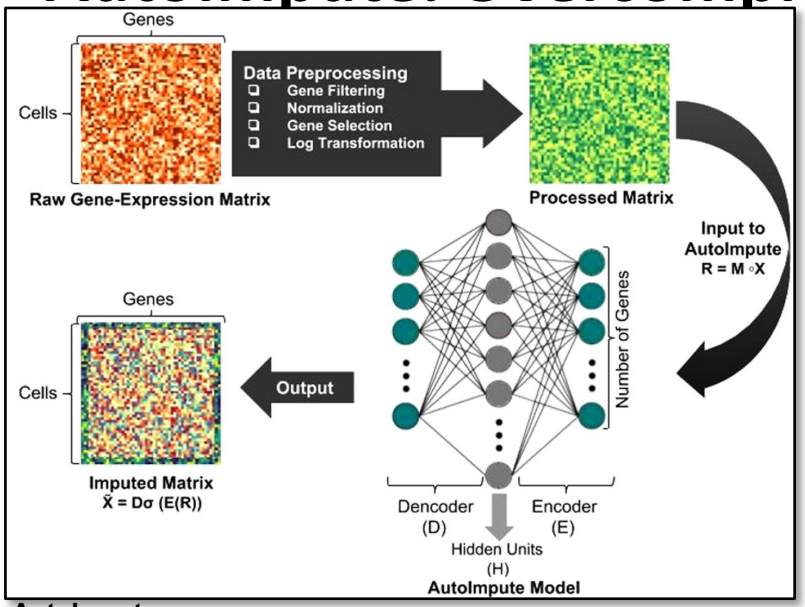


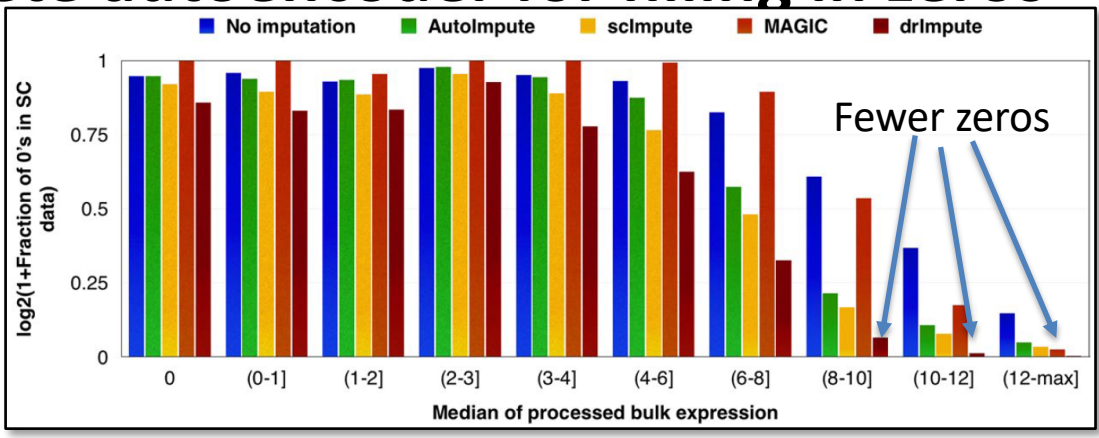
Iterative approach progressively removes cluster-specific batch effects



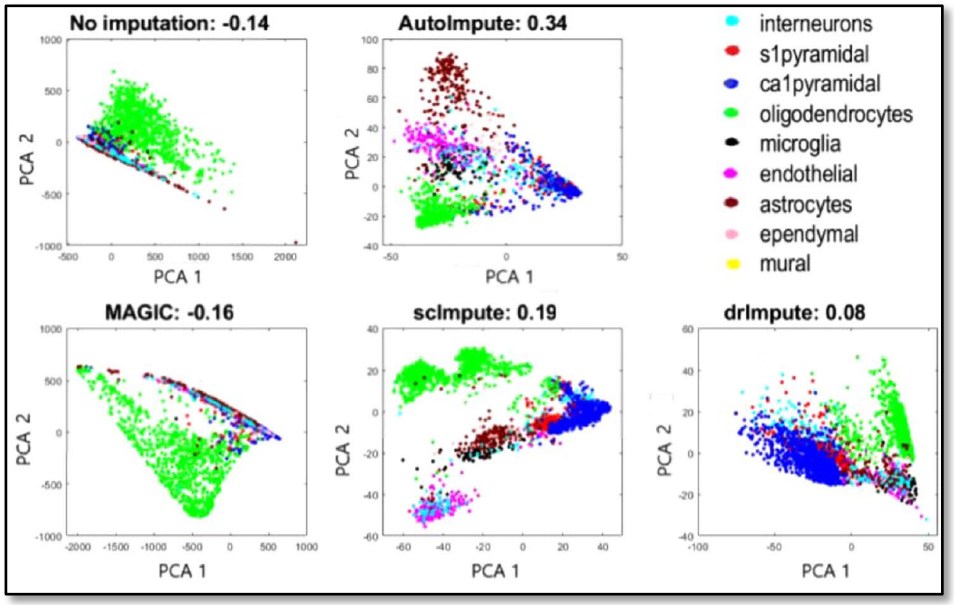
Rand Index (RI) = measure of the similarity between two data clusterings
ARI = Adjusted Rand Index, adjusted for the chance grouping of elements

AutoImpute: Overcomplete autoencoder for filling in zeros





Autoimpute captures more non-zero values for highly-expressed genes



Iterative approach progressively removes cluster-specific batch effects

AutoImpute

Filter raw gene expression data for bad genes
(normalize by library size, prune by gene-selection, log transform)
Feed processed matrix to AutoImpute model

- learn expression data representation
- reconstruct imputed matrix

Use overcomplete autoencoders to capture distribution of sparse gene expression data, and regenerate complete version of it

- Feeding sparse gene expression matrix as input to autoencoder
- train it to learn the encoder and decoder functions that best regenerate imputed expression with no dropouts
- back-propagating errors only for non-zero counts in sparse matrix

Training and Hyper-parameter Selection. The autoencoder network consists of a fully-connected multi-layer perceptron (MLP), with three layers: input, hidden and the output layer. It is trained using gradient descent with gradients computed by back-propagation to reach the minimum of the cost function (equation 8). RMSProp Optimizer was used to adjust the learning rate, such that, we avoid getting stuck at local minima and reach the minimum of the cost function faster. Both E - encoder matrix and D - decoder matrix were initialized from a random normal distribution.

The hyper-parameter selection was done after doing an extensive grid search on the following hyper-parameters:

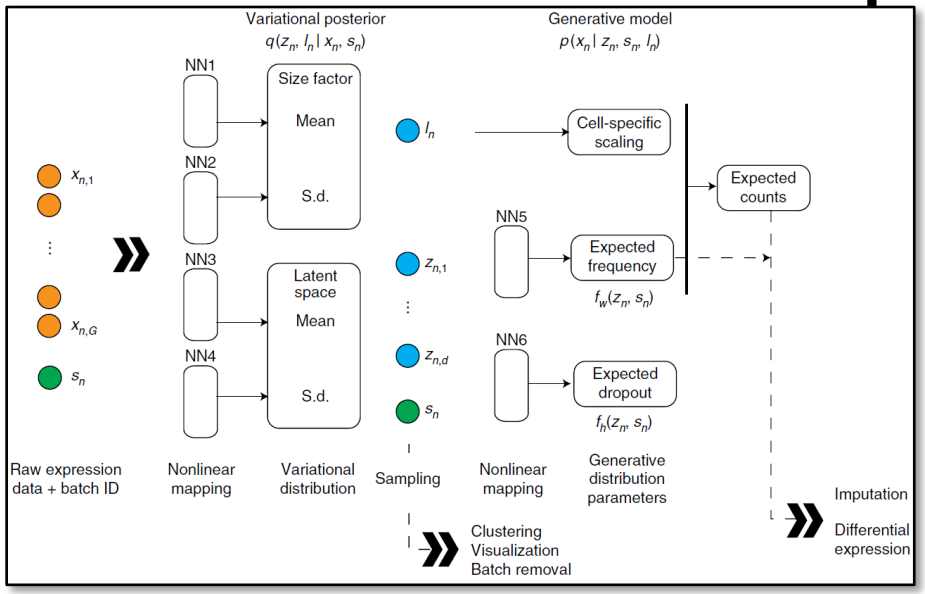
- λ - the regularization coefficient, to control the contribution of the regularization term in the loss or cost function.
- Size of the hidden layer or latent space dimensionality.
- Initial value of learning rate.
- Threshold value - We stop the gradient descent after the change in loss function value in consecutive iterations is less than the threshold value, implying convergence.

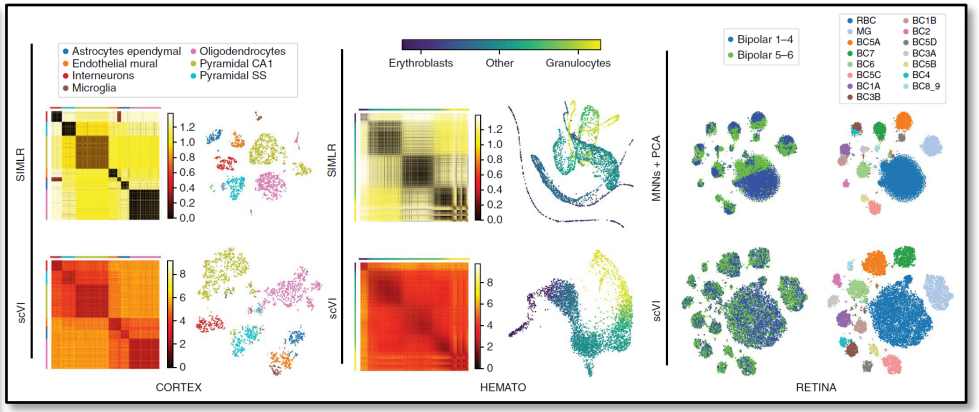
The best results were observed on the hyper-parameter choices shown in Table 2.

AutoImpute: Autoencoder based imputation of single-cell RNA-seq data

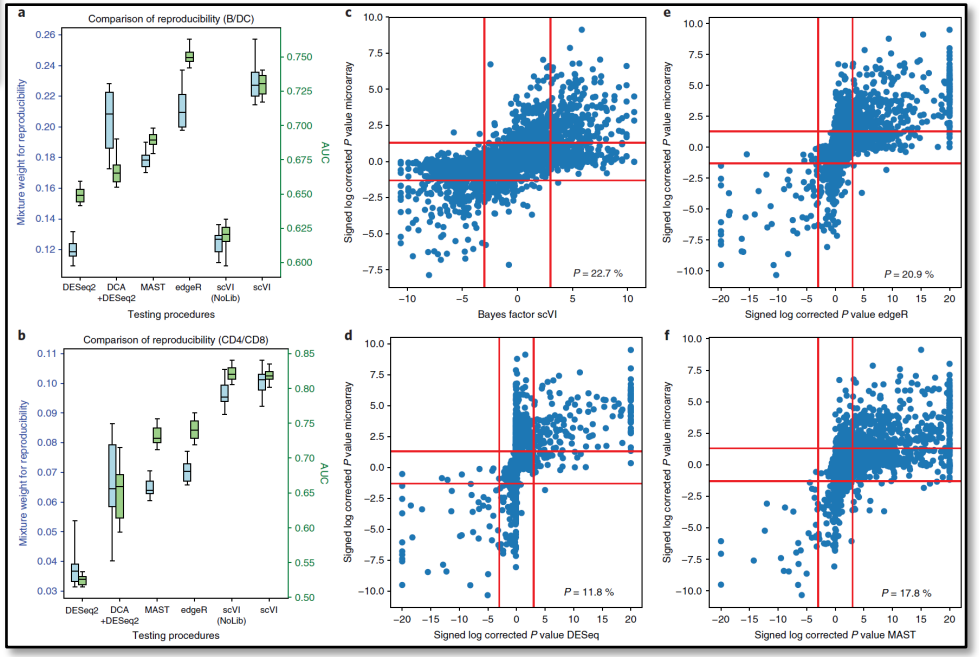
Divyanshu Talwar[1], Aanchal Mongia[1], Debarka Sengupta[1,3] & Angshul Majumdar[2]

scVI: Use NN to estimate params in variational inference





scVI retains biological signal in diverse datasets



scVI enables differential expression analysis

scVI: Learn non-linear embedding of cells for multiple analysis tasks
NN=Neural networks used to compute embedding and expr. distribution
f_w,f_h: functional representations NN5,6 to capture parameters of Gaussians

Modeled observed expression x_{ng} (gene g, cell n) as sample
Drawn from zero-inflated negative binomial (ZINB) distribution
Conditioned on the batch annotation s_n of each cell (if available)
And on two additional, unobserved random variables:

- ρ_g^n **nuisance variation**, 1-D Gaussian, model differences in capture efficiency & sequencing depth, cell-specific scaling factor
- z_n, **remaining variation**, 10-D Gaussian, model biological differences between cells.

Represent each cell as point in low-dimensional latent space (for visualization and clustering).
Neural network maps the latent variables to ZINB distribution parameters (Fig. 1a, neural networks 5 and 6).
This mapping goes through intermediate variables:

- batch-corrected, normalized estimate of the percentage of transcripts in each cell n that originate from each gene g

Use these estimates for differential expression analysis
Use scaled version (multiplying by estimated library size) for imputation.
Derived approximation for posterior distribution of latent variables q by training another neural network using variational inference and a scalable stochastic optimization procedure (NN1-NN4).

Non-Linear Dimensionality Reduction

Dimensionality reduction: Some Assumptions

- High-dimensional data often lies on or near a much lower dimensional, curved manifold.
- A good way to represent data points is by their low-dimensional coordinates.
- The low-dimensional representation of the data should capture information about high-dimensional pairwise distances.

The basic idea of non-parameteric dimensionality reduction

- Represent each data-point by a point in a lower dimensional space.
- Choose the low-dimensional points so that they optimally represent some property of the data-points (e.g. the pairwise distances).
  - Many different properties have been tried.
- Do not insist on learning a parametric "encoding" function that maps each individual data-point to its low-dimensional representative.
- Do not insist on learning a parametric "decoding" function that reconstructs a data-point from its low dimensional representative.

Two types of dimensionality reduction

- Global methods assume that all pairwise distances are of equal importance.
  - Choose the low-D pairwise distances to fit the high-D ones (using magnitude or rank order).
- Local methods assume that only the local distances are reliable in high-D.
  - Put more weight on modeling the local distances correctly.

Linear methods of reducing dimensionality

- PCA finds the directions that have the most variance.
  - By representing where each datapoint is along these axes, we minimize the squared reconstruction error.
  - Linear autoencoders are equivalent to PCA

- Multi-Dimensional Scaling arranges the low-dimensional points so as to minimize the discrepancy between the pairwise distances in the original space and the pairwise distances in the low-D space.

Metric Multi-Dimensional Scaling

- Find low dimensional representatives, y, for the high-dimensional data-points, x, that preserve pairwise distances as well as possible.
- An obvious approach is to start with random vectors for the y's and then perform steepest descent by following the gradient of the cost function.
- Since we are minimizing squared errors, maybe this has something to do with PCA?
  - If so, we don't need an iterative method to find the best embedding.

$$Cost = \sum_{i<j} (d_{ij} - \hat{d}_{ij})^2$$

$$d_{ij} = \| x_i - x_j \|^2$$

$$\hat{d}_{ij} = \| y_i - y_j \|^2$$

Converting metric MDS to PCA

- If the data-points all lie on a hyperplane, their pairwise distances are perfectly preserved by projecting the high-dimensional coordinates onto the hyperplane.
  - So in that particular case, PCA is the right solution.
- If we "double-center" the data, metric MDS is equivalent to PCA.
  - Double centering means making the mean value of every row and column be zero.
  - But double centering can introduce spurious structure.

Other non-linear methods of reducing dimensionality

- Non-linear autoencoders with extra layers are much more powerful than PCA but they can be slow to optimize and they get different, locally optimal solutions each time.

- Multi-Dimensional Scaling can be made non-linear by putting more importance on the small distances. A popular version is the Sammon mapping:

high-D distance

low-D distance

$$Cost = \sum_{i\,j} \left(\frac{\| \mathbf{x}_i - \mathbf{x}_j \| - \| \mathbf{y}_i - \mathbf{y}_j \|}{\| \mathbf{x}_i - \mathbf{x}_j \|} \right)^2$$

- Non-linear MDS is also slow to optimize and also gets stuck in different local optima each time.

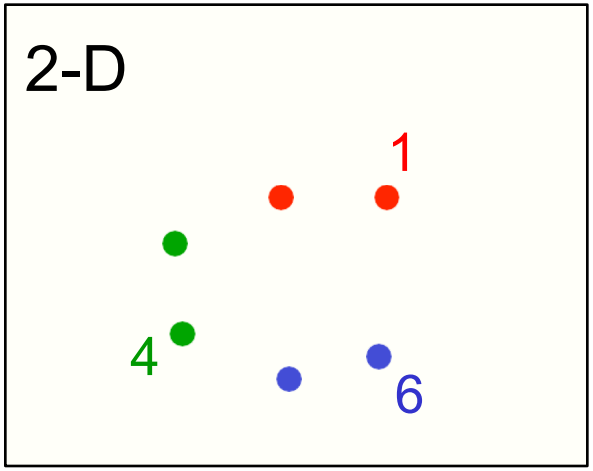
Problems with Sammon mapping

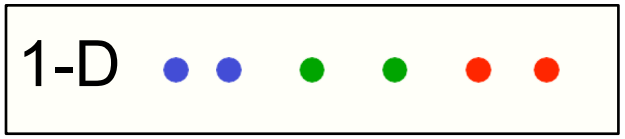
- It puts too much emphasis on getting very small distances exactly right.
- It produces embeddings that are circular with roughly uniform density of the map points.

IsoMap: Local MDS without local optima

- Instead of only modeling local distances, we can try to measure the distances along the manifold and then model these intrinsic distances.
  - The main problem is to find a robust way of measuring distances along the manifold.
  - If we can measure manifold distances, the global optimisation is easy: It's just global MDS (i.e. PCA)

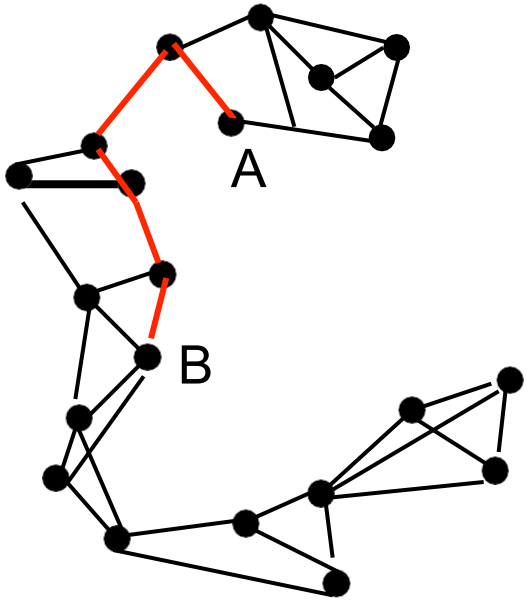
If we measure distances along the manifold, d(1,6) > d(1,4)



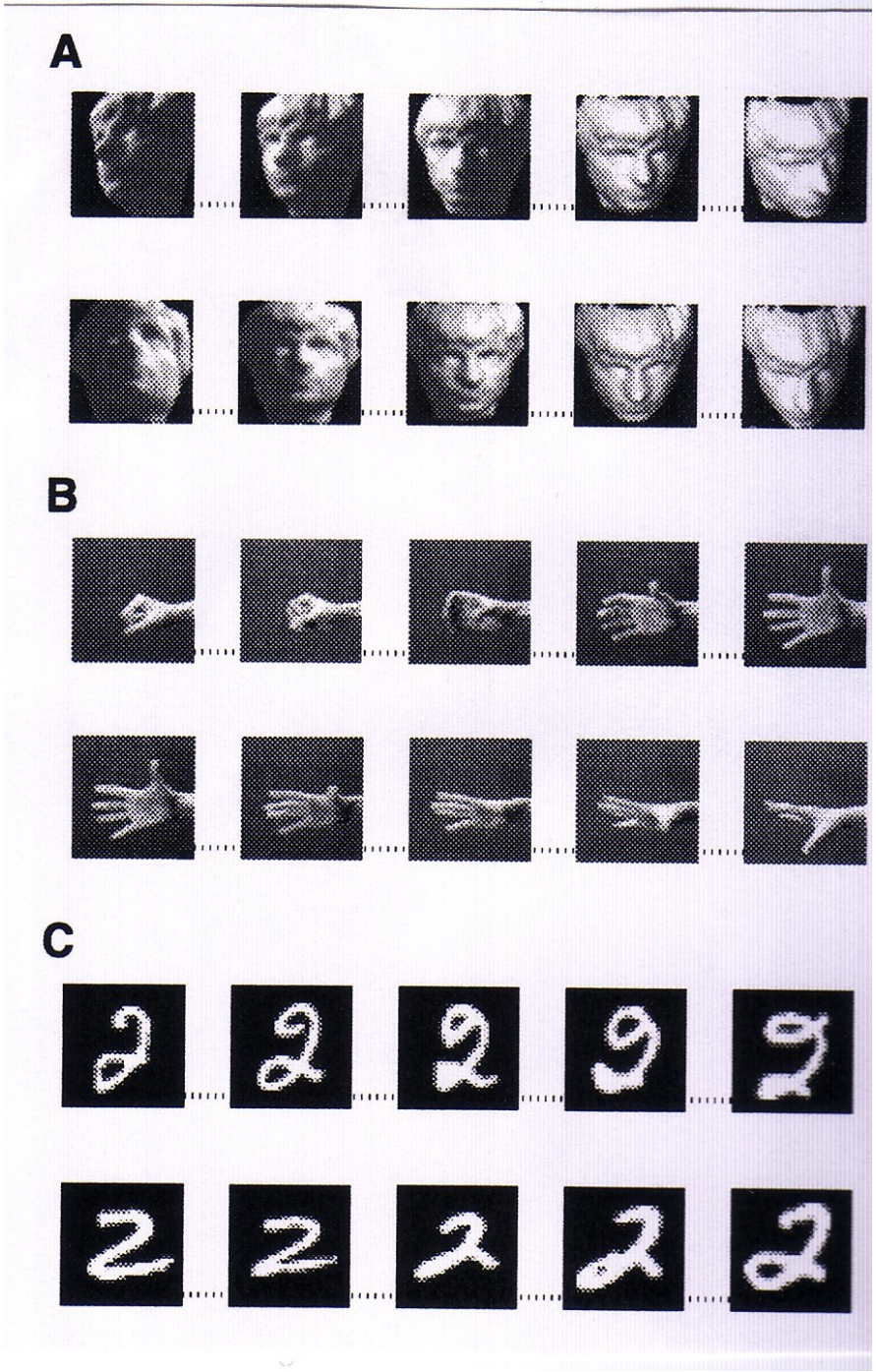


How Isomap measures intrinsic distances

- Connect each datapoint to its K nearest neighbors in the high-dimensional space.
- Put the true Euclidean distance on each of these links.
- Then approximate the manifold distance between any pair of points as the shortest path in this “neighborhood graph”.



Using Isomap to discover the intrinsic manifold in a set of face images



Linear methods cannot interpolate properly between the leftmost and rightmost images in each row.

This is because the interpolated images are NOT averages of the images at the two ends.

Isomap does not interpolate properly either because it can only use examples from the training set. It cannot create new images.

But it is better than linear methods.

Maps that preserve local geometry

- The idea is to make the local configurations of points in the low-dimensional space resemble the local configurations in the high-dimensional space.
- We need a coordinate-free way of representing a local configuration.
- If we represent a point as a weighted average of nearby points, the weights describe the local configuration.

$$\mathbf{x}_i \approx \sum_j w_{ij} \mathbf{x}_j$$

Finding the optimal weights

- This is easy.
- Minimize the squared "construction" errors subject to the sum of the weights being 1.

$$Cost = \sum_i \| \mathbf{x}_i - \sum_{j \varepsilon N(i)} w_{ij} \mathbf{x}_j \|^2, \qquad \sum_{j \varepsilon N(i)} w_{ij} = 1$$

- If the construction is done using less neighbors than the dimensionality of x, there will generally be some construction error
  - The error will be small if there are as many neighbors as the dimensionality of the underlying noisy manifold.

A sensible but inefficient way to use the local weights

- Assume a low-dimensional latent space.
  - Each datapoint $\mathbf{x}_i$ has latent coordinates $\mathbf{y}_i$.
- Find a set of latent points that minimize the construction errors produced by a two-stage process:
  - 1. First use the latent points to compute the local weights that construct $\mathbf{y}_i$ from its neighbors.
  - 2. Use those weights to construct the high-dimensional coordinates of a datapoint $\mathbf{x}_i$ from the high-dimensional coordinates of its neighbors.
- Unfortunately, this is a hard optimization problem.
  - Iterative solutions are expensive because they must repeatedly measure the construction error in the high-dimensional space.

Local Linear Embedding: A less sensible but more efficient way to use local weights

- Instead of using the the latent points plus the other datapoints to construct each held-out datapoint, do it the other way around.
- Use the datapoints to determine the local weights, then try to construct each latent point from its neighbors.
  - Now the construction error is in the low-dimensional latent space.
- We only use the high-dimensional space once to get the local weights.
  - The local weights stay fixed during the optimization of the latent coordinates.
  - This is a much easier search.

The convex optimization

fixed weights

$$Cost = \sum_i \| \mathbf{y}_i - \sum_{j \varepsilon N(i)} w_{ij} \mathbf{y}_j \|^2$$

- Find the y's that minimize the cost subject to the constraint that the y's have unit variance on each dimension.
  - Why do we need to impose a constraint on the variance?

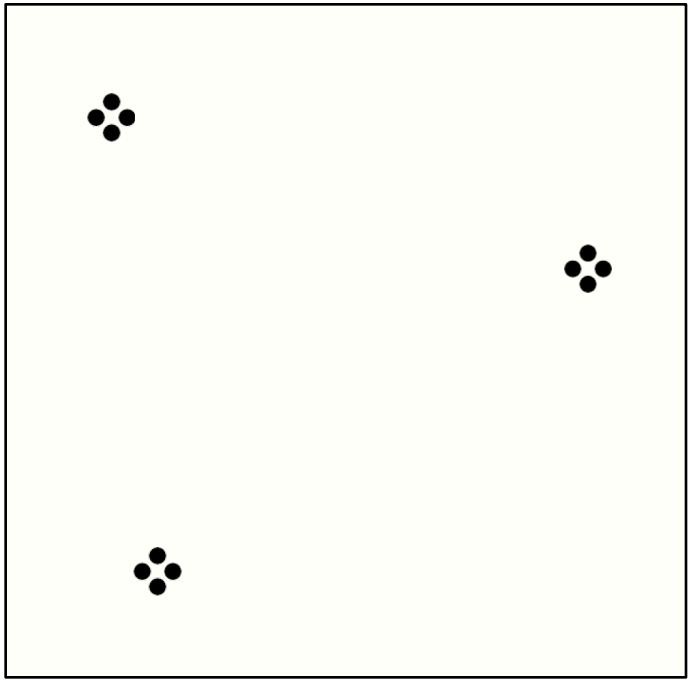
The collapse problem

- If all of the latent points are identical, we can construct each of them perfectly as a weighted average of its neighbors.
  - The root cause of this problem is that we are optimizing the wrong thing.
  - But maybe we can fix things up by adding a constraint that prevents collapse.
- Insist that the latent points have unit variance on each latent dimension.
  - This helps a lot, but sometimes LLE can satisfy this constraint without doing what we really intend.

Failure modes of LLE

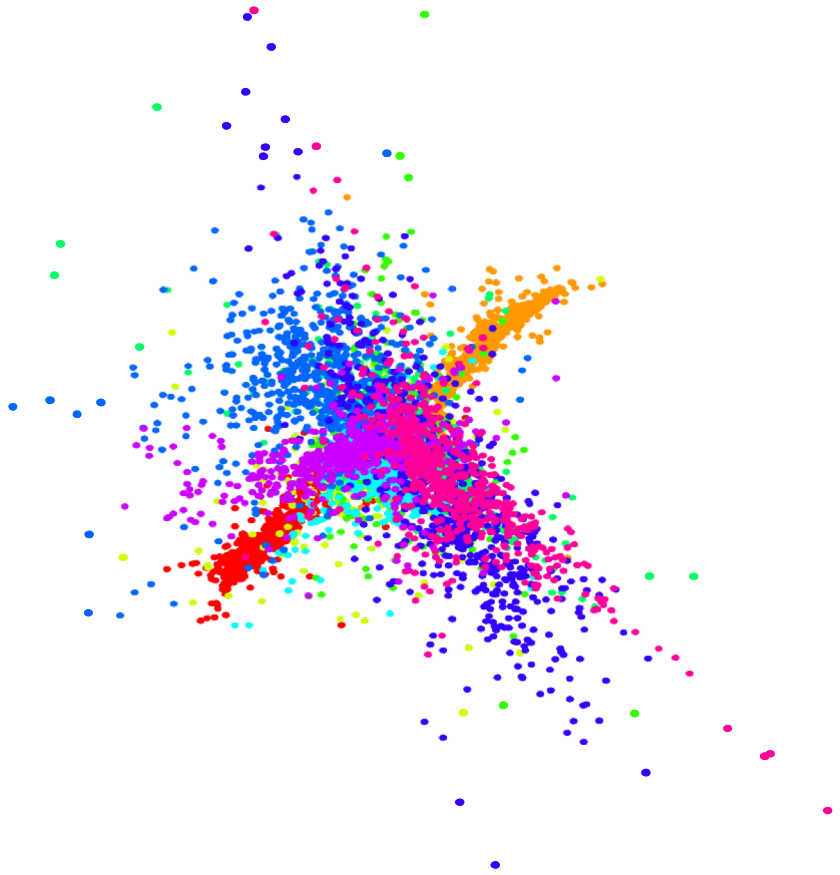
- If the neighborhood graph has several disconnected pieces, we can satisfy the unit variance constraint and still have collapses.

- Even if the graph is fully connected, it may be possible to collapse all the densely connected regions and satisfy the variance constraint by paying a high cost for a few outliers.



A typical embedding found by LLE

- LLE embeddings often look like this.
- Most of the data is close to the center of the space.
- A few points are far from the center to satisfy the unit variance constraint.



A comment on LLE

- It has two very attractive features
  - 1. The only free parameters are the dimensionality of the latent space and the number of neighbors that are used to determine the local weights.
  - 2. The optimization is convex so we don't need multiple tries and we don't need to fiddle with optimization parameters.
- It has one bad feature:
  - It is not optimizing the right thing!
  - One consequence is that it does not have any incentive to keep widely separated datapoints far apart in the low-dimensional map.

Maximum Variance Unfolding

- This fixes one of the problems of LLE and still manages to be a convex optimization problem.
- Use a few neighbors for each datapoint and insist that the high-dimensional distances between neighbors are exactly preserved in the low-dimensional space.
  - This is like connecting the points with rods of fixed lengths.
- Subject to the rigid rods connecting the low-dimensional points, maximize their squared separations.
  - This encourages widely separated datapoints to remain separated in the low-dimensional space.

How to solve many problems in AI

- 1. Map from the data domain to a domain of feature vectors in which the important relationships can be modeled by linear operations.
- 2. Do some linear operations.
- 3. Map the answer back to the data domain.

Modeling relational data

- Suppose we have a set of facts of the form ARB
  - i.e. The relation R maps A to B as in Allan has-mother Beatrice
- We could model the facts using matrix algebra.
  - Learn a vector for each object
  - Learn a matrix for each relation
  - The aim is to make A*R=B
- This doesnt work because all the vectors learn to be zero.
  - We need A*R to be closer to B than to C.

A discriminative cost function

$$p_{B|AR} = \frac{e^{\|AR-B\|^2}}{\sum_C e^{\|AR-C\|^2}}$$

Applying the idea to dimensionality reduction

- 0. Compute a big probability table that contains the probability that each high-dimensional data-point , i, would pick another data-point , j, as its “neighbor”.
- 1. Use a learned look-up table to convert each high-dimensional data-point to a 2-D feature vector.
- 2. Multiply by the identity matrix.
- 3. Compare the resulting 2-D feature vector with all the other 2-D feature vectors to get a predicted distribution over data-points in the original data-space.
- Learn the look-up tables so that the probabilities computed in the 2-D space match the probabilities computed in the original space.

A probabilistic version of local MDS

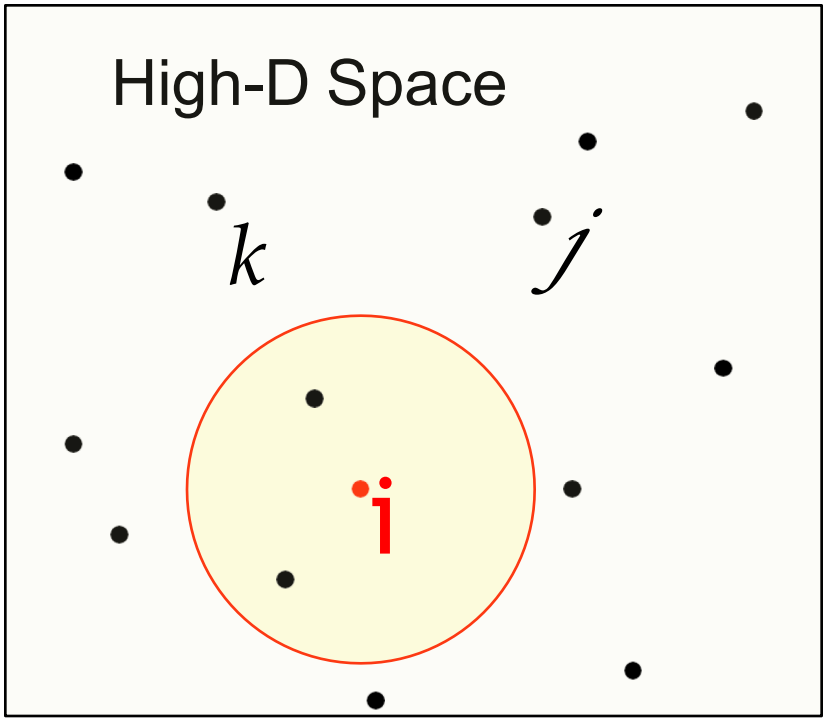
- It is more important to get local distances right than non-local ones, but getting infinitessimal distances right is not infinitely important.
  - All the small distances are about equally important to model correctly.
  - Stochastic neighbor embedding has a probabilistic way of deciding if a pairwise distance is “local”.

Stochastic Neighbor Embedding

- First convert each high-dimensional similarity into the probability that one data point will pick the other data point as its neighbor.
- To evaluate a map:
  - Use the pairwise distances in the low-dimensional map to define the probability that a map point will pick another map point as its neighbor.
  - Compute the Kullback-Leibler divergence between the probabilities in the high-dimensional and low-dimensional spaces.

A probabilistic local method

- Each point in high-D has a conditional probability of picking each other point as its neighbor.
- The distribution over neighbors is based on the high-D pairwise distances.
  - If we do not have coordinates for the datapoints we can use a matrix of dissimilarities instead of pairwise distances.



$$p_{j|i} = \frac{e^{-d_{ij}^2 / 2\sigma_i^2}}{\sum_k e^{-d_{ik}^2 / 2\sigma_i^2}}$$

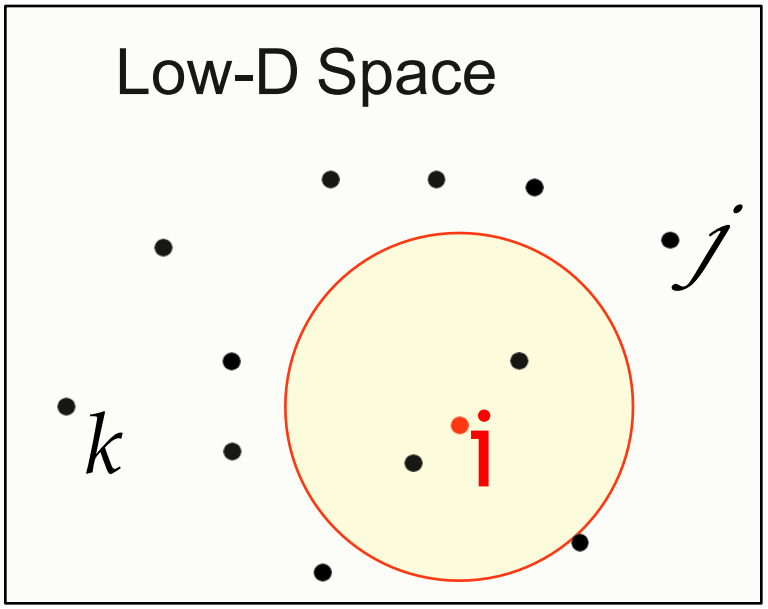
probability of picking j given that you start at i

Throwing away the raw data

- The probabilities that each points picks other points as its neighbor contains all of the information we are going to use for finding the manifold.
  - Once we have the probabilities $p_{j|i}$ we do not need to do any more computations in the high-dimensional space.
  - The input could be "dissimilarities" between pairs of datapoints instead of the locations of individual datapoints in a high-dimensional space.

Evaluating an arrangement of the data in a low-dimensional space

- Give each datapoint a location in the low-dimensional space.
  - Evaluate this representation by seeing how well the low-D probabilities model the high-D ones.



$$q_{j|i} = \frac{e^{-d_{ij}^2}}{\sum_k e^{-d_{ik}^2}}$$

probability of picking j given that you start at i

The cost function for a low-dimensional representation

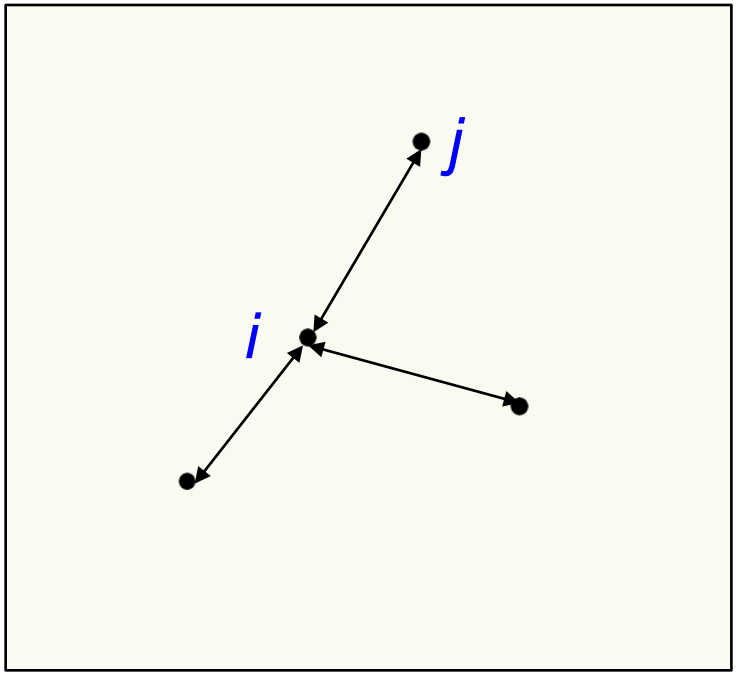
$$Cost = \sum_i KL(P_i \,||\, Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

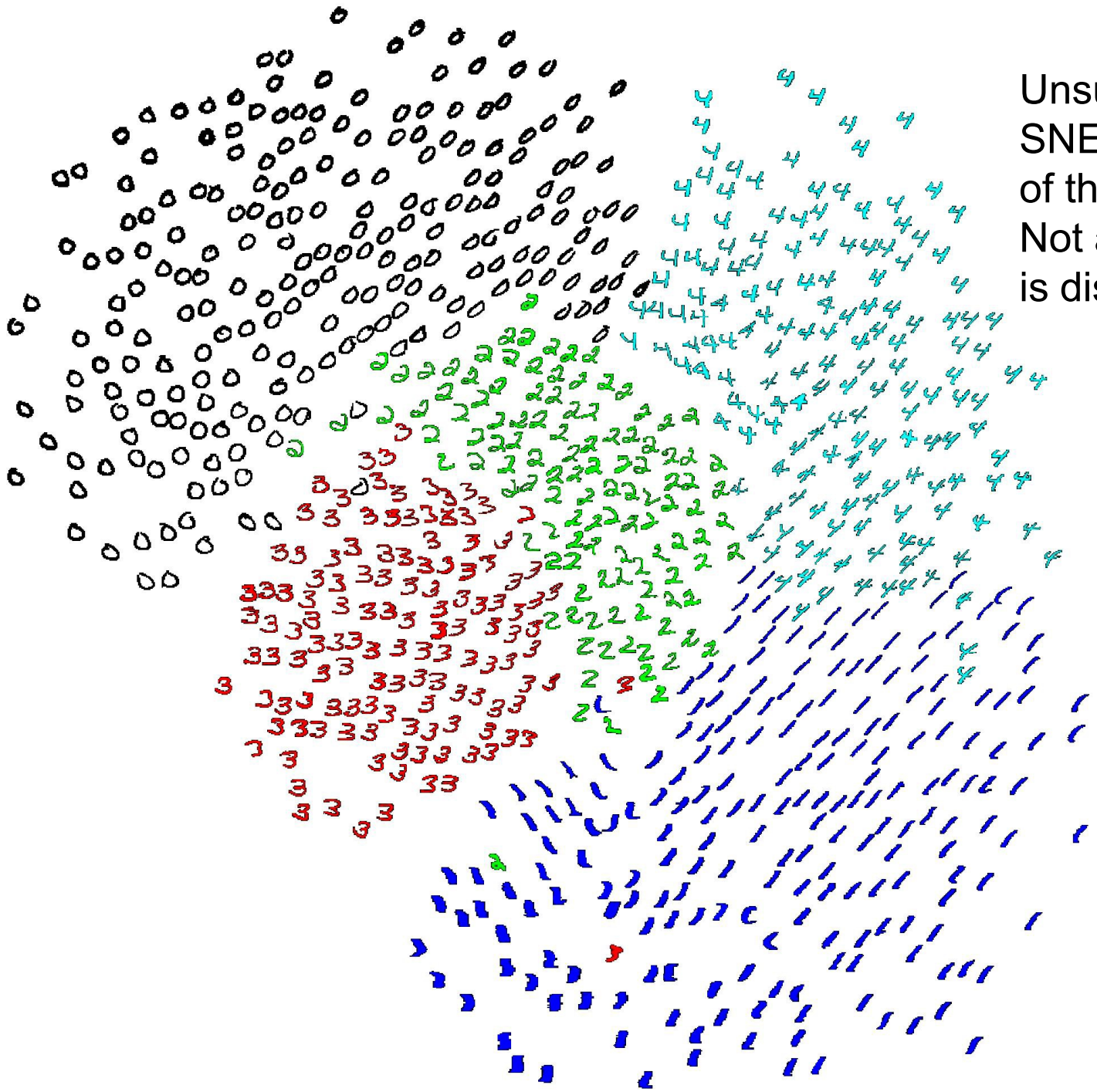
- For points where p_{ij} is large and q_{ij} is small we lose a lot.
  - Nearby points in high-D really want to be nearby in low-D
- For points where q_{ij} is large and p_{ij} is small we lose a little because we waste some of the probability mass in the Q_i distribution.
  - Widely separated points in high-D have a mild preference for being widely separated in low-D.

The forces acting on the low-dimensional points

$$\frac{\partial Cost}{\partial \mathbf{y}_i} = 2\sum_j (\mathbf{y}_j - \mathbf{y}_i)\left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j}\right)$$

- Points are pulled towards each other if the p's are bigger than the q's and repelled if the q's are bigger than the p's





Unsupervised SNE embedding of the digits 0-4. Not all the data is displayed

Picking the radius of the gaussian that is used to compute the p's

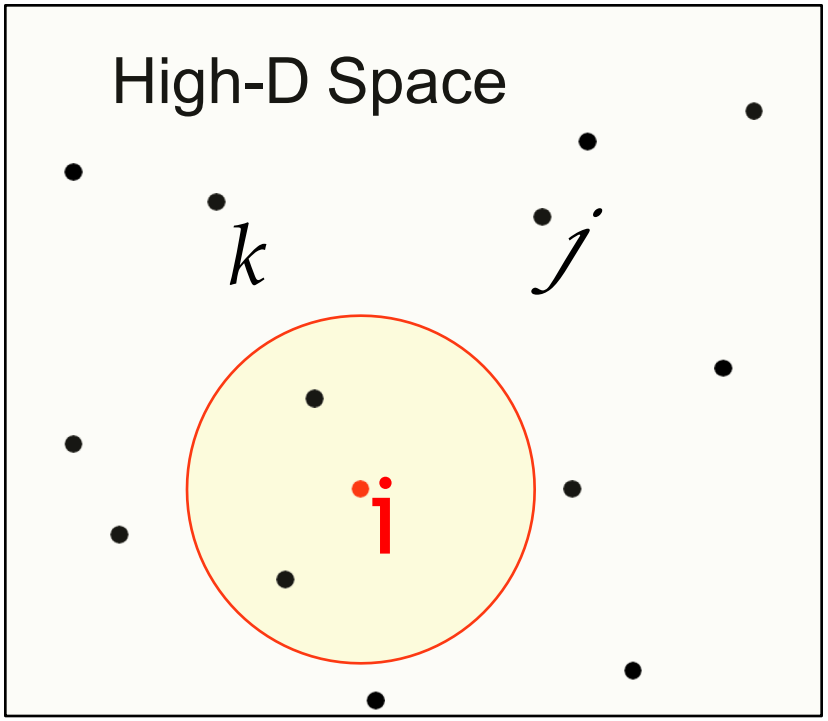
- We need to use different radii in different parts of the space so that we keep the effective number of neighbors about constant.
- A big radius leads to a high entropy for the distribution over neighbors of i.
- A small radius leads to a low entropy.
- So decide what entropy you want and then find the radius that produces that entropy.
- Its easier to specify 2^entropy
  - This is called the perplexity
  - It is the effective number of neighbors.

Symmetric SNE

- There is a simpler version of SNE which seems to work about equally well.
- Symmetric SNE works best if we use different procedures for computing the p's and the q's
  - This destroys the nice property that if we embed in a space of the same dimension as the data, the data itself is the optimal solution.

Computing the p's for symmetric SNE

- Each high dimensional point, i, has a conditional probability of picking each other point, j, as its neighbor.
- The conditional distribution over neighbors is based on the high-dimensional pairwise distances.



$$p_{j|i} = \frac{e^{-d_{ij}^2 / 2\sigma_i^2}}{\sum_k e^{-d_{ik}^2 / 2\sigma_i^2}}$$

probability of picking j
given that you start at i

Turning conditional probabilities into pairwise probabilities

To get a symmetric probability between i and j we sum the two conditional probabilities and divide by the number of points (points are not allowed to choose themselves).

joint probability of picking the pair i,j

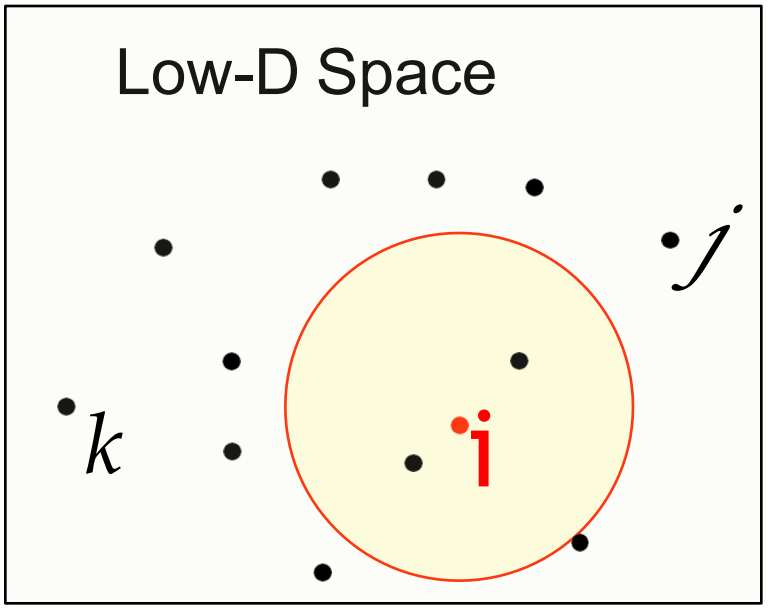
$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

This ensures that all the pairwise probabilities sum to 1 so they can be treated as probabilities.

$$\sum_{i,j} p_{ij} = 1$$

Evaluating an arrangement of the points in the low-dimensional space

- Give each data-point a location in the low-dimensional space.
  - Define low-dimensional probabilities symmetrically.
  - Evaluate the representation by seeing how well the low-D probabilities model the high-D affinities.



$$q_{ij} = \frac{e^{-d_{ij}^2}}{\sum_{k<l} e^{-d_{kl}^2}}$$

The cost function for a low-dimensional representation

$$Cost = KL(P \| Q) = \sum_{i<j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- It's a single KL instead of the sum of one KL for each datapoint.

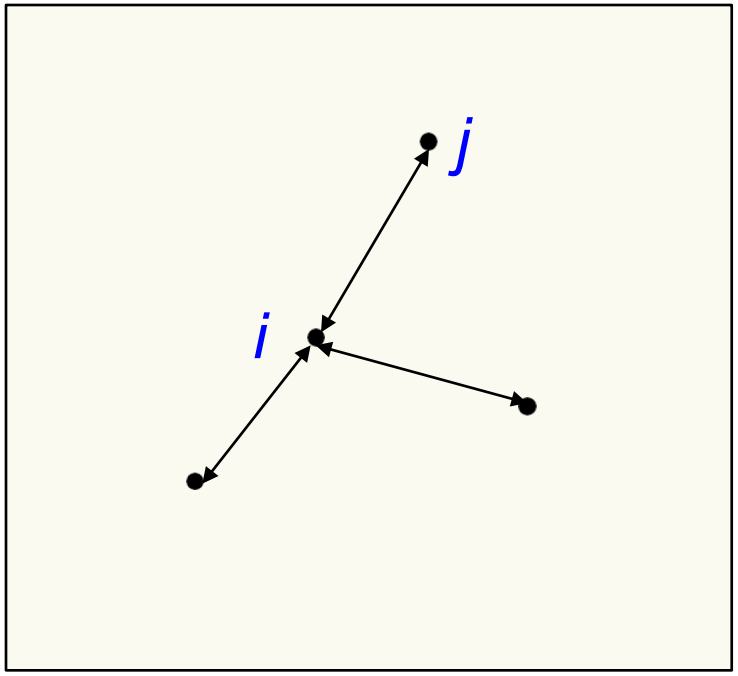
The forces acting on the low-dimensional points

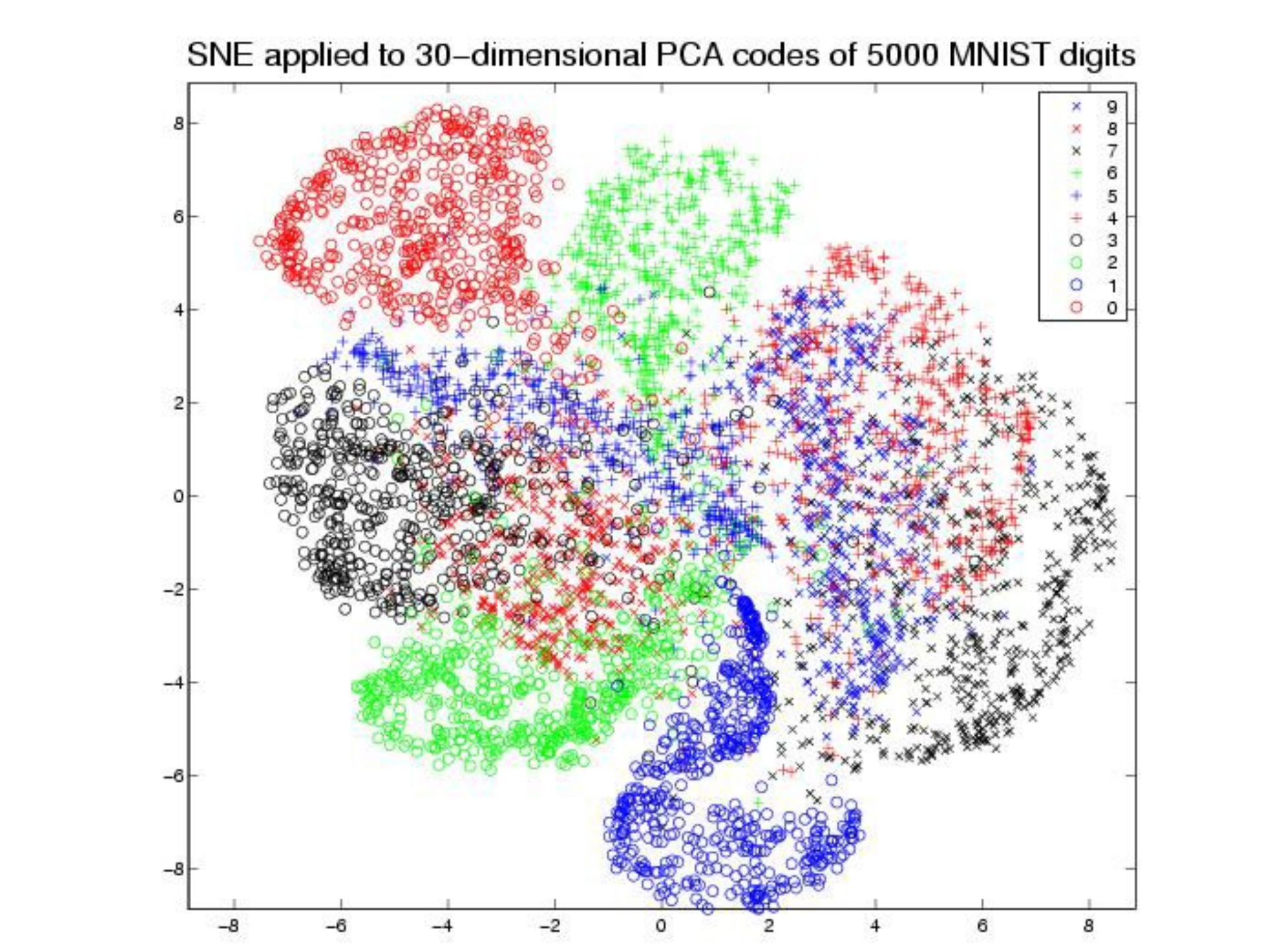
$$\frac{\partial KL(P \| Q)}{\partial \mathbf{y}_i} = 2\sum_j (\mathbf{y}_i - \mathbf{y}_j)(p_{ij} - q_{ij})$$

extension

stiffness

- Points are pulled towards each other if the p's are bigger than the q's and repelled if the q's are bigger than the p's
  - Its equivalent to having springs whose stiffnesses are set dynamically.




SNE applied to 30−dimensional PCA codes of 5000 MNIST digits
9
8
7
6
5
4
3
2
1
0
8
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6
8

Optimization methods for SNE

- We get much better global organization if we use annealing.
  - Add Gaussian noise to the y locations after each update.
  - Reduce the amount of noise on each iteration.
  - Spend a long time at the noise level at which the global structure starts to form from the hot plasma of map points.
- It also helps to use momentum (especially at the end).
- It helps to use an adaptive global step-size.

More optimization tricks for SNE

- Anneal the perplexity.
  - This is expensive because it involves computing distances in the high-dimensional data-space.
- Dimension decay
  - Use additional dimensions to avoid local optima, then penalize the squared magnitudes of the map points on the extra dimensions.
    - Turn up the penalty coefficient until all of the map points have very small values on those extra dimensions.
- Neither of these tricks is a big win in general.

A more interesting variation that uses the probabilistic foundation of SNE

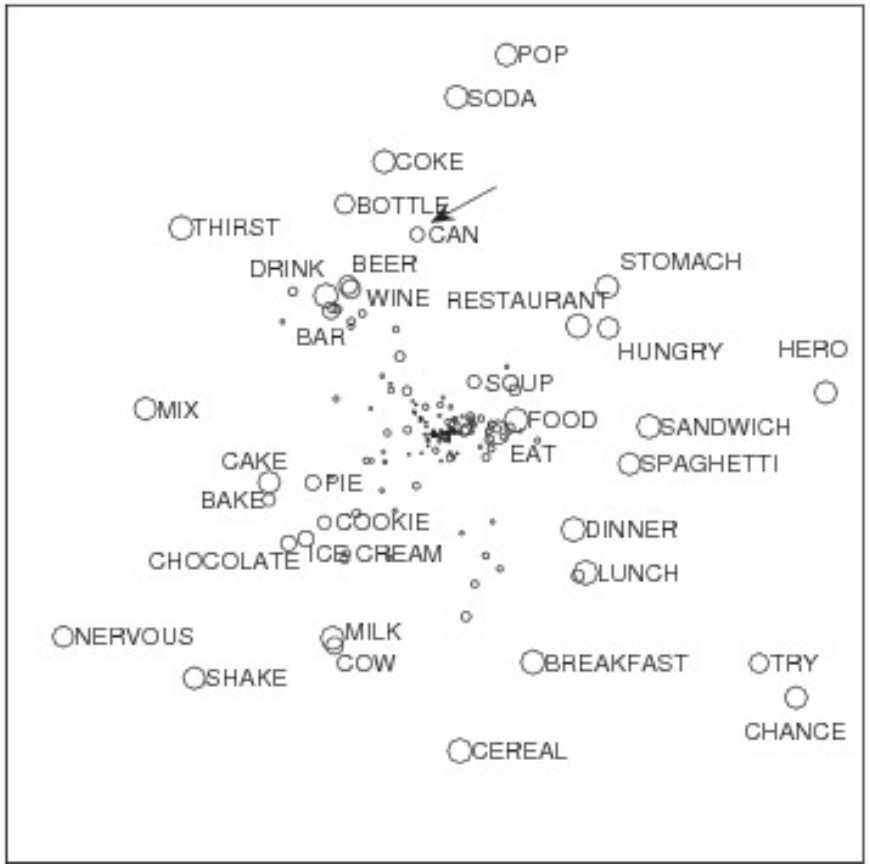
- All other dimensionality reduction methods assume that each data point is represented by ONE point in the map.
- But suppose we had several different maps.
  - Each map has a representative of each datapoint and the representative has a mixing proportion.
  - The overall qij is a sum over all maps

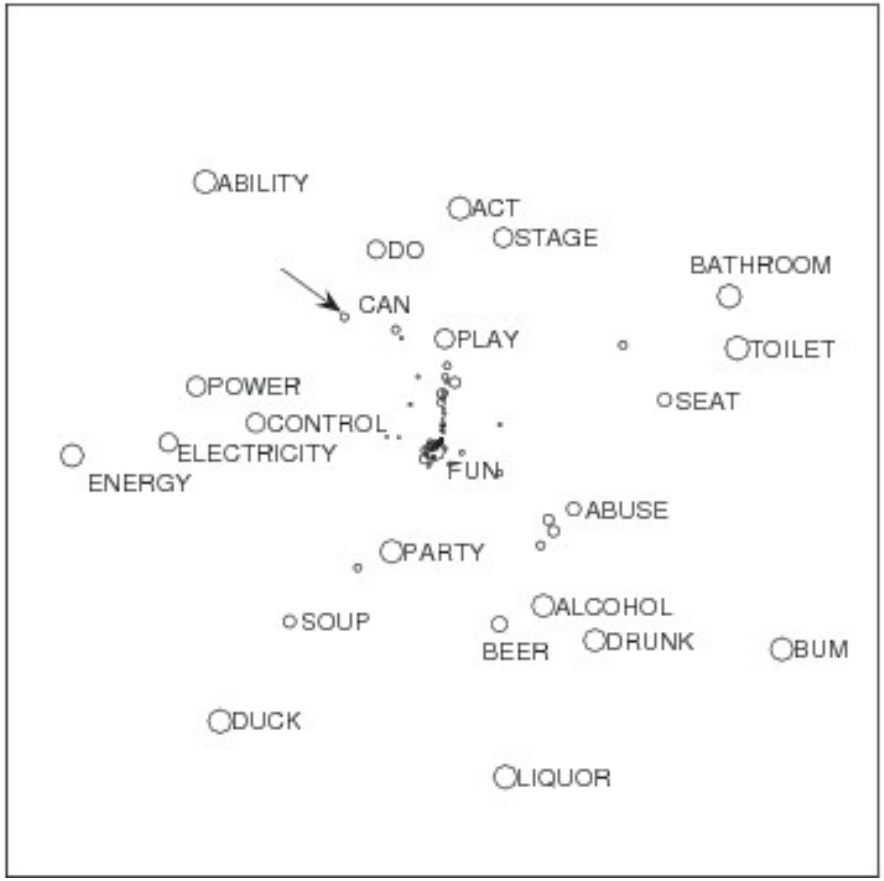
$$q_{ij} = \sum_m q_{ij}^m \qquad q_{ij}^m = \frac{\pi_i^m \pi_j^m \exp(-d_{ij}^m)}{Z}$$

A nice dataset for testing “Aspect maps”

- Give someone a word and ask them to say the first other word they associate with it.
  - Different senses of a word will have different associations and so they should show up in different aspect maps.

Two of the 50 aspect maps for the Florida word association data





The relationship between aspect maps and clustering

- If we force all of the locations in each map to be the same, it's a form of spectral clustering!
- Putting a point into a map is not just based on the affinities it has to the other points in the map.
  - It depends on whether it can find a location in the map that allows it to mathc the pattern of affinities.
  - It has a very abstract resemblance to mixtures of experts vs ordinary mixtures.

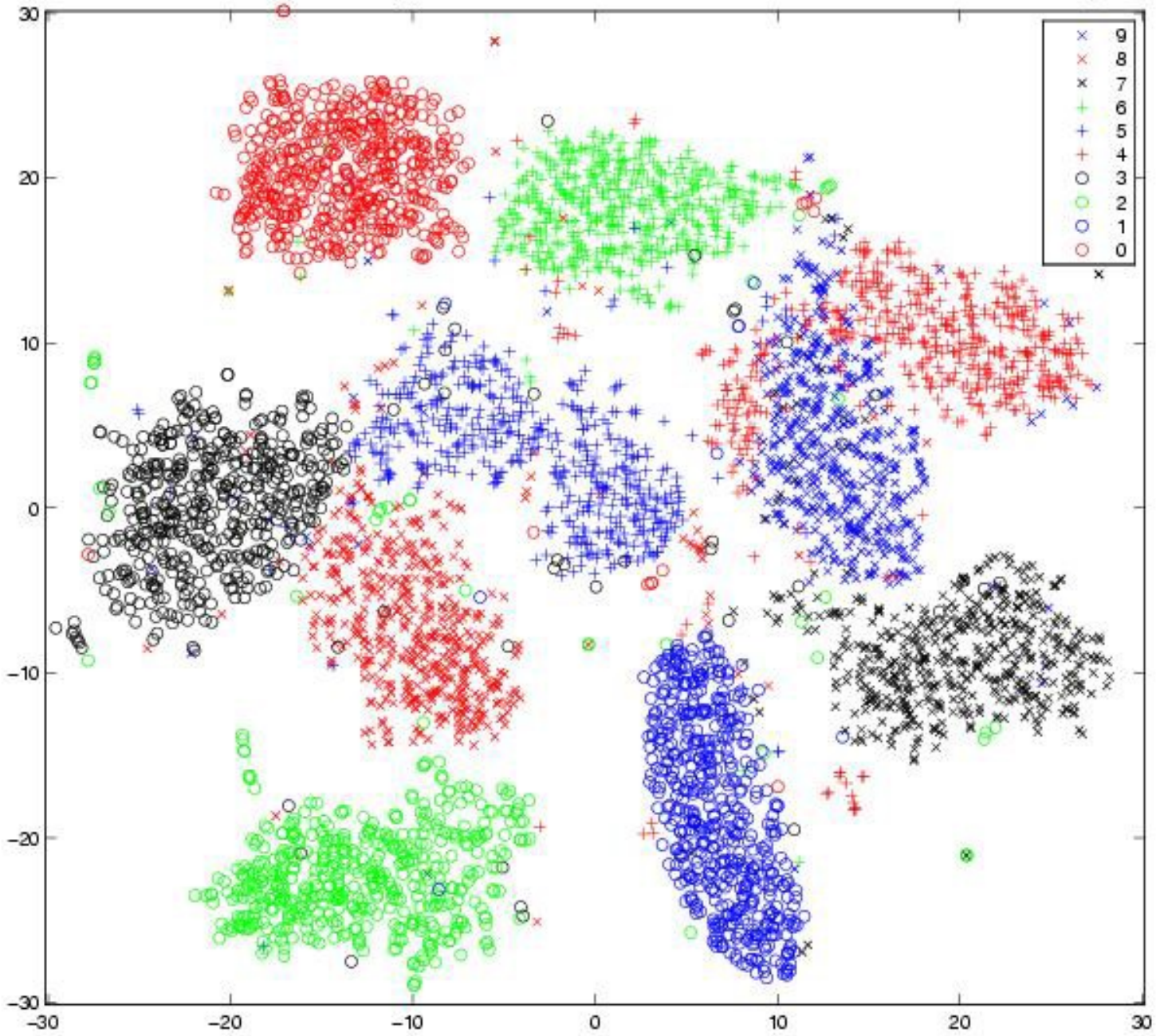
A weird behaviour of aspect maps

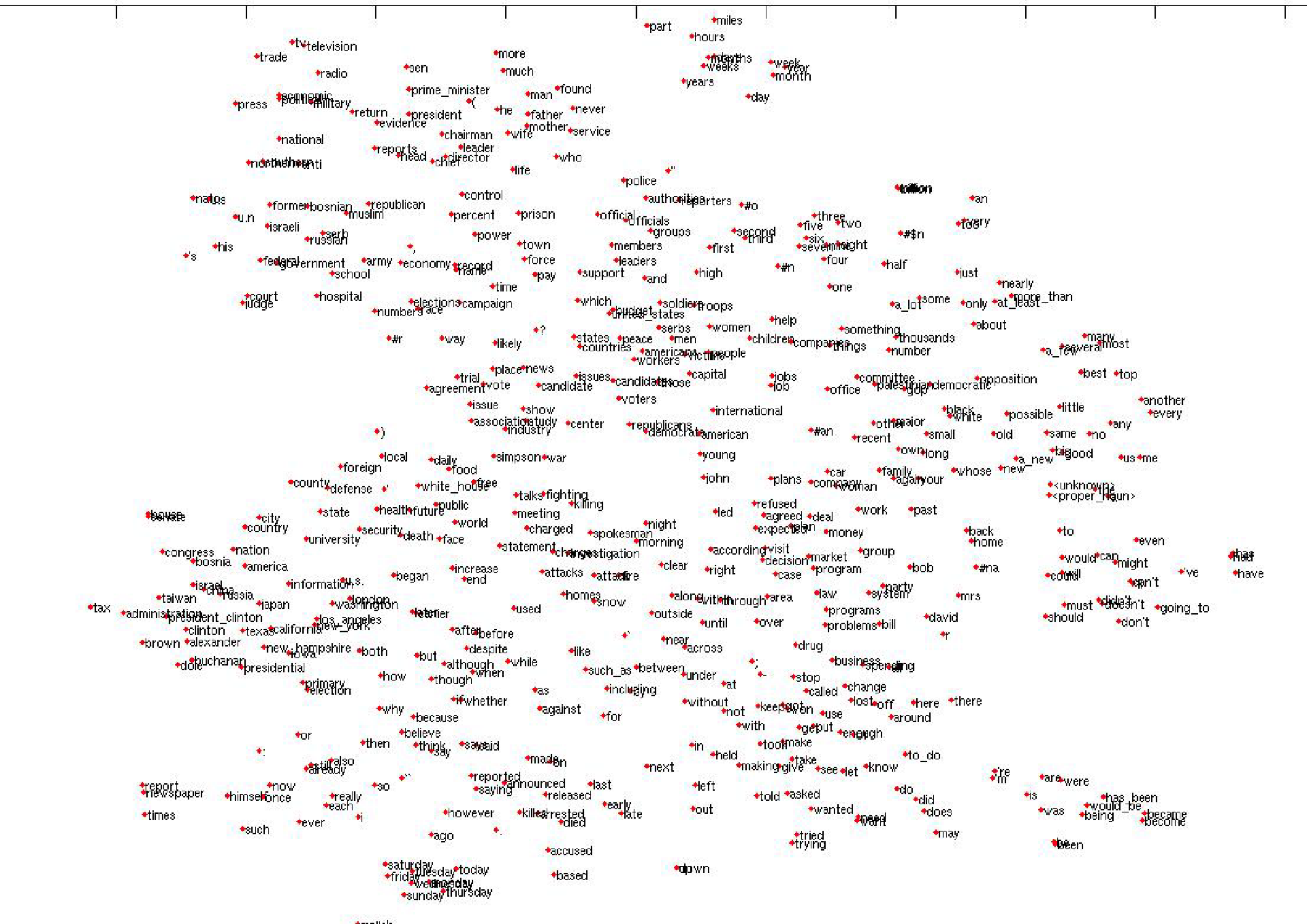
- If we use just 2 aspect maps, one of them collapses all of the map points to the same location.
  - Its trying to tell us something!
- It wants to use a uniform background probability for all pairs

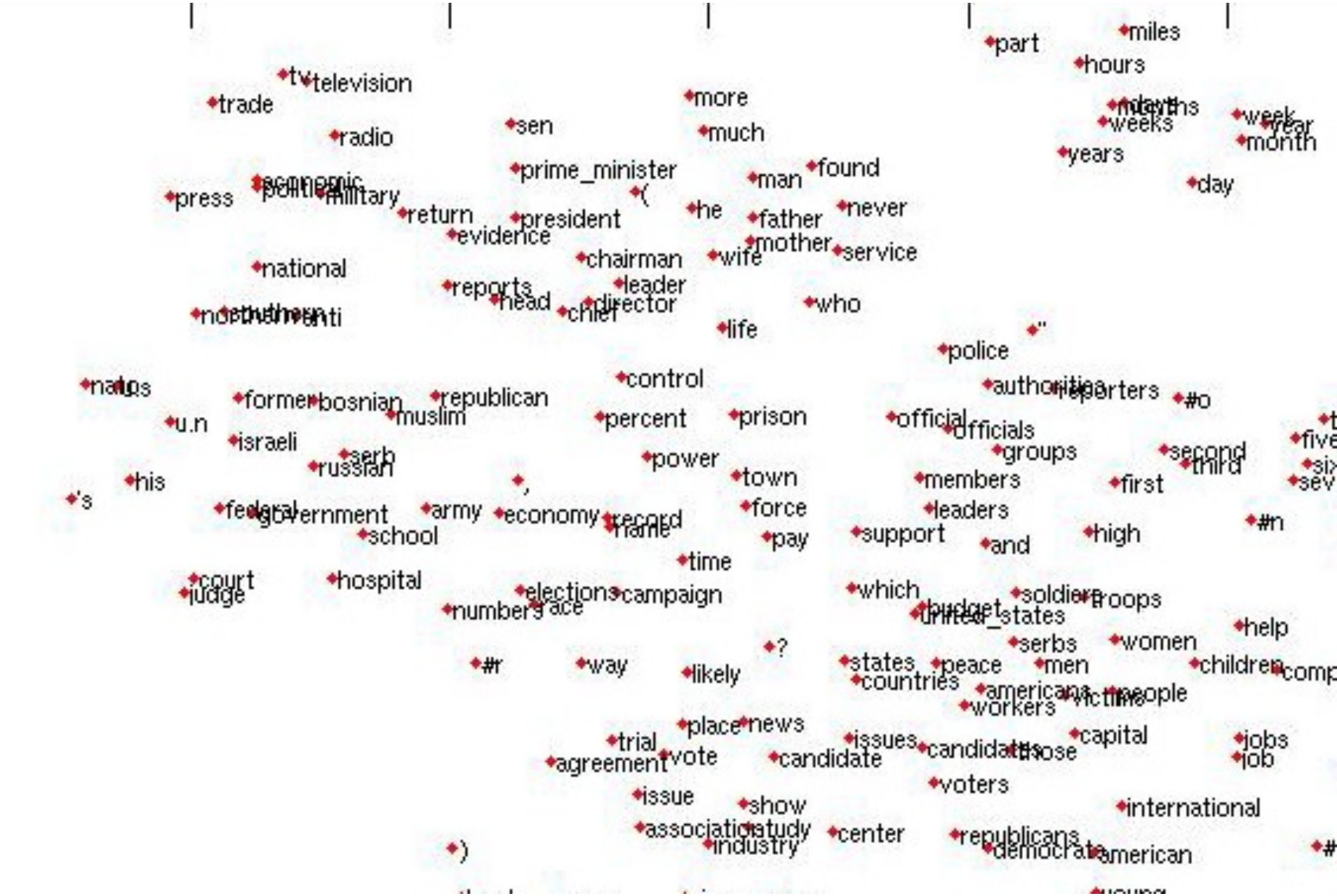
$$q_{ij} = \pi_1 \, q_{ij}^1 + \frac{1-\pi_1}{N^2}$$

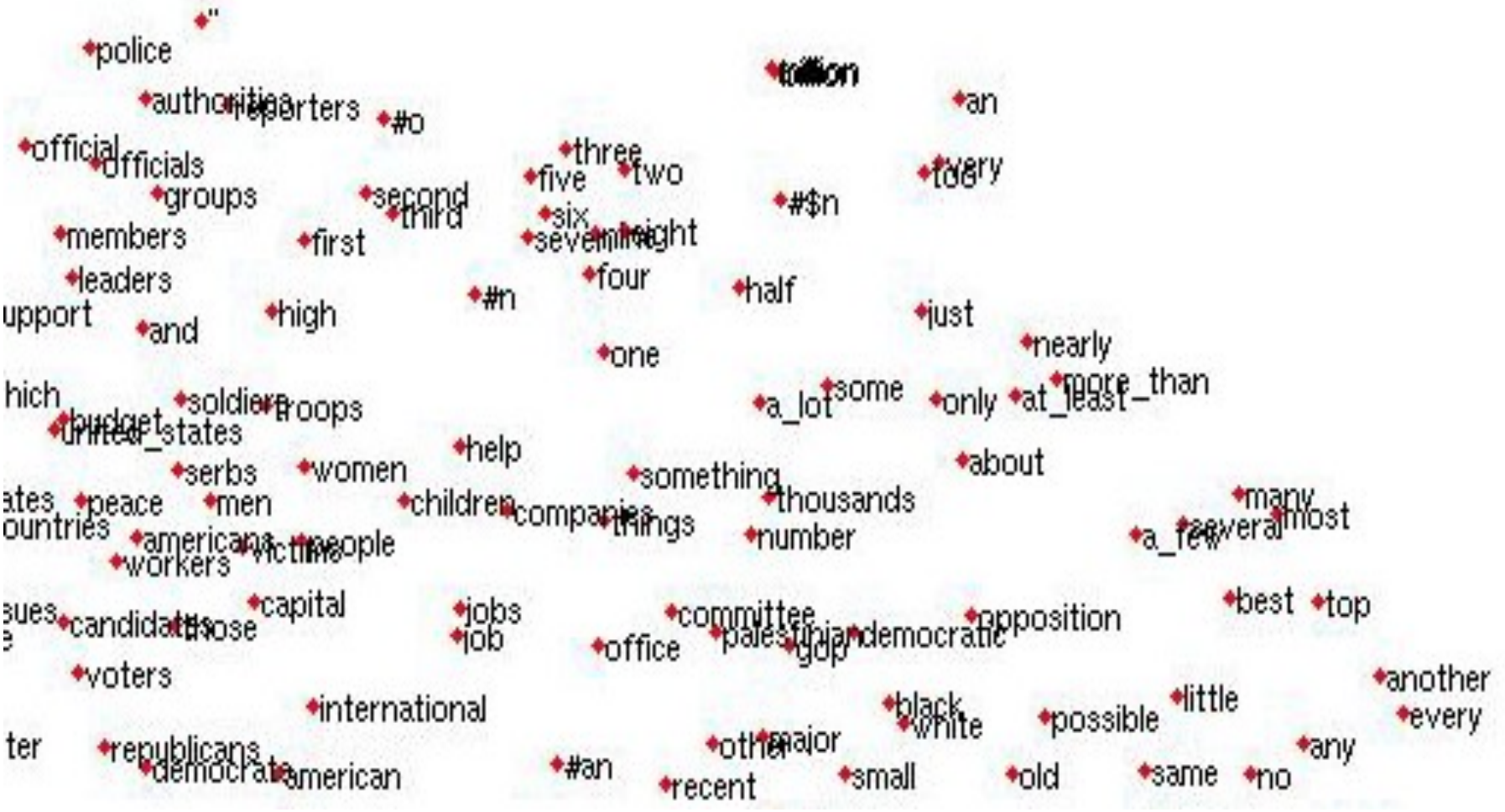
Why SNE does not have gaps between classes

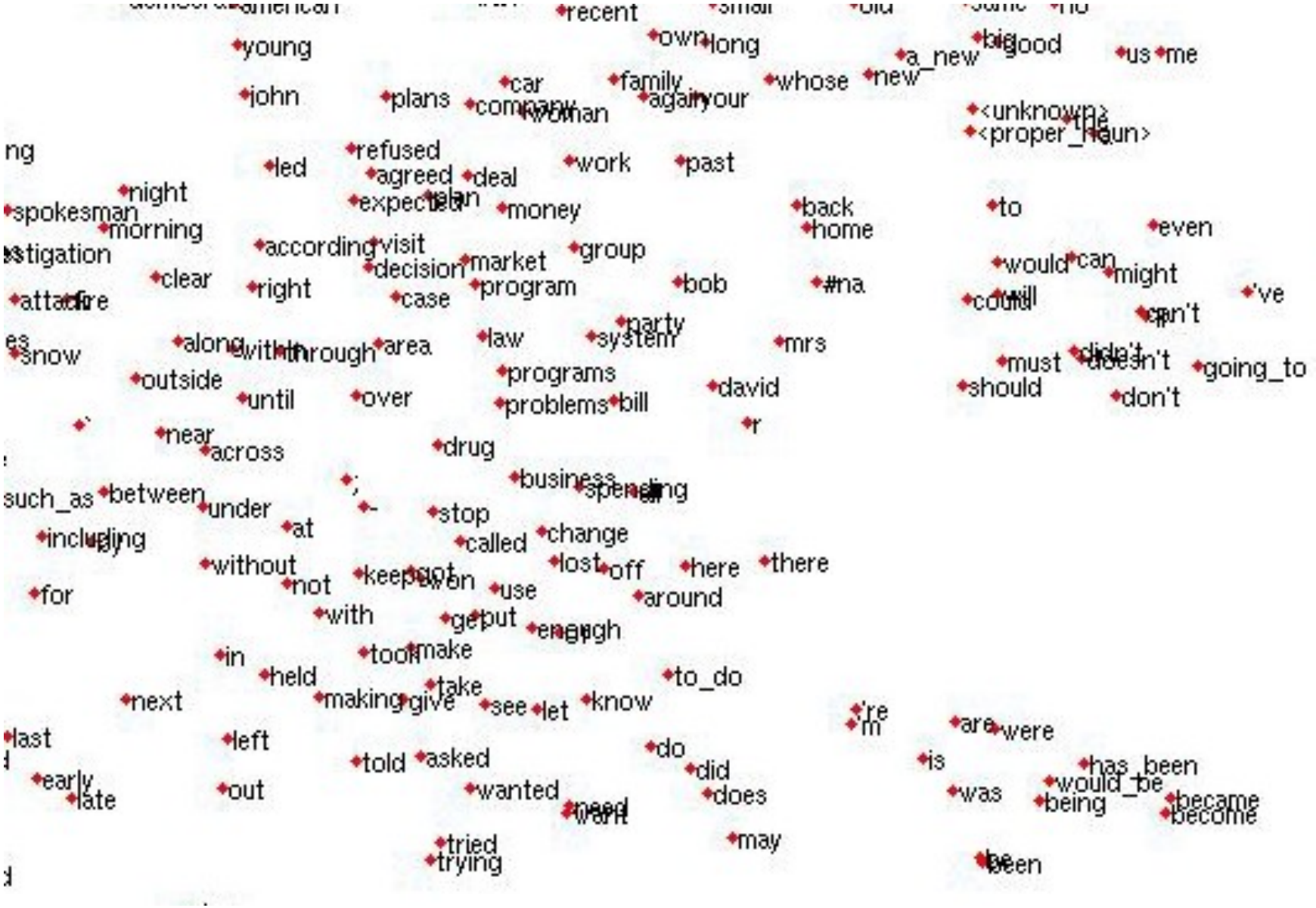
- In the high-dimensional space there are many pairs of points that are moderately close to each other.
  - The low-D space cannot model this. It doesn't have enough room around the edges.
- So there are many pij's that are modeled by smaller qij's.
  - This has the effect of lots of weak springs pulling everything together and crushing different classes together in the middle of the space.
- A uniform background model eliminates this effect and allows gaps between classes to appear.
  - It is quite like Maximum Variance Unfolding

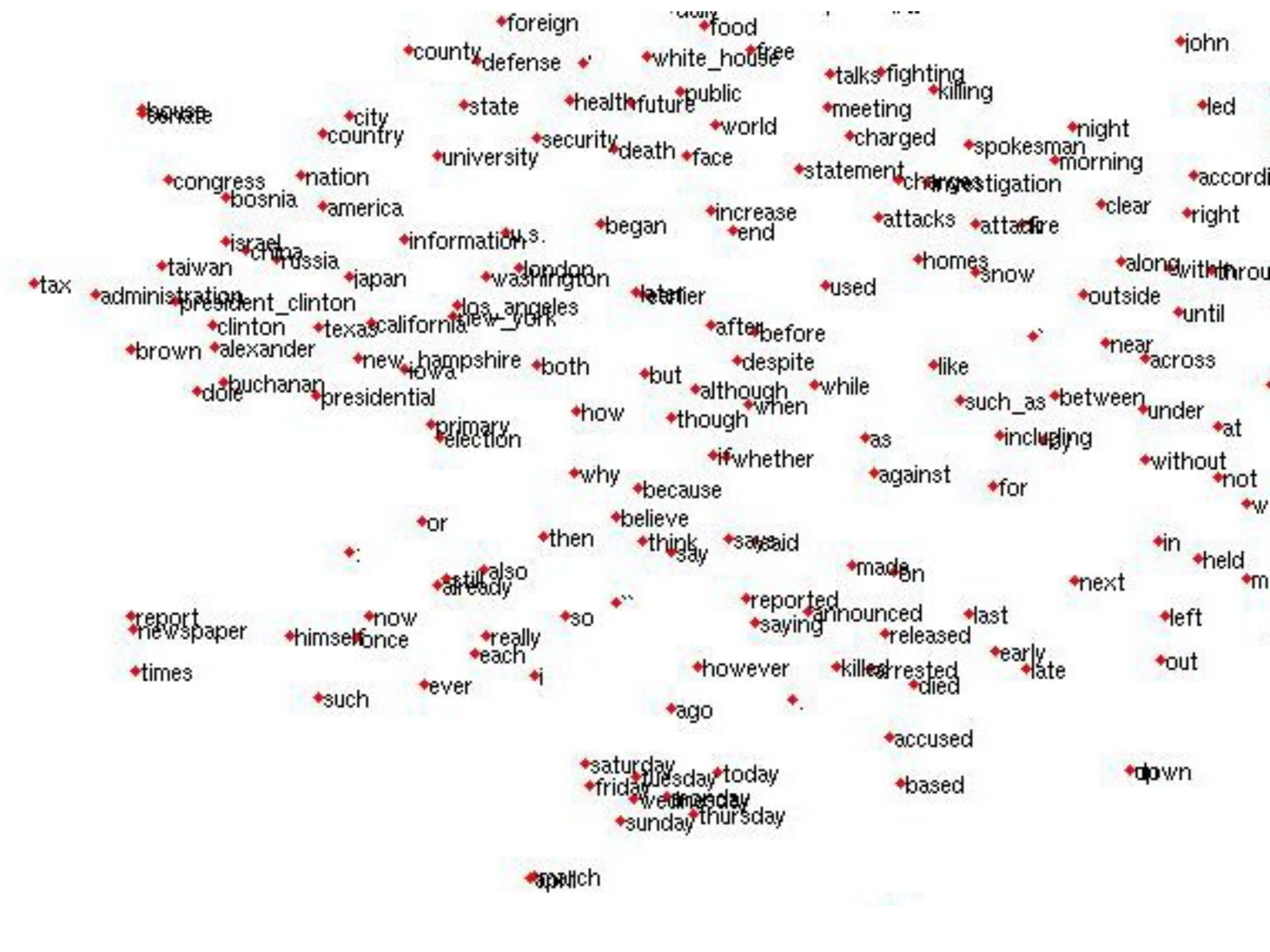

SNE then UNI−SNE applied to 30−D PCA codes of 5000 MNIST digits
9
8
7
6
5
4
3
2
1
0
30
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30


part
miles
hours
weeks
week
month
years
day
tv
television
trade
radio
press
military
return
evidence
sen
prime_minister
president
chairman
reports
head
chief
leader
director
more
much
man
found
he
father
never
wife
mother
service
who
life
national
police
authorities
reporters
an
control
percent
prison
official
officials
groups
second
third
three
five
two
six
seven
eight
#$n
too
u.n
former
bosnian
republican
muslim
israeli
russian
serb
his
's
power
town
force
members
leaders
first
four
half
just
federal
government
army
economy
record
school
pay
support
and
high
#n
nearly
one
time
court
judge
hospital
numbers
elections
campaign
which
budget
united_states
soldiers
troops
a_lot
some
only
at_least
more_than
help
about
way
likely
states
countries
peace
men
serbs
women
children
companies
something
things
thousands
number
many
several
most
a_few
americans
workers
victims
people
place
news
trial
vote
agreement
issues
candidate
candidates
those
capital
jobs
job
office
committee
palestinian
democratic
gop
opposition
best
top
issue
show
association
study
industry
center
voters
international
republicans
democrats
american
#an
other
major
recent
small
old
black
white
possible
little
same
no
any
another
every
local
daily
food
foreign
simpson
war
young
own
long
big
good
us
me
county
defense
white_house
free
john
plans
car
company
woman
family
again
your
whose
new
a_new
<unknown>
<proper_noun>
talks
fighting
killing
state
health
future
public
refused
agreed
expected
deal
plan
work
past
house
senate
city
country
meeting
night
university
security
death
world
face
charged
spokesman
morning
money
back
home
to
even
congress
nation
bosnia
america
statement
charges
investigation
according
visit
decision
market
group
would
can
might
has
had
information
u.s.
began
increase
end
attacks
attack
fire
clear
right
case
program
bob
#na
could
will
can't
've
have
israel
china
russia
taiwan
japan
washington
london
tax
administration
president_clinton
los_angeles
later
earlier
used
homes
snow
along
with
through
area
law
system
party
mrs
must
didn't
doesn't
going_to
should
don't
clinton
texas
california
new_york
outside
until
over
programs
problems
bill
david
brown
alexander
new_hampshire
iowa
both
after
before
despite
like
near
across
drug
dole
buchanan
presidential
but
although
when
while
such_as
between
under
business
spending
primary
election
how
though
as
including
at
stop
called
change
if
whether
without
not
keep
got
won
lost
off
here
there
why
because
against
for
use
around
or
then
believe
think
say
said
says
with
get
put
enough
in
tool
make
held
take
to_do
made
on
next
making
give
see
let
know
're
'm
are
were
still
also
already
report
newspaper
himself
once
now
really
each
so
reported
saying
announced
released
last
left
told
asked
do
did
is
has_been
would_be
was
being
became
become
times
such
ever
i
however
killed
arrested
died
early
late
out
wanted
need
want
does
ago
tried
trying
may
be
been
accused
saturday
friday
tuesday
today
wednesday
monday
sunday
thursday
based
down


part
miles
hours
tv
television
trade
more
sen
much
radio
weeks
years
month
prime_minister
found
man
press
military
return
president
he
father
never
evidence
mother
service
chairman
wife
national
reports
leader
director
who
life
police
control
republican
former
bosnian
muslim
percent
prison
official
officials
u.n
israeli
power
groups
second
third
first
his
town
members
federal
government
army
economy
force
leaders
school
pay
support
and
high
time
court
judge
hospital
elections
campaign
which
soldiers
troops
budget
united_states
help
numbers
serbs
women
?
#r
way
likely
states
peace
men
children
countries
americans
people
workers
place
news
capital
jobs
trial
issues
candidate
vote
agreement
voters
issue
show
international
association
study
center
republicans
industry
democrats
american


police
reporters
#o
official
officials
groups
members
leaders
first
second
third
three
five
two
six
four
#n
half
high
and
one
just
nearly
more_than
at_least
only
some
a_lot
soldiers
troops
budget
united_states
help
about
serbs
women
something
thousands
peace
men
children
companies
things
number
many
most
several
a_few
americans
people
workers
capital
jobs
job
committee
office
democratic
gop
opposition
best
top
candidates
voters
international
republicans
democrats
american
#an
other
major
recent
small
black
white
possible
little
another
every
any
old
same
no


recent
young
own
long
big
good
us
me
a_new
john
plans
car
family
whose
new
company
woman
again
your
<unknown>
<proper_noun>
refused
led
agreed
deal
work
past
night
spokesman
morning
expected
money
back
home
to
even
according
visit
decision
market
group
would
can
might
clear
right
case
program
bob
#na
could
've
attack
party
snow
along
through
area
law
system
mrs
didn't
doesn't
going_to
outside
programs
must
should
david
until
over
problems
bill
don't
near
across
drug
such_as
between
under
business
stop
at
called
change
without
keep
not
use
lost
off
here
there
around
for
with
get
put
enough
in
took
make
held
to_do
take
next
making
give
see
let
know
're
'm
are
were
last
left
told
asked
do
did
is
has_been
would_be
early
late
out
wanted
does
was
being
became
become
need
want
tried
may
trying
been


foreign
food
john
county
defense
white_house
free
talks
fighting
killing
public
state
health
future
meeting
led
city
world
night
country
security
death
face
charged
spokesman
university
statement
morning
congress
nation
bosnia
america
clear
right
began
increase
attacks
end
information
u.s.
homes
snow
along
taiwan
russia
japan
london
washington
tax
used
outside
president_clinton
los_angeles
until
clinton
texas
california
new_york
after
before
near
brown
alexander
new_hampshire
despite
like
across
iowa
both
but
although
while
such_as
between
dole
buchanan
presidential
when
under
how
though
at
primary
election
as
including
without
if
whether
why
against
not
because
for
believe
or
then
think
said
in
held
:
also
made
on
next
already
reported
report
announced
last
left
now
so
saying
newspaper
himself
once
really
released
early
each
however
killed
arrested
late
out
times
i
ever
died
such
ago
accused
saturday
today
tuesday
friday
based
sunday
thursday
speech

From UNI-SNE to t-SNE

- Laurens van der Maaten started experimenting on UNI-SNE and we soon realised that it was easier to replace the mixture of a Gaussian and a uniform by an infinite mixture of Gaussians:

$$q_{ij} \propto \frac{1}{1+d_{ij}^2}$$

- By using a Gaussian to compute p_{ij} and a heavy-tailed student's t to compute q_{ij} we partially compensate for the different rates of growth of volume as you move away from a point in 2-D and in N-D.

t-SNE

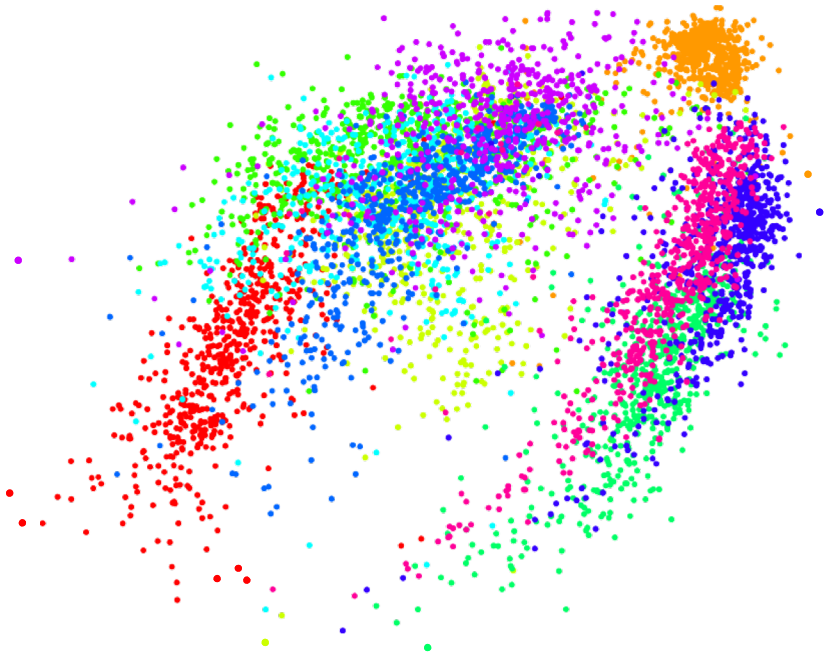
- Instead of using a gaussian plus a uniform, why not use gaussians at many different spatial scales?
  - This sounds expensive, but if we use an infinite number of gaussians, its actually cheaper because we avoid exponentiating.

$$q_{ij} \propto \frac{1}{1+d_{ij}^2}$$

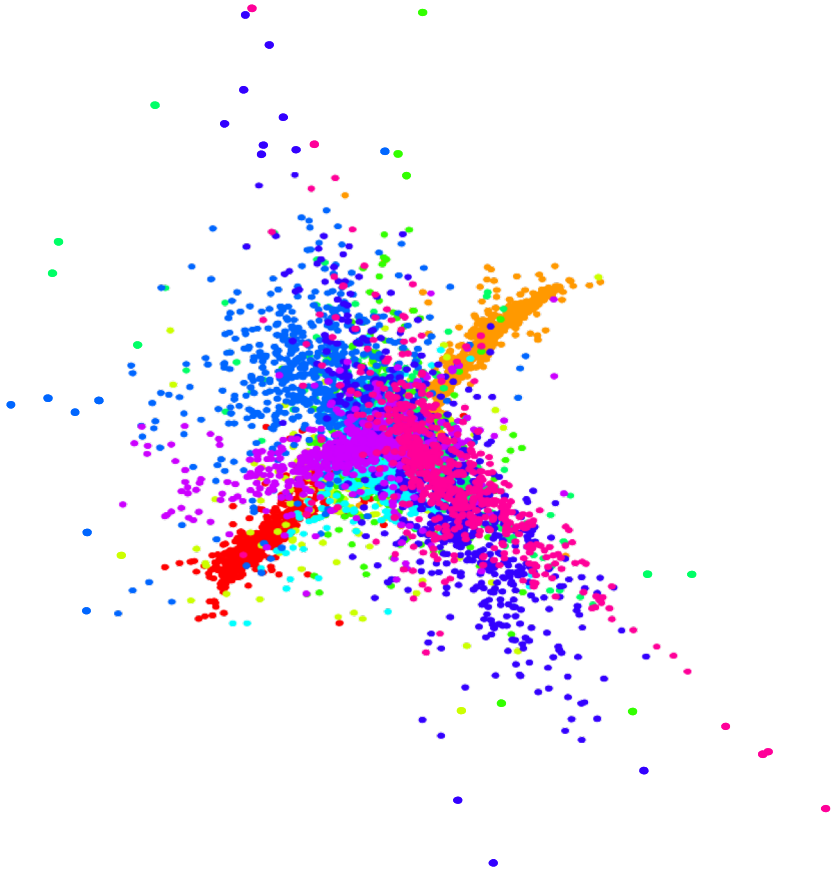
Optimization hacks

- Reputable hack: Introduce a penalty term that keeps all the map-points close together.
  - Then gradually relax the penalty to break symmetry slowly.
- Disreputable hack: Allow the probabilities to add up to 4.
  - This causes the map-points to curdle into small clusters leaving lots of space for clusters to move past each other.
  - Then make the probabilities add up to 1.

Two other state-of-the-art dimensionality reduction methods on the 6000 MNIST digits

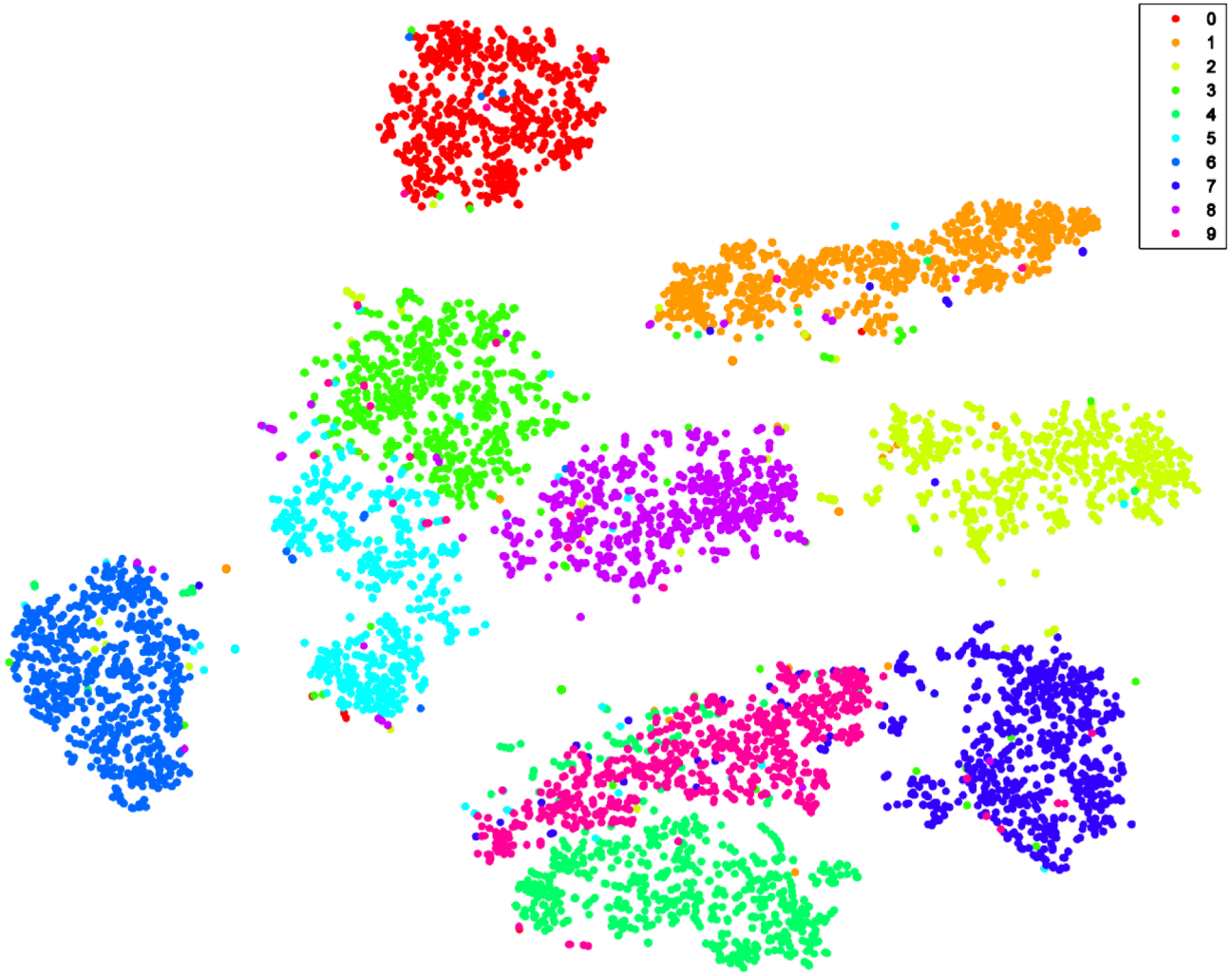


Isomap



Locally Linear Embedding

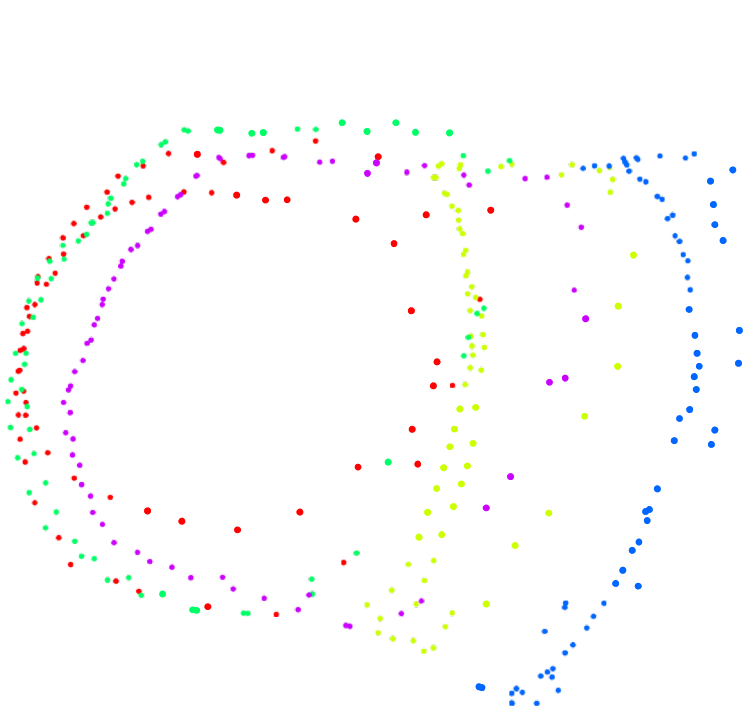
t-SNE on the 6000 MNIST digits



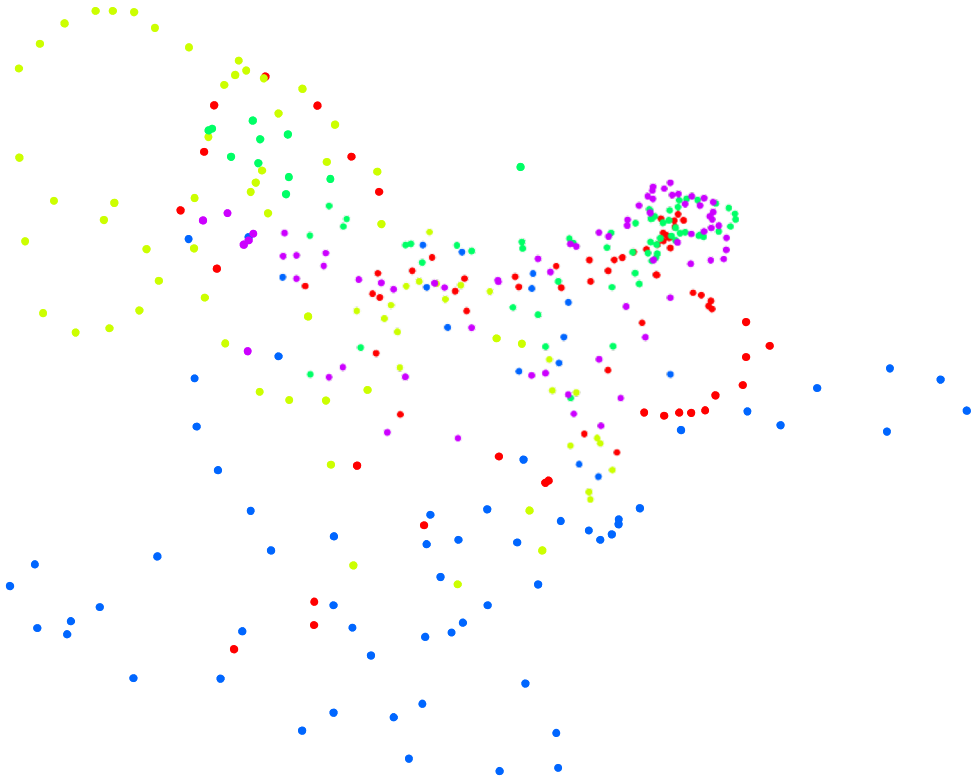
The COIL20 dataset

Each object is rotated about a vertical axis to produce a closed one-dimensional manifold of images.

Isomap & LLE for COIL20 dataset

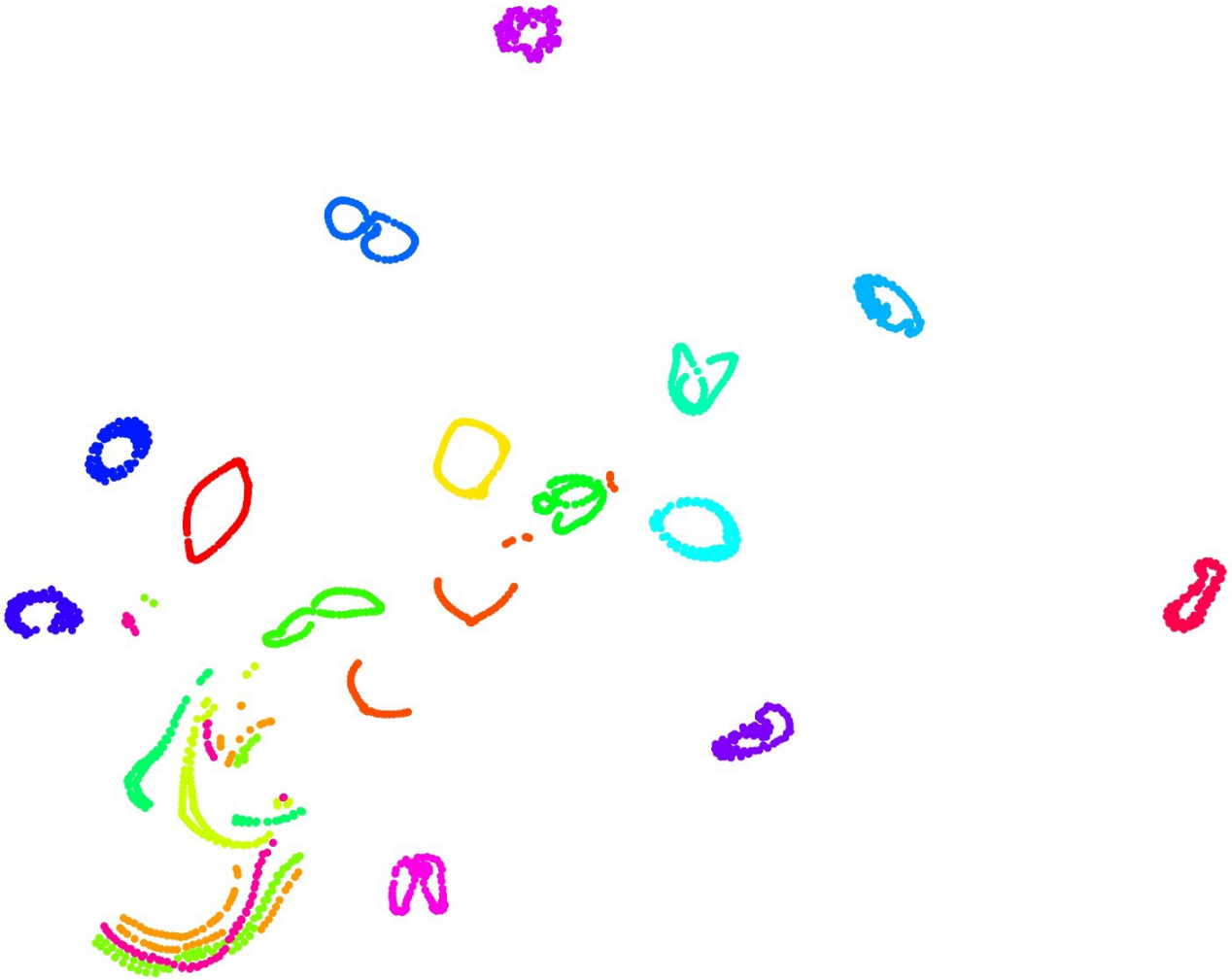


Isomap



Locally Linear Embedding

t-SNE for COIL20 dataset



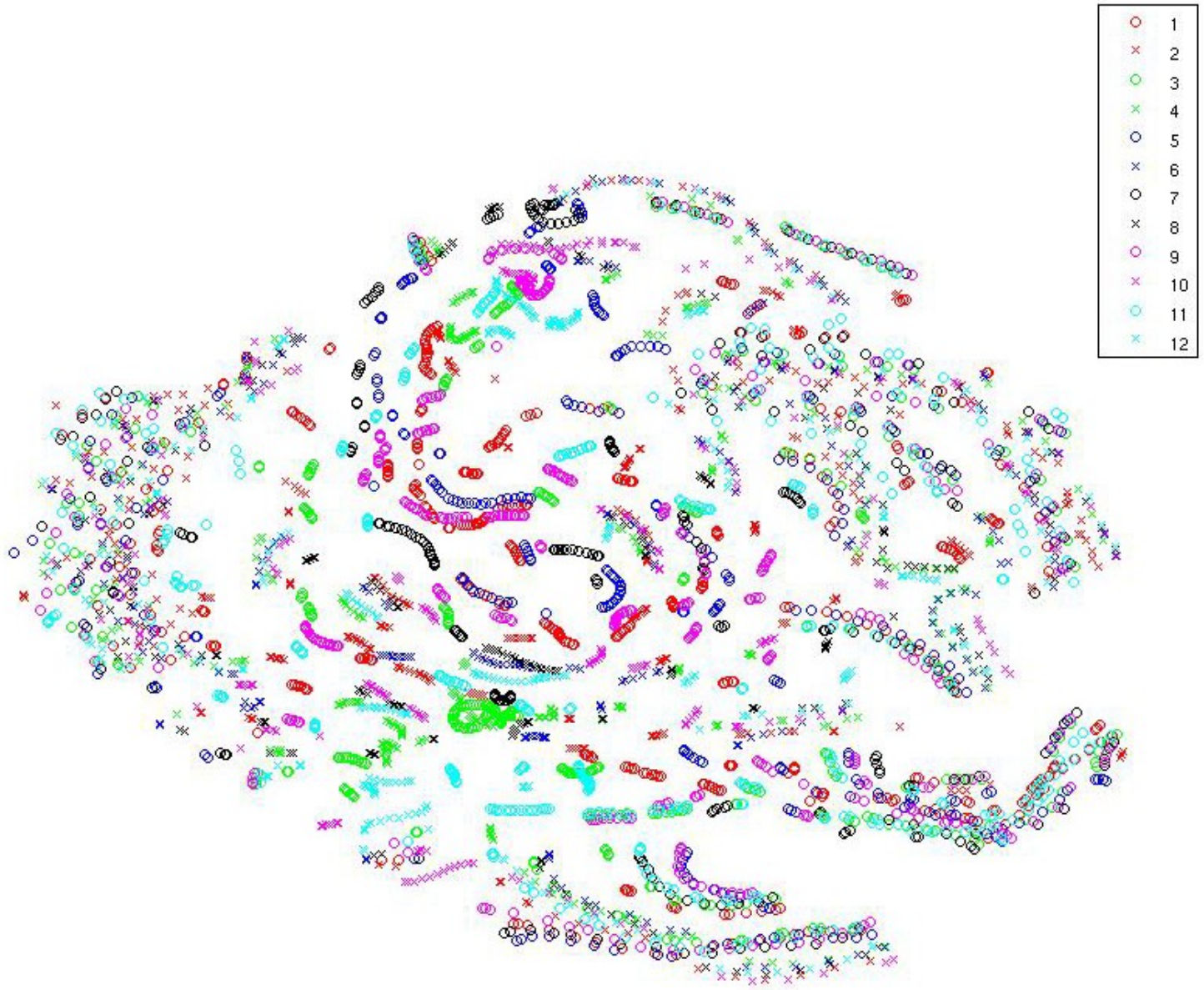
Show the map of 2000 English words
produced by Joseph Turian using
t-SNE on the feature vectors learned by
Colobert and Weston (ICML 2008)

Using t-SNE to see what you are thinking

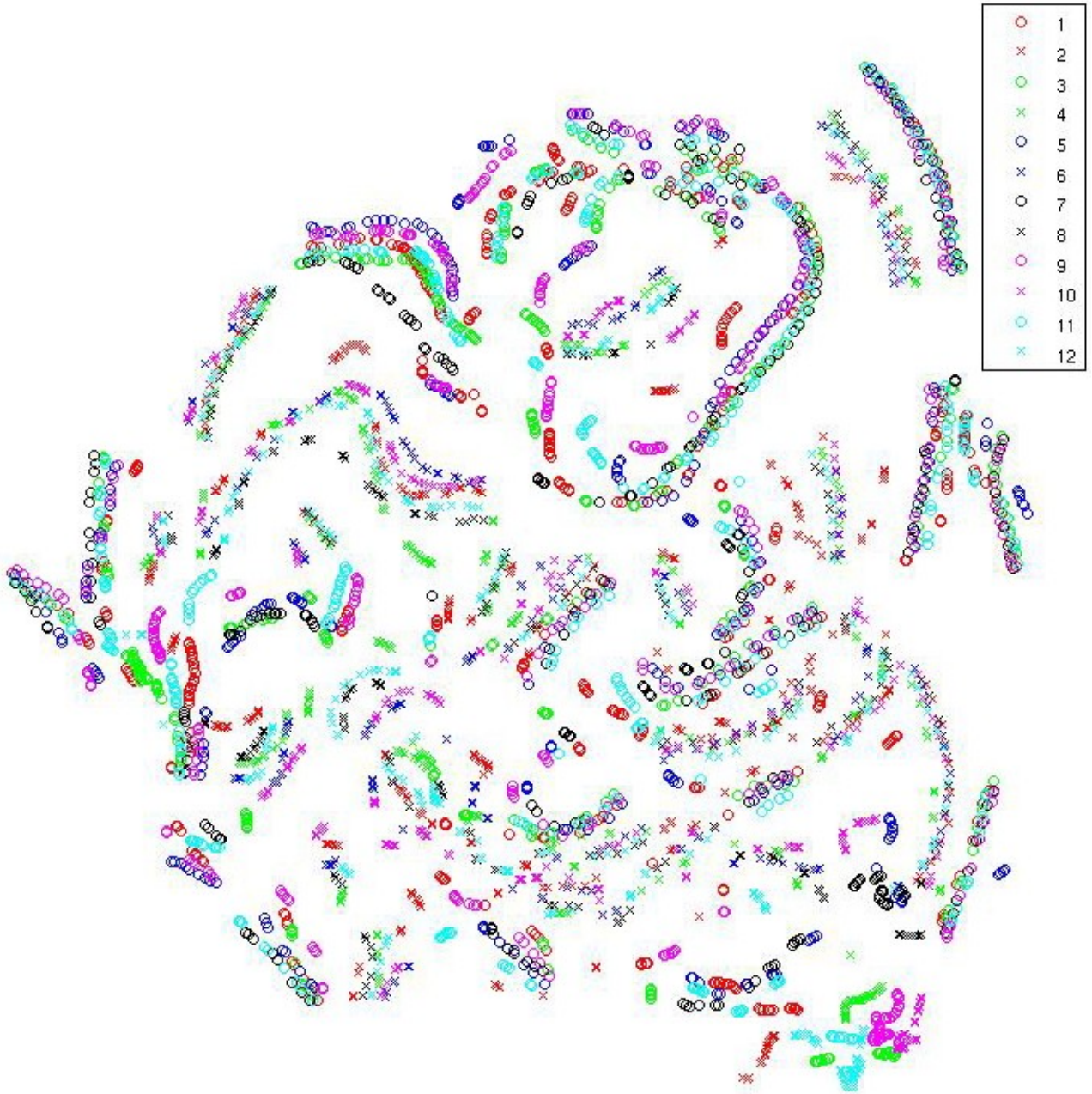
Using t-SNE to see how a DBN that recognizes phonemes deals with speaker variation

- Current speech recognizers try to preprocess the input to eliminate differences between speakers.
- We get very good performance from a DBN without doing this.
  - Its very difficult to improve the DBN by adding speaker information.
- Maybe the DBN is using its hidden layers to get rid of speaker variation .
  - This would explain why speaker information is not much help.

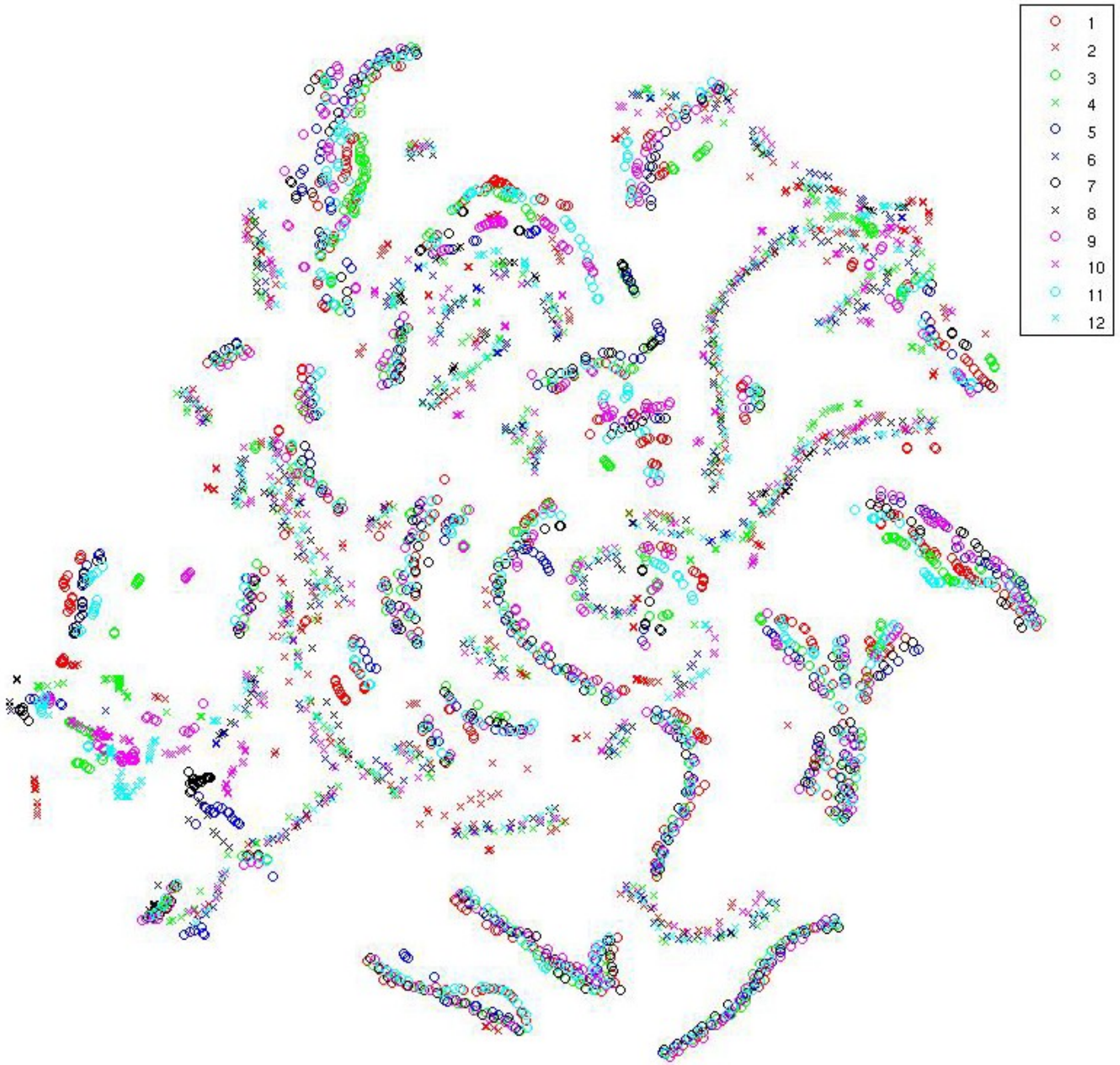
t-SNE applied to windows of 15 input frames for 6 speakers saying two sentences



t-SNE applied to the first hidden layer



t-SNE applied to the 8th hidden layer



Some recent developments

- Miguel Carreira-Perpinan (ICML 2010) showed that the original SNE cost function can be rewritten so that it is equivalent to Laplacian Eigenmaps with an extra repulsion term that spreads out the map points (like in MVU).

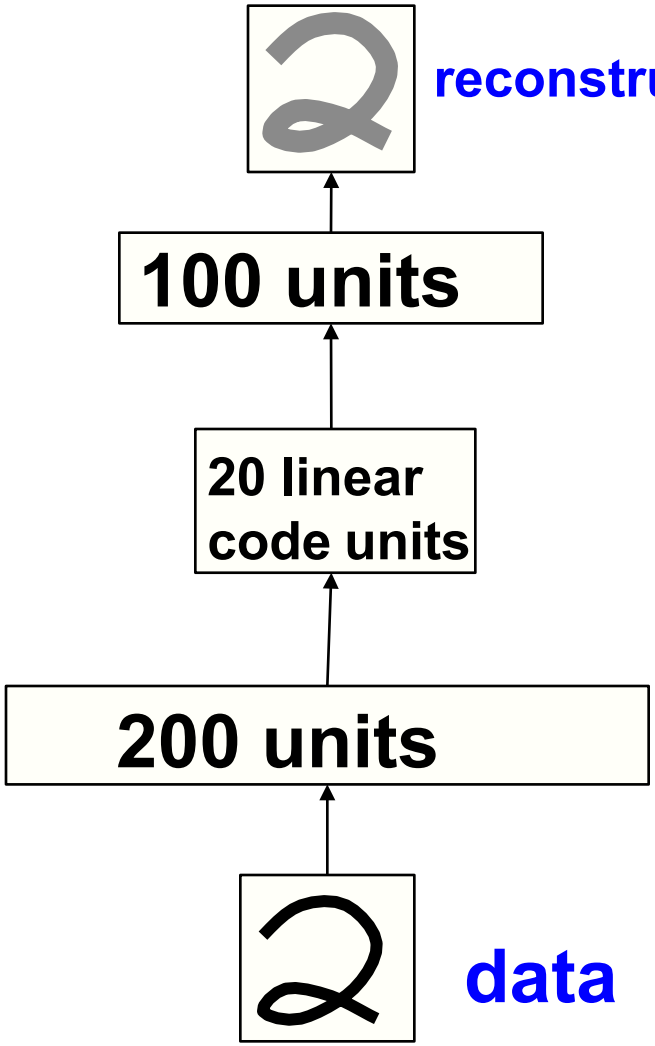
- This led to a much faster optimization method. The fast code is now on the t-SNE webpage.

Combining non-parametric dimensionality reduction with neural networks

- If we have a smooth objective function that assigns a value to a set of codes, we can backpropagate its derivatives through a feedforward neural net.
  - The neural net is like the “encoder” part of an autoencoder.
- We could combine this objective function with the objective of getting good reconstructions from the codes.

Evaluating the codes found by an autoencoder

- Use 3000 images of handwritten digits from the USPS training set.
  - Each image is 16x16 and fairly binary.
- Use a highly non-linear autoencoder
  - Use logistic output units and linear code units.

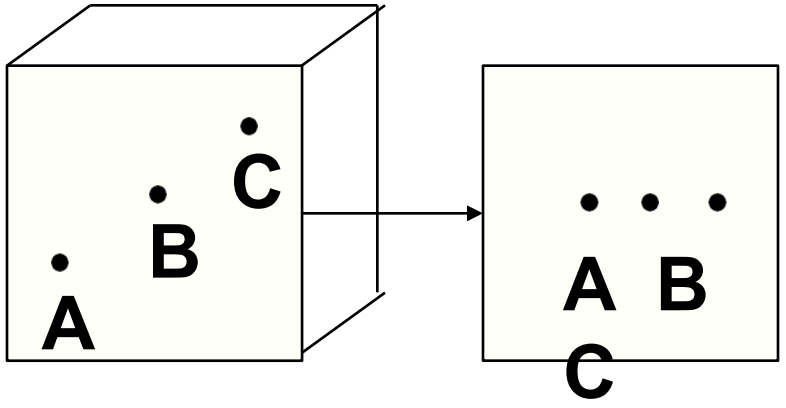


Does code space capture the structure of the data?

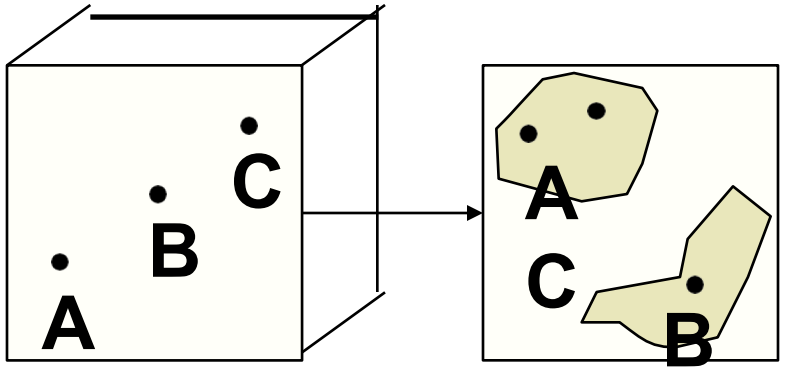
- We would like the code space to model the underlying structure of the data.
  - Digits in the same class should get closer together in code space.
  - Digits of different classes should get further apart.
- We can use k nearest neighbors to see if this happens.
  - Hold out each image in turn and try to label it by using the labels of its k nearest neighbors.
- In pixel space we get 5.9% errors. In code space we get about 12% errors. Why doesn't it work?

How it goes wrong

- PCA is not powerful enough to really mangle the data.



- Non-linear auto-encoders can fracture a manifold into many different domains.
  - This can lead to very different codes for nearby data-points.

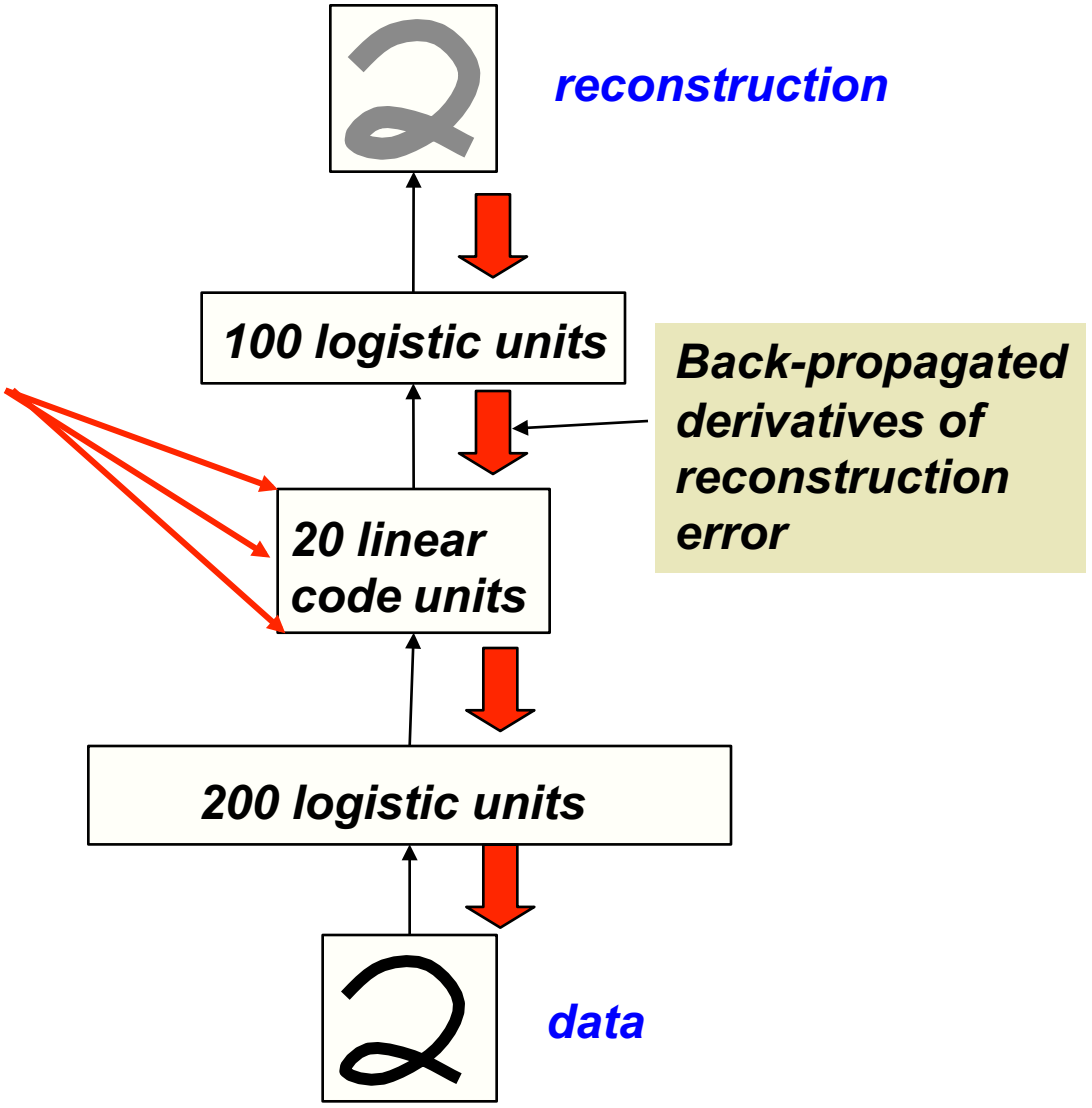


How to fix it

- We need a regularizer that will make it costly to fracture the manifold.
  - There are many possible regularizers.
- Stochastic neighbor embedding can be used as a regularizer.
  - Its like putting springs between the codes to prevent the codes for similar datapoints from being too far apart.

How the gradients are combined

Forces generated by springs attaching this code to the codes for all the other data-points. The stiffness of each spring is dynamically set to be: $p_{ij} - q_{ij}$

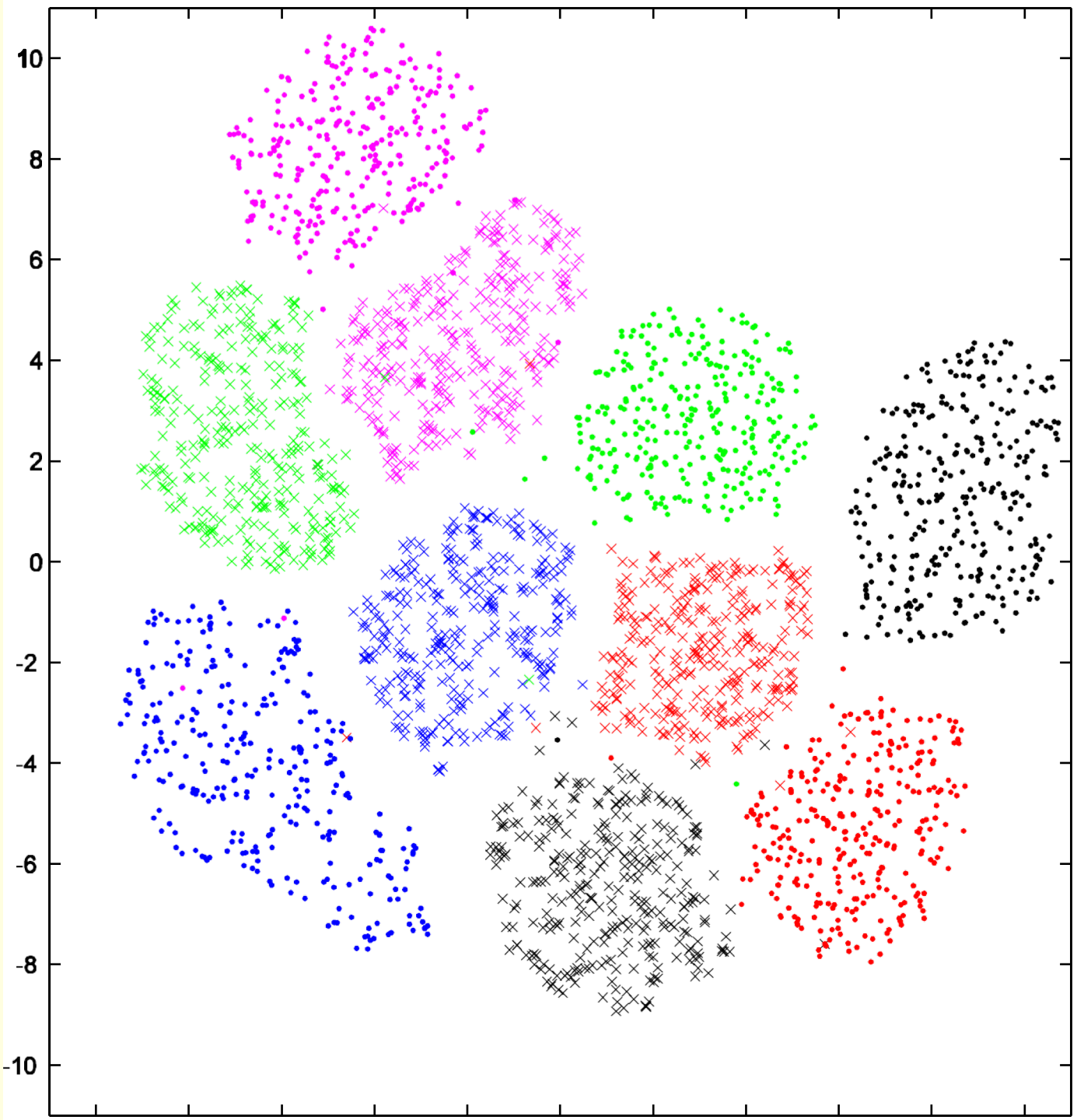


How well does it work?

- Instead of rising from 5.9% to 12%, the hold-one-out KNN error falls to about 2%
  - The strength of the regularizer must be chosen sensibly.
  - The SNE regularizer alone gives about 4.5% hold-one-out KNN errors.
- Can we visualize the codes that are produced using the regularizer?

Learning codes with some pairwise information about labels

- If we pair each digit with another of the same class, there are very nice category boundaries between digits.


10
8
6
4
2
0
-2
-4
-6
-8
-10

A more efficient version

- The derivatives that come from the autoencoder will stop the codes from all collapsing to a point.
  - So we don't need the quadratically expensive normalization term that is used in computing the qij's
- We should be able to just add attractive forces between codes that correspond to similar inputs.
  - The similarity could be measured in the input space or it could be based on extra information (e.g. identity of class labels).

Non-linear Neighborhood Components Analysis (Salakhutdinov and Hinton, 2007)

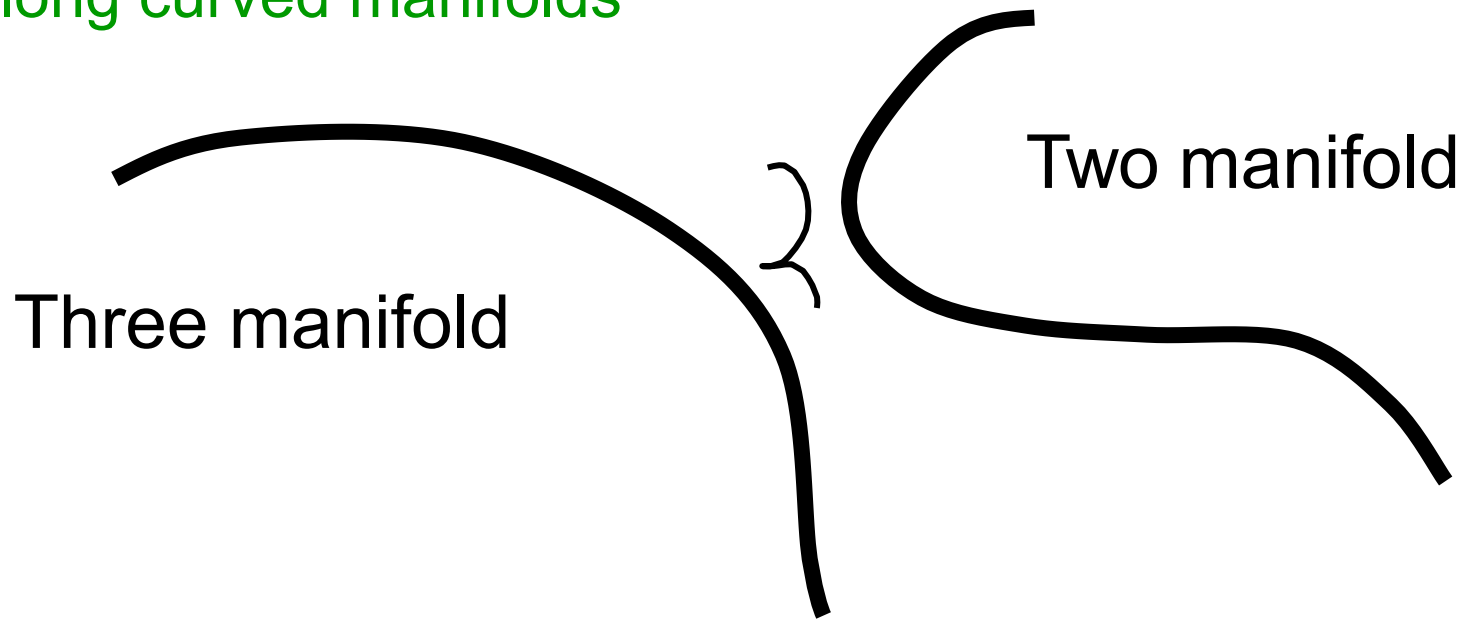
- Use a feed forward neural net to learn codes that make nearest neighbor classification work well.
- In code space, we can predict the class of a point by summing the probabilities assigned to other points of that class when we use the stochastic neighbor probabilities.

$$p(class(i) = c) = \sum_{j \in c} p_{j|i}$$

- So we do gradient ascent in the log probabilities of getting the correct class for each point when it is held-out

What NNCA achieves

- Linear Discriminant Analysis tries to find a projection of the data that makes each point be close to other points of the same class and far from other points of different classes.
  - This is a bad objective function if the classes form long curved manifolds



NCA

- NCA is the linear version of NNCA. It aims to find projections in which each datapoint is close to some other datapoints of the same class and not too close to datapoints of other classes.
  - This does not force all points of the same class to be similar.
  - But it can allow manifolds to become disconnected.