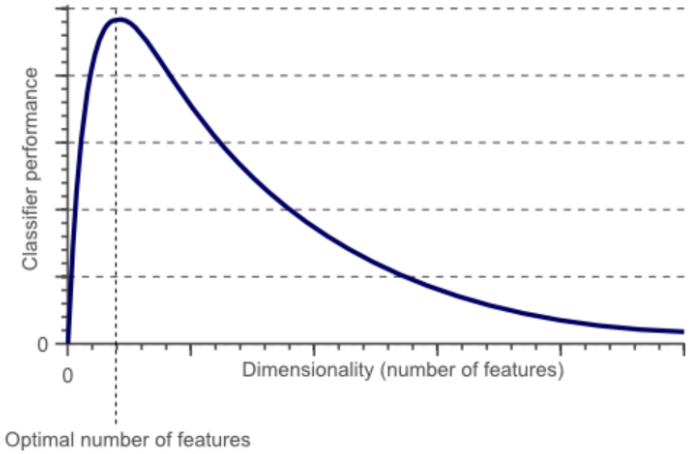
DIMENSIONALITY REDUCTION ALGORITHMS

--- DEBBRATA SAHA

CURSE OF DIMENSIONALITY

- Overfitting
  - The more features we use, the more complex a model gets
  - Eventually cause the model to fit the training data too well, causing overfitting.
  - Unable to predict the real life test data accurately



The Curse of Dimensionality

DIMENSIONALITY REDUCTION

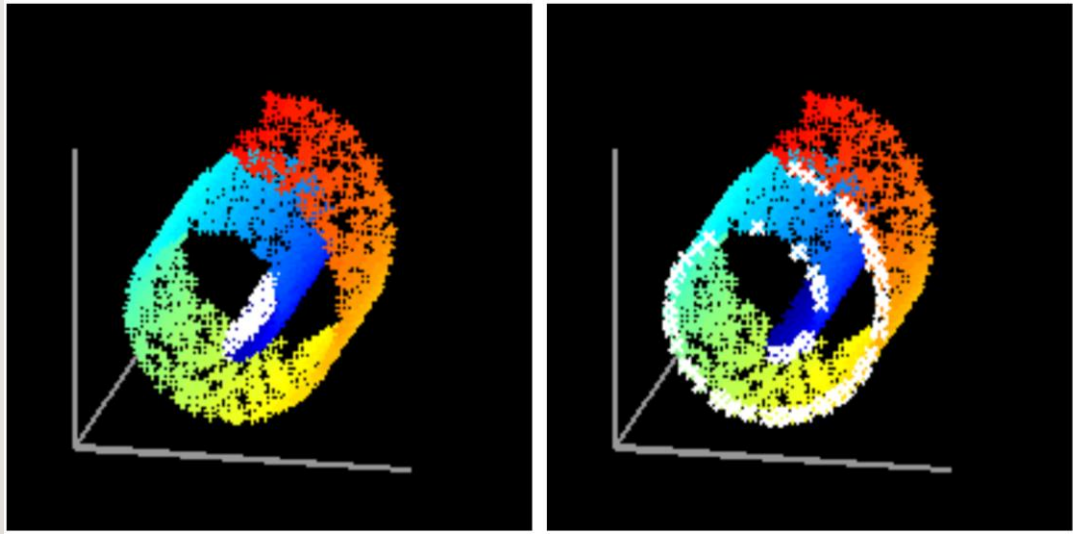
- Less dimensions mean less computing. Less data means that algorithms train faster.
- Less data means less storage space required
- Removes redundant features and noise
- Helps to remove the overfitting
- Less misleading data means model accuracy improves

DIMENSIONALITY REDUCTION ALGORITHMS

- Linear
  - Principle Component Analysis (PCA)
  - Multi dimensional scaling (MDS)
- Non linear
  - ISOMAP
  - Locally Linear Embedding(LLE)
  - T-SNE

LOCALLY LINEAR EMBEDDING (LLE)

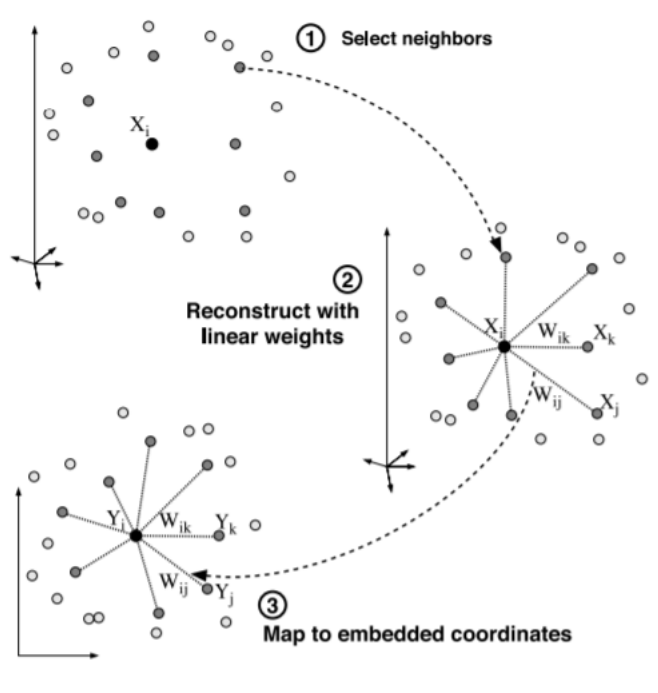
- Unsupervised method for dimensionality reduction
- Reduce dimensions while trying to preserve the geometric features of the original non-linear feature structure



Source : https://cs.nyu.edu/~roweis/lle/swissroll.html

Nonlinear Dimensionality Reduction by Locally Linear Embedding
Sam T. Roweis and Lawrence K. Saul
www.sciencemag.org SCIENCE VOL 290 22 DECEMBER 2000

LLE



$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2 \qquad (1)$$

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j \right|^2 \qquad (2)$$

LLE

- Solve small pieces at a time like piecewise regression
- Have to optimize two objective functions

ISOMAP

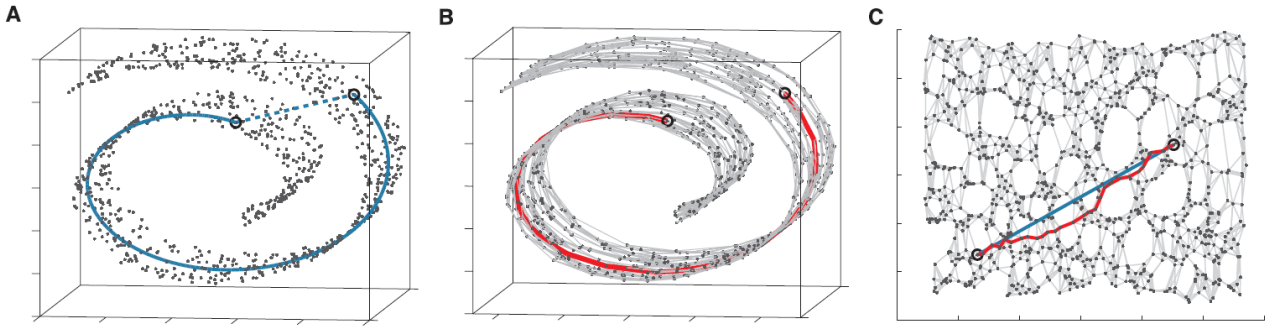


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

A Global Geometric Framework for Nonlinear Dimensionality Reduction

Joshua B. Tenenbaum,* Vin de Silva, John C. Langford

www.sciencemag.org SCIENCE VOL 290 22 DECEMBER 2000

ISOMAP

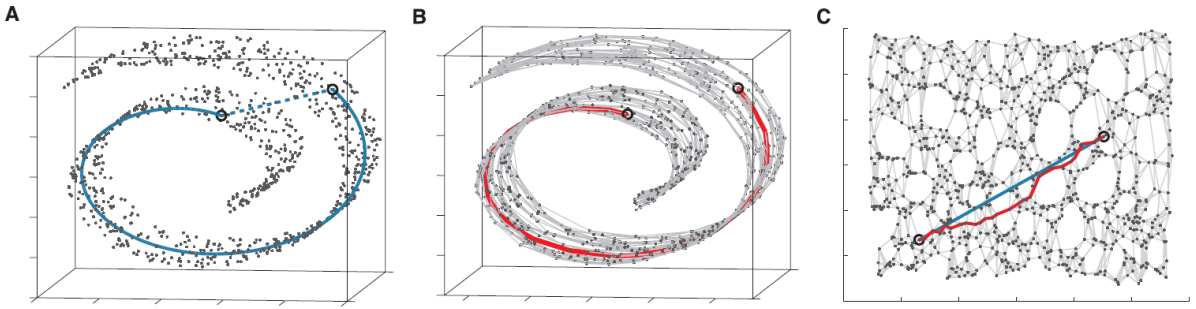


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

Step		
1	Construct neighborhood graph	Define the graph G over all data points by connecting points i and j if [as measured by $d_X(i,j)$] they are closer than ϵ (ϵ-Isomap), or if i is one of the K nearest neighbors of j (K-Isomap). Set edge lengths equal to $d_X(i,j)$.
2	Compute shortest paths	Initialize $d_G(i,j) = d_X(i,j)$ if i,j are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G = \{d_G(i,j)\}$ will contain the shortest path distances between all pairs of points in G (*16*, *19*).
3	Construct d-dimensional embedding	Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (*17*), and v_p^i be the i-th component of the p-th eigenvector. Then set the p-th component of the d-dimensional coordinate vector $\mathbf{y}_i$ equal to $\sqrt{\lambda_p} v_p^i$.

Objective function:

$$E = \|\tau(D_G) - \tau(D_Y)\|_{L^2}$$

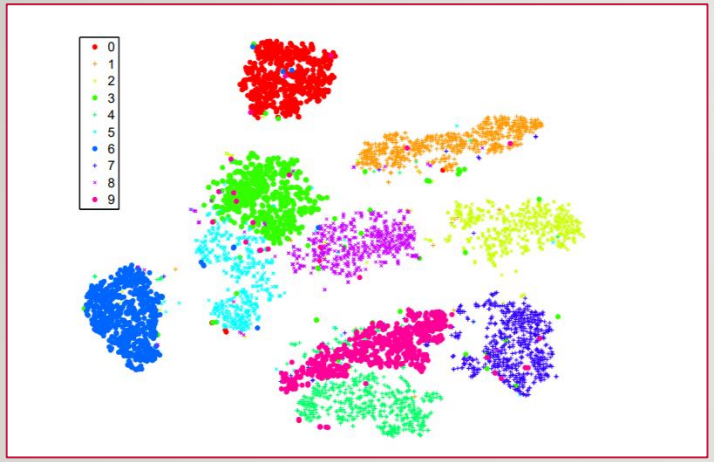
T-SNE (T-DISTRIBUTED STOCHASTIC NEIGHBOR EMBEDDING)

Visualizing Data using t-SNE

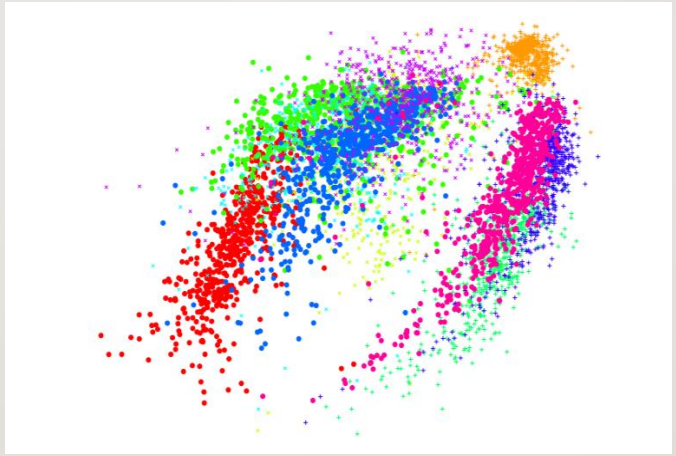
Laurens van der Maaten, Geoffrey Hinton

Journal of Machine Learning Research 9 (2008) 2579-2605

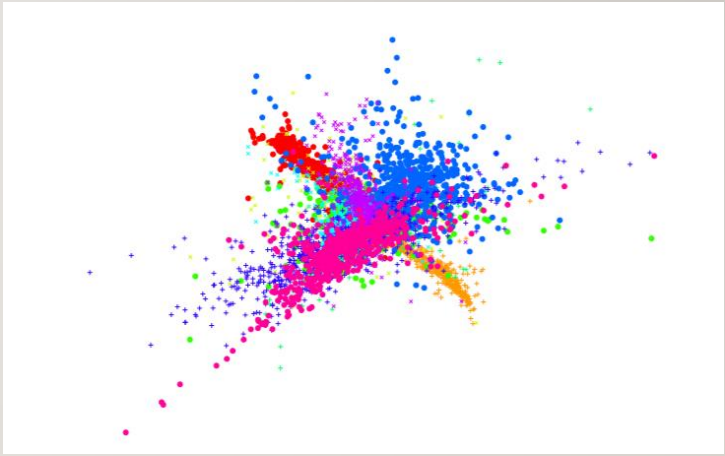
MNIST DATASET LAYOUT



Visualization by t-SNE

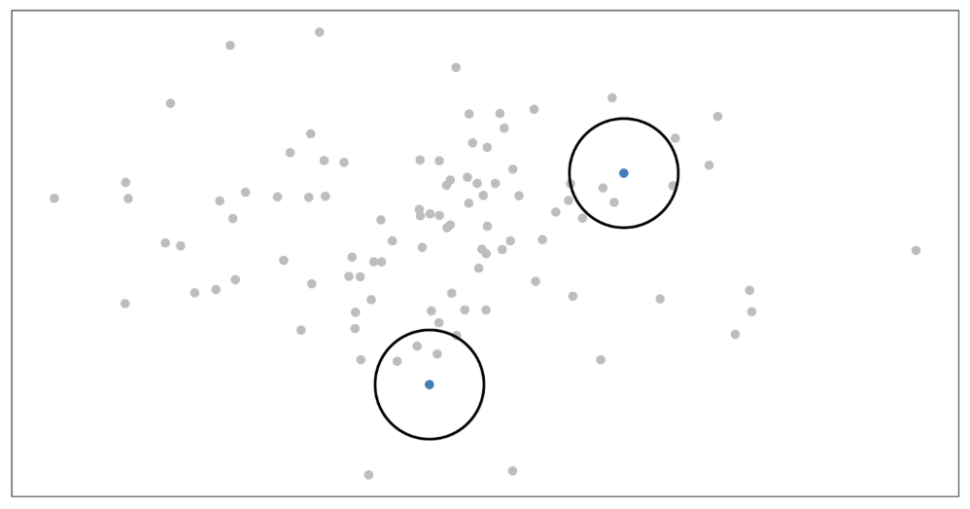


Visualization by ISOMAP



Visualization by LLE

T-SNE PROCEDURE



$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)},$$

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}.$$

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

T-SNE ALGORITHM

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity *Perp*,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity *Perp* (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for t=1 **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  end
end

DIVIDE AND CONCUR

- A method for solving constraint problems
- Take advantage of the division into sub problems
- Works iteratively and deterministically

Divide and concur: A general approach to constraint satisfaction

Simon Gravel and Veit Elser

Phys. Rev. E **78**, 036706 – Published 22 September 2008

METHOD

- Each iterative step is defined by two fundamental operations
- In the first operations, the problem is divided into its constituent constraints
- Each constraint is solved independently, ignoring possible conflicts between different constraints
- In the second operation, conflicts between constraints are resolved by concur operation

PROCEDURE

- Suppose Y has n constraints.
- Create n replicas for n constraints (, ,)

- **Divide Step:**

- Apply associated projection on replica which acts separately on each of the replicas

$$P_D(\mathbf{y}) = P_1(\mathbf{x}^{(1)}) \otimes P_2(\mathbf{x}^{(2)}) \otimes \cdots \otimes P_N(\mathbf{x}^{(N)})$$

- **Concur Step.**

- Replaces the value of each replica by the average value of all the replicas.

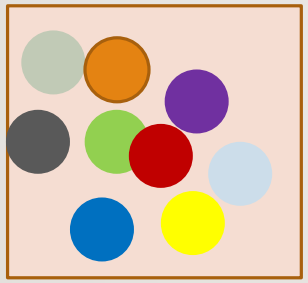
$$P_C(\mathbf{y}) = \bar{\mathbf{x}} \otimes \bar{\mathbf{x}} \otimes \cdots \otimes \bar{\mathbf{x}}$$

SPHERE PACKING PROBLEM

- Packing of n spheres in a finite D-dimensional volume
- Each sphere must avoid n - 1 other spheres and lie within a certain volume
- There are altogether n constraints per sphere
- Create n replicas for each sphere

SPHERE PACKING PROBLEM

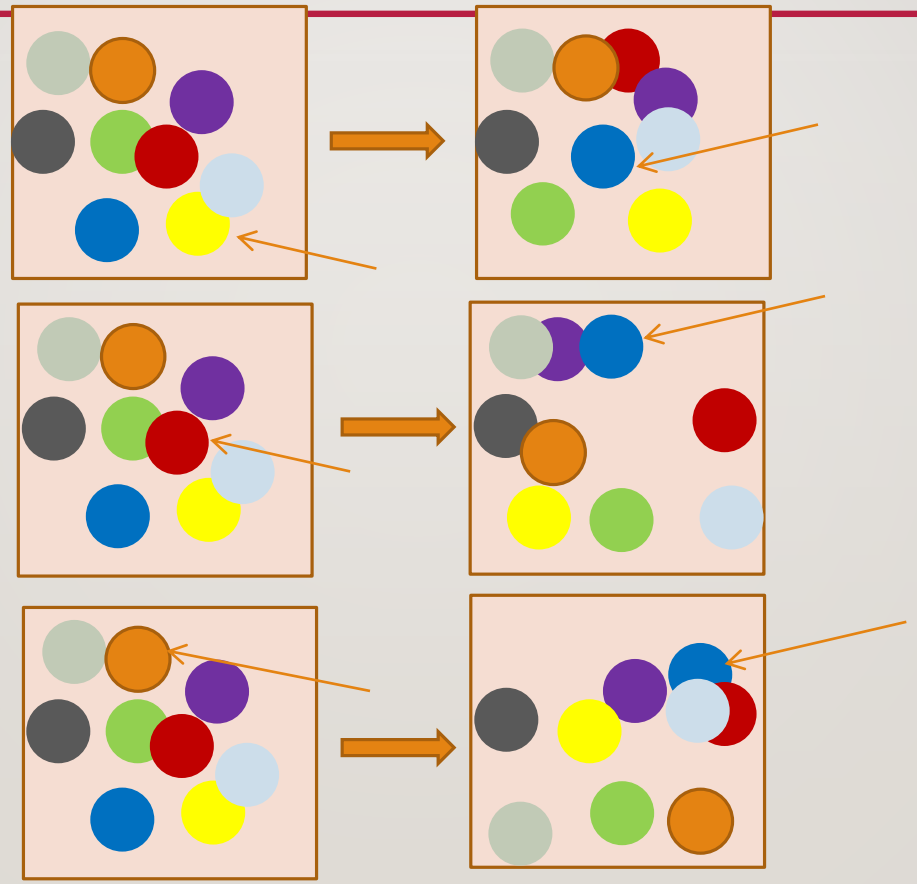
- Divide step:



- Spheres = 9
- 9 replicas for each sphere

SPHERE PACKING PROBLEM

Divide Step:



SPHERE PACKING PROBLEM

- Concur step: Replaces the value of each replica by the average value of all the replicas