Neighborhood Component Analysis

Presented by Youyou Wang
University of Missouri-Columbia

Paper

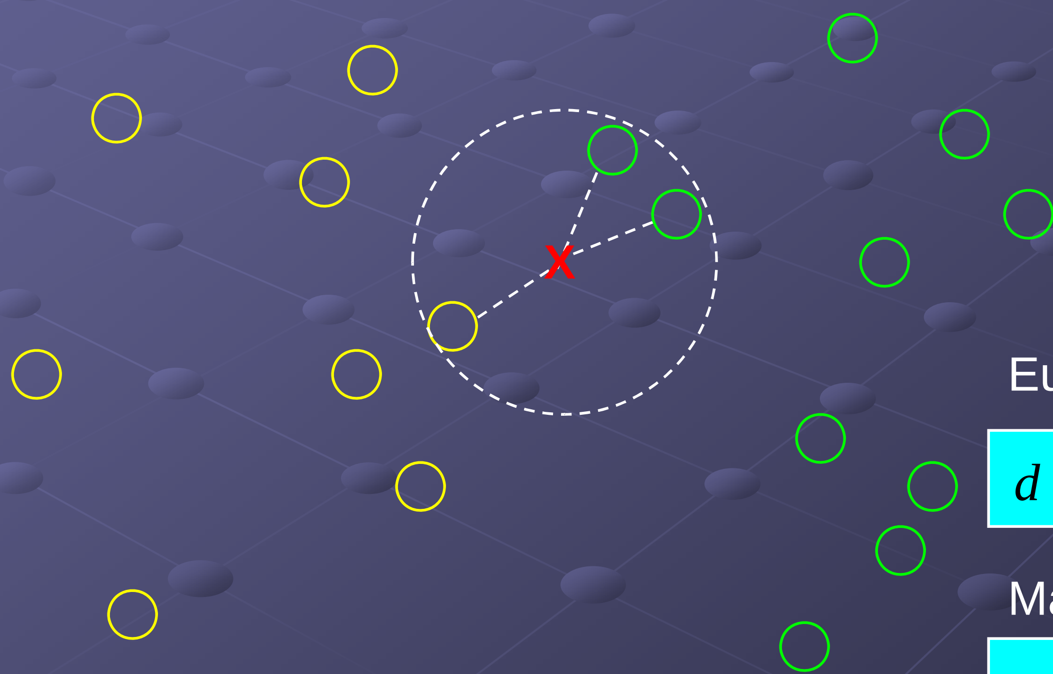
- “Neighborhood Components Analysis”

  J. Goldbergerm,S. Roweis, G. Hinton, R. Salakhutdinov

  University of Toronto, In Advances in Neural Information Process System 17, 2004

K-Nearest Neighbor

- The ***k*-nearest neighbor algorithm** is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors.



Euclidean:

$$d(x, y) = \sqrt{\left(x^2 + y^2\right)}$$

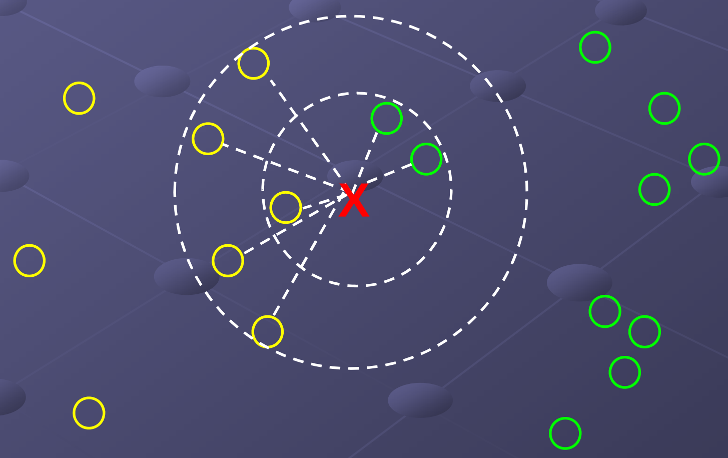
Mahalanobis:

$$d(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}$$

Advantage and Disadvantages

- Advantages:
- Simple
- The decision surfaces are nonlinear
- The quality of the predictions automatically improve as the amount of training data increases

- Disadvantages:
- The computational load of the classifier is quite high at test time since we must store and search through the entire training set to find the neighbors of a query point before we can do classification.
- We must define what we mean by 'nearest‘

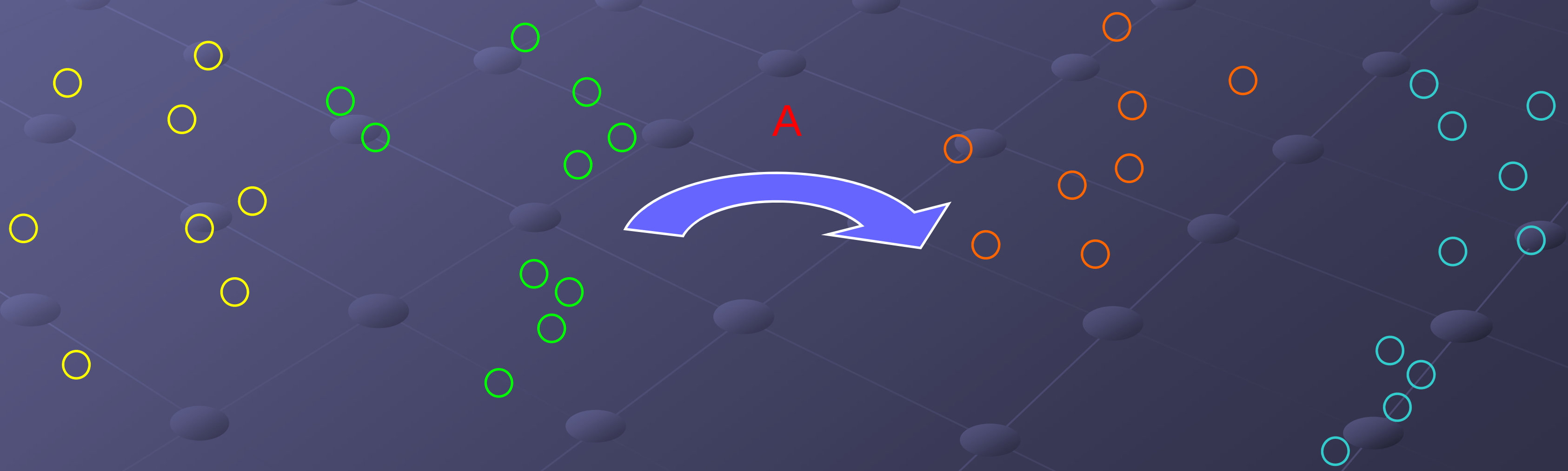


Neighborhood Analysis

- Restrict to find a Mahalanobis distance
  $d(x, y) = (x - y)^T Q (x - y)$

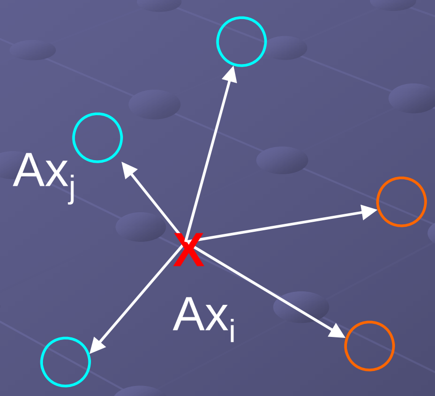
  A symmetric positive semi-define matrices $Q = A^T A$
  $d(x, y) = (Ax - Ay)^T (Ax - Ay)$

- We learn a linear transformation of the input space such that in the transformed space, KNN performs well.



Neighborhood Analysis

The probability for each point i to select point j as its neighbor and inherits its class label, in the transformed



Softmax over Euclidean distance in transformed space

$$p_{ij} = \frac{\exp\left(-\left\|Ax_i - Ax_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\left\|Ax_i - Ax_j\right\|^2\right)}$$

Bridge the discrete representation in KNN to continuous

Neighborhood Analysis

- The probability that point I will be correctly classified:

$$p_i = \sum_{j \in C_i} p_{ij}$$

$$C_i = \{ j \mid c_i = c \quad \}$$

Maximize f(A)

- The expected numbe ctly classified under this sch

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i$$

Neighborhood Analysis

- Differentiate f with respect to the transformation matrix A:

$$\frac{\partial f}{\partial A} = -2A\sum_i \sum_{j\in C_i} p_{ij}\left(x_{ij}x_{ij}^T - \sum_k p_{ij}x_{ij}x_{ij}^T \right)$$

$$\frac{\partial f}{\partial A} = -2A\sum_i \left(p_i \sum_k p_i x_{ij}x_{ij}^T - \sum_{j\in C_i} p_{ij}x_{ij}x_{ij}^T \right)$$

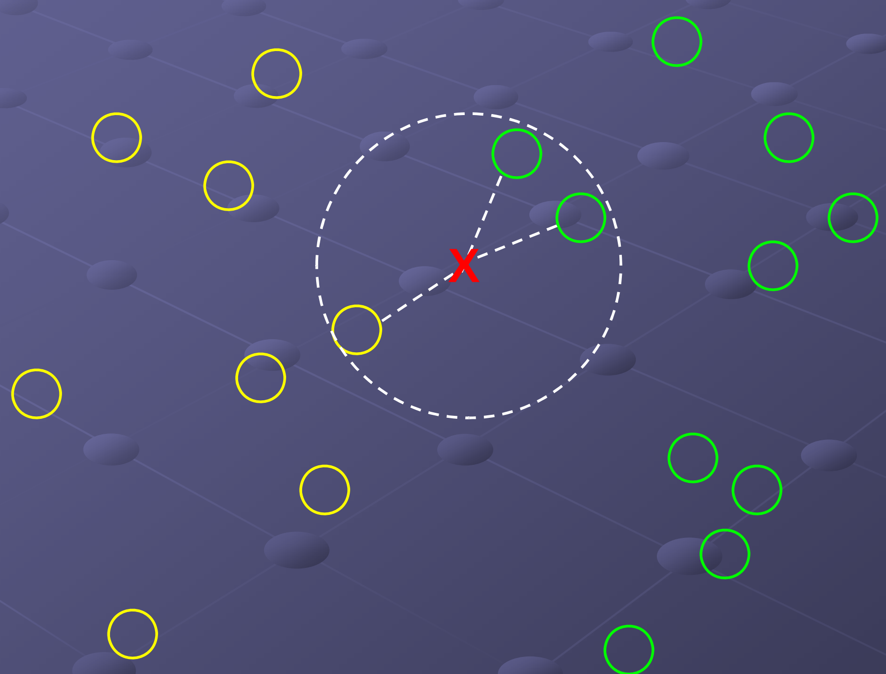
- A Natural Alternative

$$g(A) = \sum_i \log\left(\sum_{j\in C_i} p_{ij} \right) = \sum_i \log\left(p_i\right)$$

$$\frac{\partial g}{\partial A} = -2A\sum_i \left(p_i \sum_k p_i x_{ij}x_{ij}^T - \frac{\sum_{j\in C_i} p_{ij}x_{ij}x_{ij}^T}{\sum_{j\in C} p_{ij}} \right)$$

Discussion

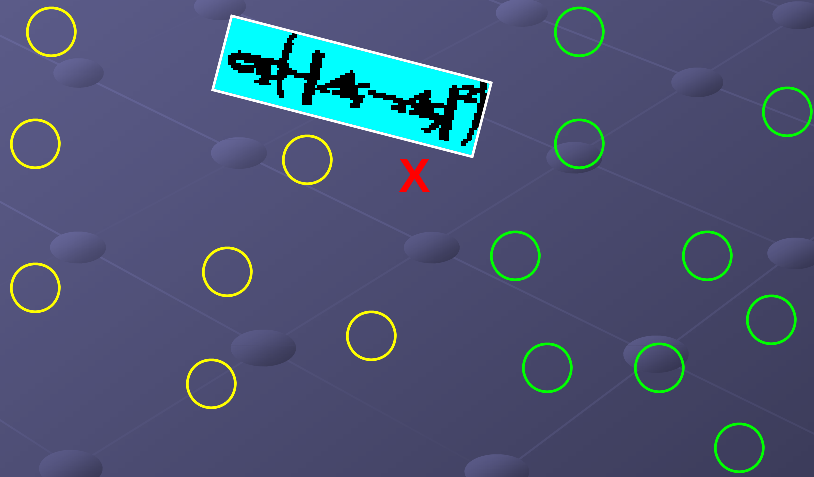
- No over fitting
  - The larger we can drive f during training the better our test performance will be.



A comparison with training data, but not from a learned function

Discussion

- K->p_{ij}
  - By learning the overall scale of A as well as the relative directions of its row we are also effectively learning a real-valued estimate of the optimal number of neighbors.



$$p_{ij} = \frac{\exp\left(-\left\|Ax_i - Ax_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\left\|Ax_i - Ax_j\right\|^2\right)}$$

Discussion

- Dimensionality Reduction
  - Restrict A to be a nonsquare matrix of size d by D, where d << D. Selecting d = 2 or d = 3.

Xlassify a new point xtest by first computing its projection $y_{test} = Ax_{test}$ and then do KNN on y_{test} using the y_n and simple Euclidean metric.

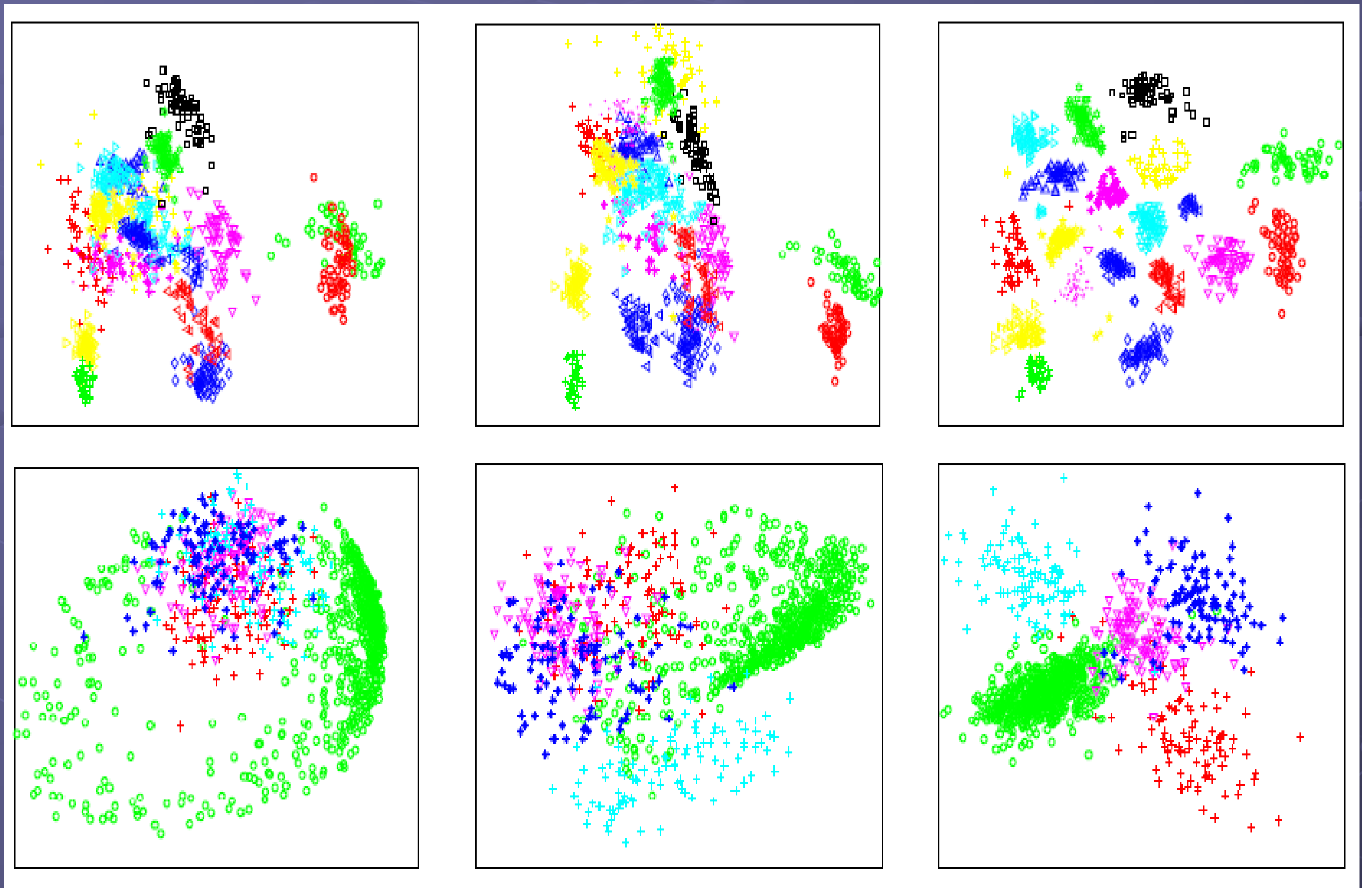
By using KD-tree to increase the speed of search, the storage requirements are O(dN)+Dd compared with O(DN)

Results

UCI database

70% training

30% testing



Faces

d = 560

Digits

d =256

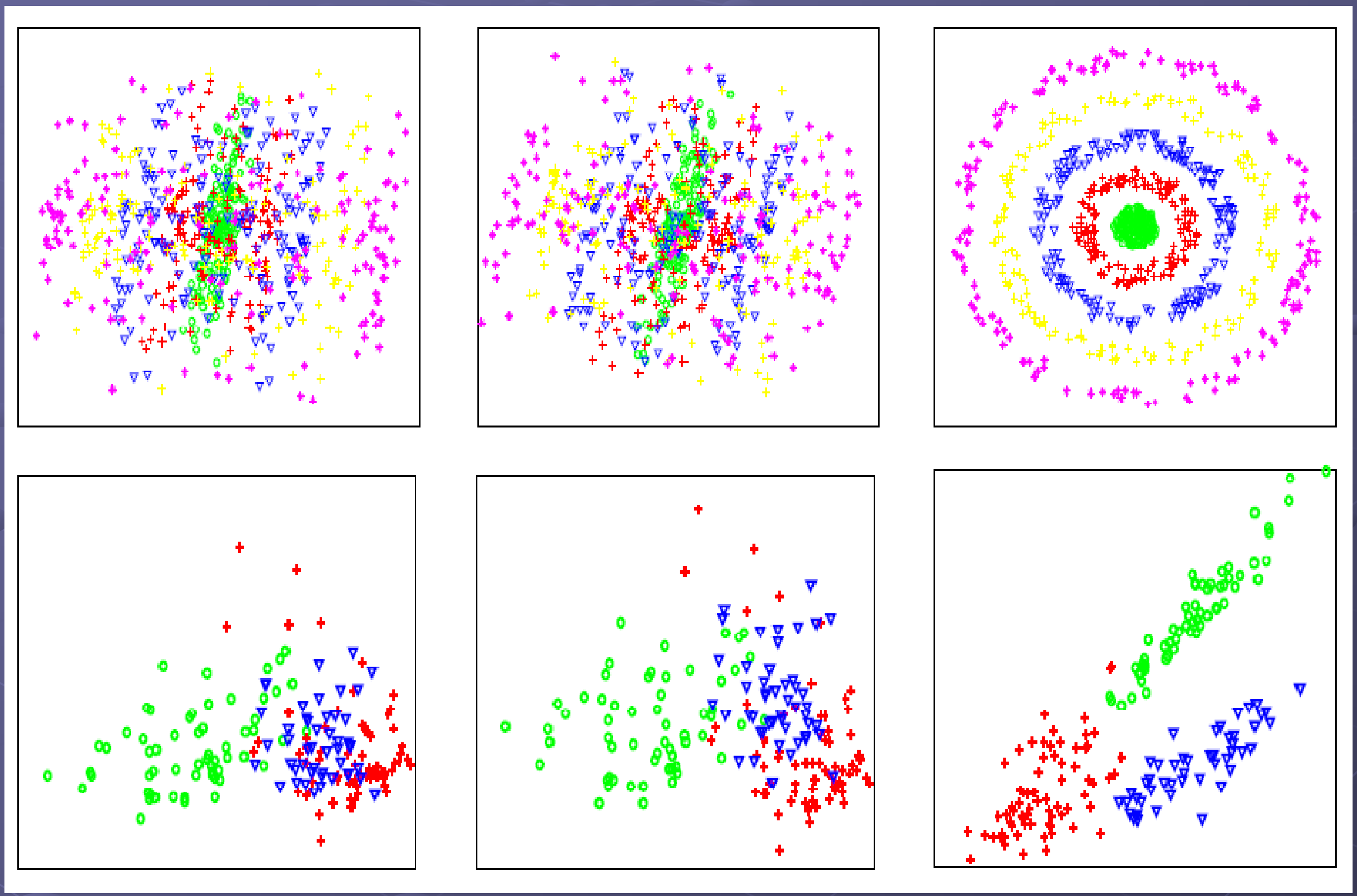
PCA LDA NCA

Results

UCI database

70% training

30% testing



Concentric ring

d = 3

Wine

d = 13

PCA LDA NCA

Extension

- Discrete -> Continuous
- Linear Transformation -> Non-Linear
- Supervised -> Semi-Supervised