Robust Distance Metric Learning in the Presence of Label Noise

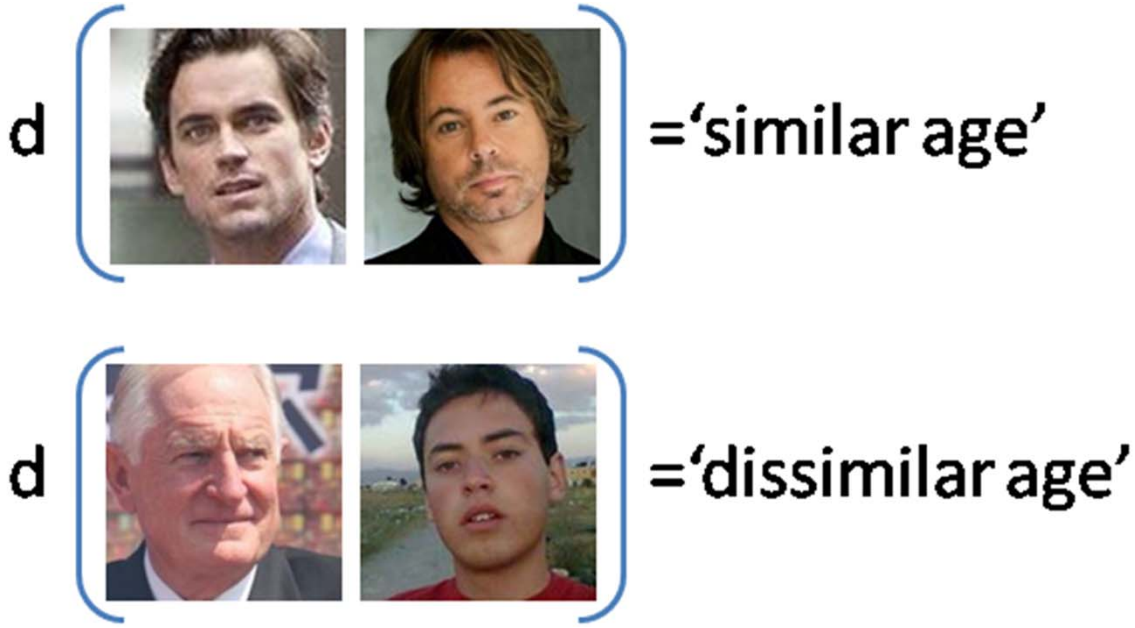
Dong Wang & **Xiaoyang Tan**

Nanjing University of Aeronautics and Astronautics, China

Similarity

Similarity

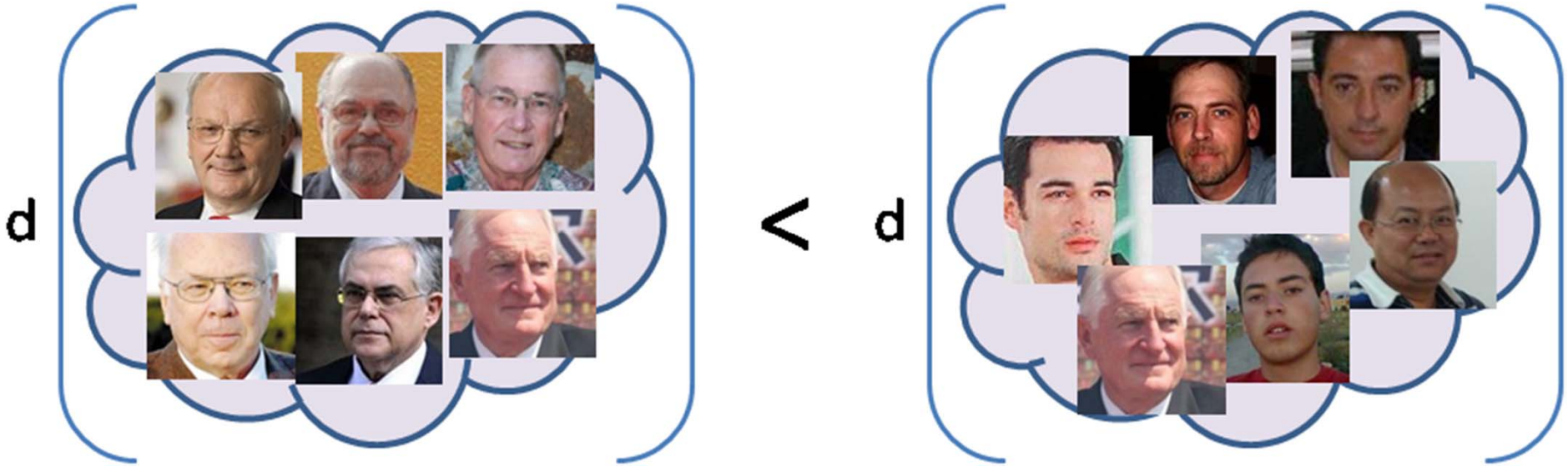
Learning a desired similarity function



Usually the dissimilarity function takes the form of Mahalanobis distance metric

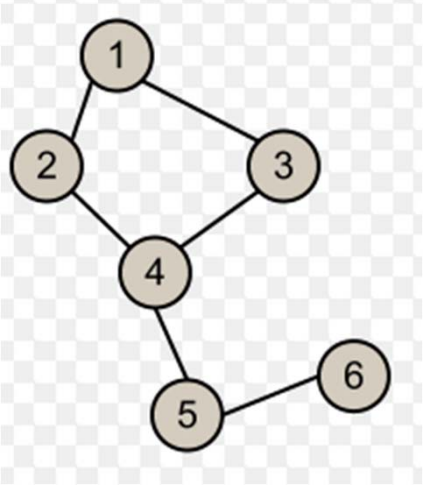
$$d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)$$

Learning a desired similarity function



Neighborhood Components Analysis (NCA)

- Step 1. Encoding the data points with an adjacency graph.



$$p_{ij} = \frac{exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} exp(-\|Ax_i - Ax_k\|^2)}, \quad p_{ii} = 0$$

$$\begin{aligned} d(x_i, x_j) &= (x_i - x_j)^T M (x_i - x_j) \\ &= (x_i - x_j)^T A^T A (x_i - x_j) \\ &= \|Ax_i - Ax_j\|^2 \end{aligned}$$

If two points are similar to each other in a linearly transformed space,

Then the weight of corresponding edge should be relatively large.

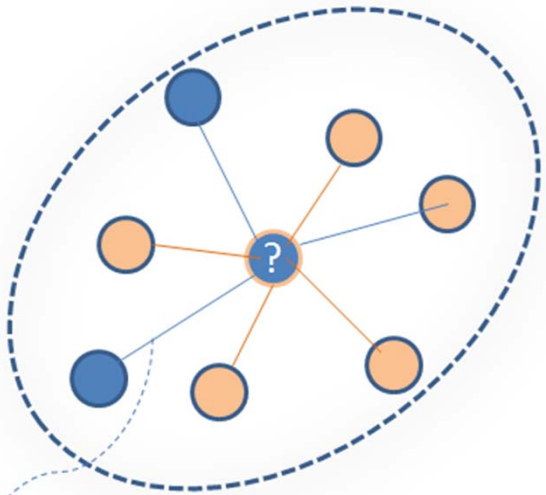
This is just a similarity function between two nodes.

J. Goldberger, G. Hinton, S. Roweis, R. Salakhutdinov. (2005) Neighbourhood Components Analysis. NIPS. 17, 513-520, 2005

Neighborhood Components Analysis (NCA)

- Step 2. calculate the total score of each class (group)

$$p(y_i = k|x_i) = \sum_{j \in C_i^k} p_{ij} = \sum_j p_{ij}\ 1(y_j = k)$$



The contribution of xi to class k:
1)Find its connections to all the nodes labeled as k
2)Sum up the weights of those edges.

a leave one-out estimation – the label of current point is assumed to be missing.

$$p_{ij} = \frac{exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} exp(-\|Ax_i - Ax_k\|^2)}, \quad p_{ii} = 0$$

Neighborhood Components Analysis (NCA)

- Step 2. calculate the total score of each class

$$p(y_i = k|x_i) = \sum_{j \in C_i^k} p_{ij} = \sum_j p_{ij}\ 1(y_j = k)$$

summing up these scores to get the total score of each class

$$= \sum_{i \in C_k} P(y_i = k|x_i)$$

The objective of NCA: maximize the average sore of all the classes, according to A.

$$f(A) = \sum_i \sum_k \log(p_i^k) \qquad p_i^k \equiv p(y_i = k|x_i)$$

Neighborhood Components Analysis (NCA)

- Step 3 Search the best M using gradient ascent

$$\frac{\partial f}{\partial A} = 2A\sum_i\Big(\sum_j p_{ij}x_{ij}x_{ij}^T - \sum_j p_{ij}x_{ij}x_{ij}^T \sum_k \frac{1(y_i=k)1(y_j=k)}{p_i^k}\Big)$$

Total scatter matrix

Intra-class Scatter matrix

To make progress,

seek a direction that makes the intra-class become ‘smaller’ (more tighter) .

NCA assumes CLEAN labels...

inaccurate labels may mislead the NCA's learning direction

(through bad intra-class scatter matrix)

However, in practice the supervisory information (labels) is often inaccurate….

1) labels obtained from the web (crowdsourcing or harvested through web search)

2) too many data points to label (human error)

Deal with inaccurate labels

1. Estimate first for each data point the probability of its label being noised and then...

   1) clean those points whose labels are likely to be noisy (overreacting?)

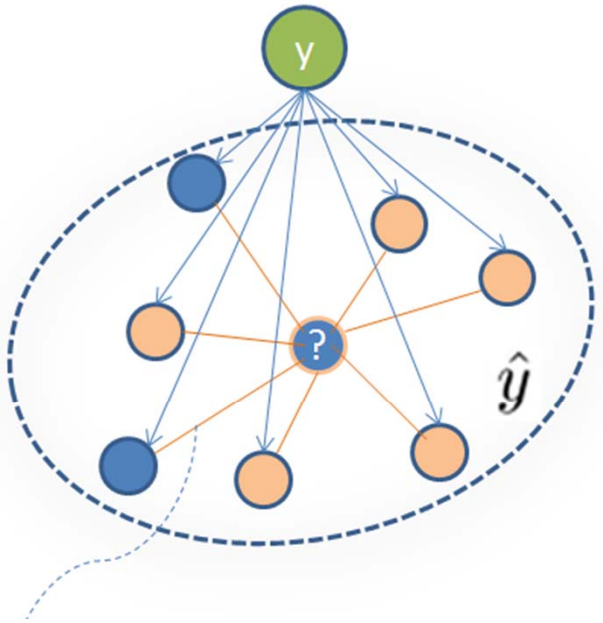
   2) not remove them completely but warn the classifier not to trust them too much

2. set up a firewall - constrain the influence of…

   1) individual point: robust loss functions (e.g., minmax)

   2) the whole dataset: regularization methods (e.g., L2.)

Our Idea



A simple modifications to NCA:

No matter what u see, it could be k, so sum over all the neighbors when computing the contribution of xi to k.

$$p(y_i = k|x_i, \theta) = \sum_j p_{ij} \cdot 1(y_j = k)$$

$$p(y_i = k|x_i, \theta) = \sum_j p_{ij} \cdot p(y_j = k|\hat{y}_j)$$

The goal: estimate both the parameter A and the uncertainty model p(yj| hat yj)

E-step

In the E step, we estimate true label of **each point**, using the current model parameters :

$$Q(y_i = k|x_i, \hat{y}_i = j, \theta) = \frac{p_i^k \gamma_{jk}}{\sum_k p_i^k \gamma_{jk}}$$

Should explain most
Of the observed labels

Prior of observed label j belonging to the k-the class

$$\gamma_{jk} = p(y = k|\hat{y} = j)$$

Likelihood - evidence from the data set

$$p_i^k = p(y_i = k|x_i, A) = \sum_j p_{ij} \cdot p(y_j = k|\hat{y}_j)$$

M-step – update A

$$\frac{\partial L_c}{\partial A} = \sum_{i=1}^{N} \sum_{k} \frac{\alpha_{ik}}{p_i^k} \frac{\partial p_i^k}{\partial A}$$

$$= 2A \sum_{i=1}^{N} (\sum_{l=1}^{N} p_{il} x_{il} x_{il}^T - \sum_{j=1}^{N} p_{ij} x_{ij} x_{ij}^T \sum_{k} \frac{\alpha_{ik} \cdot \gamma_{jk}}{p_i^k})$$

soft update here !

$$\sum_{k} \frac{\alpha_{ik} \cdot \gamma_{jk}}{p_i^k} \propto \sum_{k} p(y = k|\hat{y} = i) \cdot p(y = k|\hat{y} = j)$$

V.S. the standard NCA update:

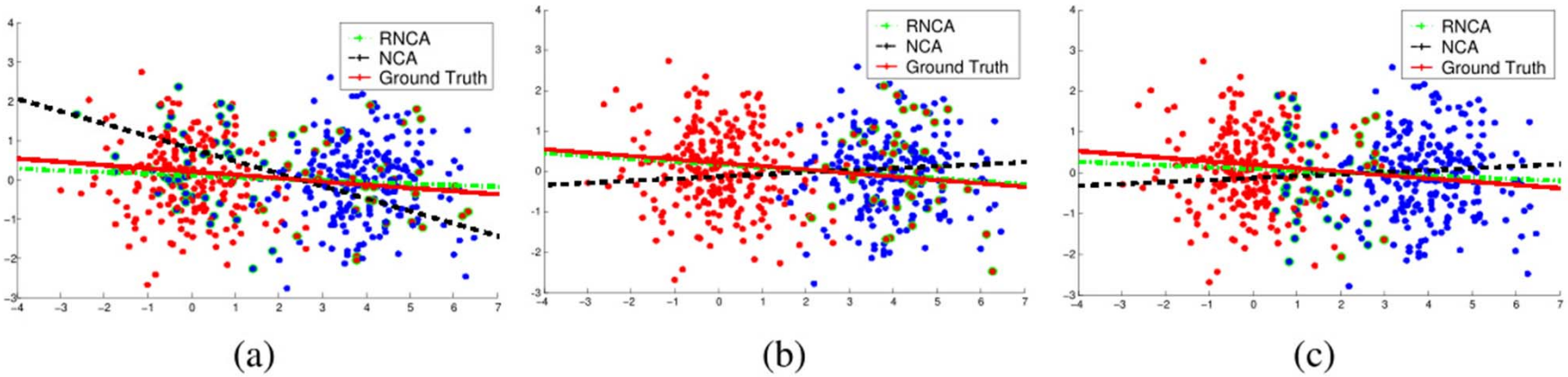
$$\frac{\partial f}{\partial A} = 2A \sum_{i} (\sum_{j} p_{ij} x_{ij} x_{ij}^T - \sum_{j} p_{ij} x_{ij} x_{ij}^T \sum_{k} \frac{1(y_i = k) 1(y_j = k)}{p_i^k})$$

M-step – update gamma

$$\gamma_{ik} = p(y = k|\hat{y} = i) = \frac{\sum_{j=1}^{N} Q(y_j = k|x_j, \hat{y}_j = i) \cdot 1\{\hat{y}_j = i\}}{\sum_{j=1}^{N} 1\{\hat{y}_j = i\}}$$

It is a N x K matrix, describing the prob. that when you observe i, it is actually k.

Learning a 1D transformation



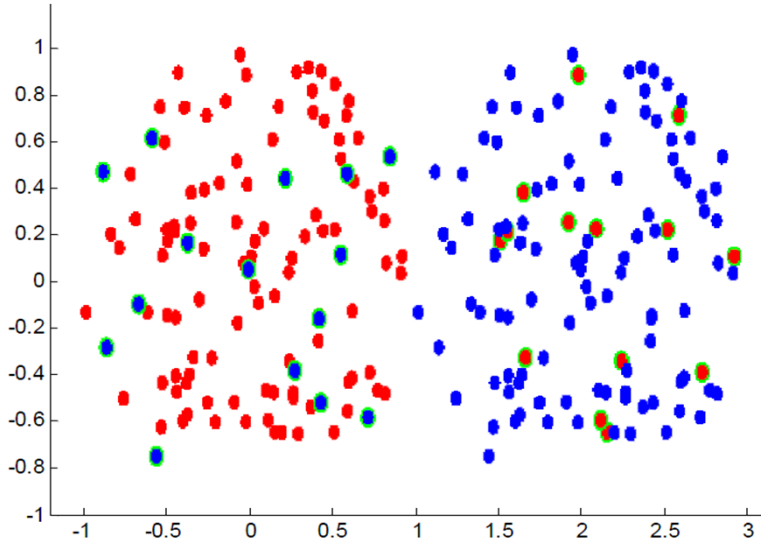
Sample two set of points, add 15% label noise, run metric learning alg.

Three types of label noise are added (from left to right): (a) symmetric random label noise; (b) asymmetric random label noise; (c) label noise occurs on the boundary between two classes.

The projection direction (A) learnt using clean labels (benchmark): Red Line
The one found by NCA (with noisy labels): Black line
The one found by our algorithm (with noisy labels) – Blue line

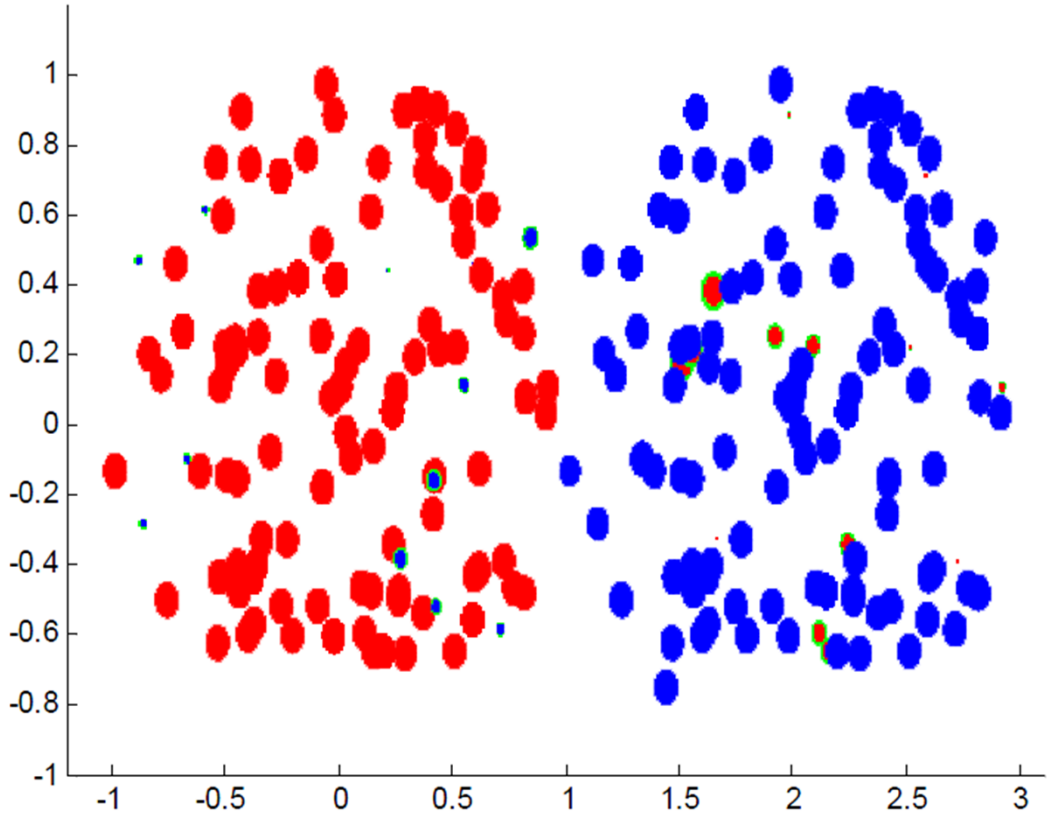
more tolerant to label noise than NCA

The quality of the prediction of the labeling model



$$\alpha_{jk} = p(y_j = k | x_j, \hat{y}_j, A)$$

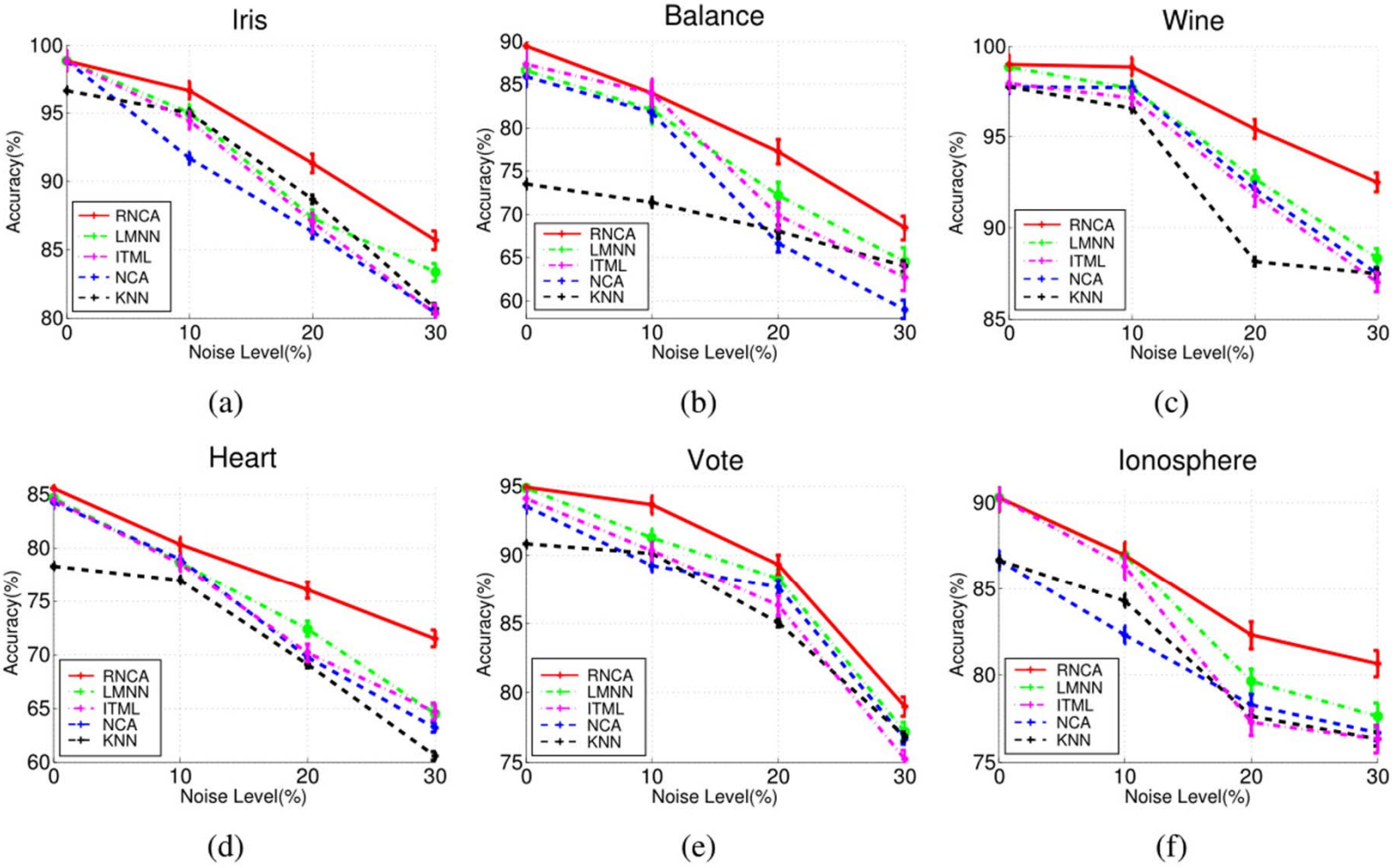
The confidence that the model think what the true label is.



After 4 loops, the size of points with label noise becomes much smaller

=> the model explains well the true labels of most data points.

Experiments with simulated label noise (UCI)



Real world Dataset with label noise

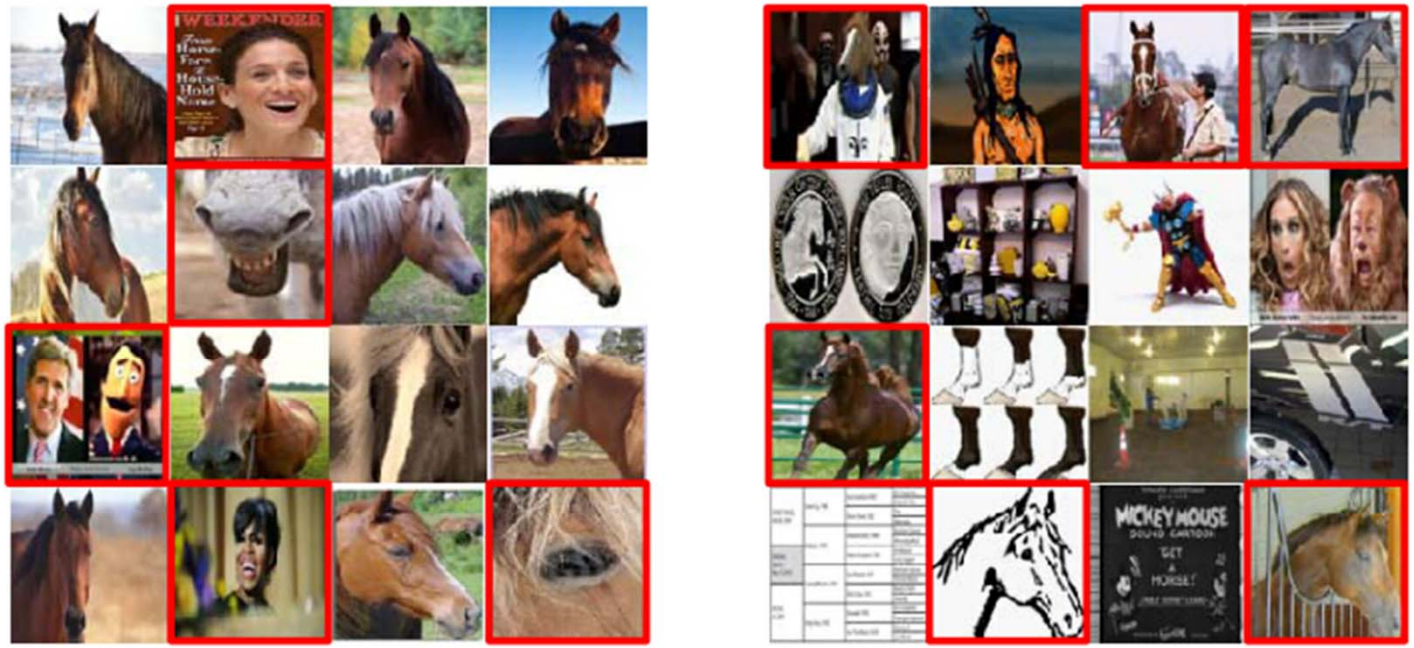
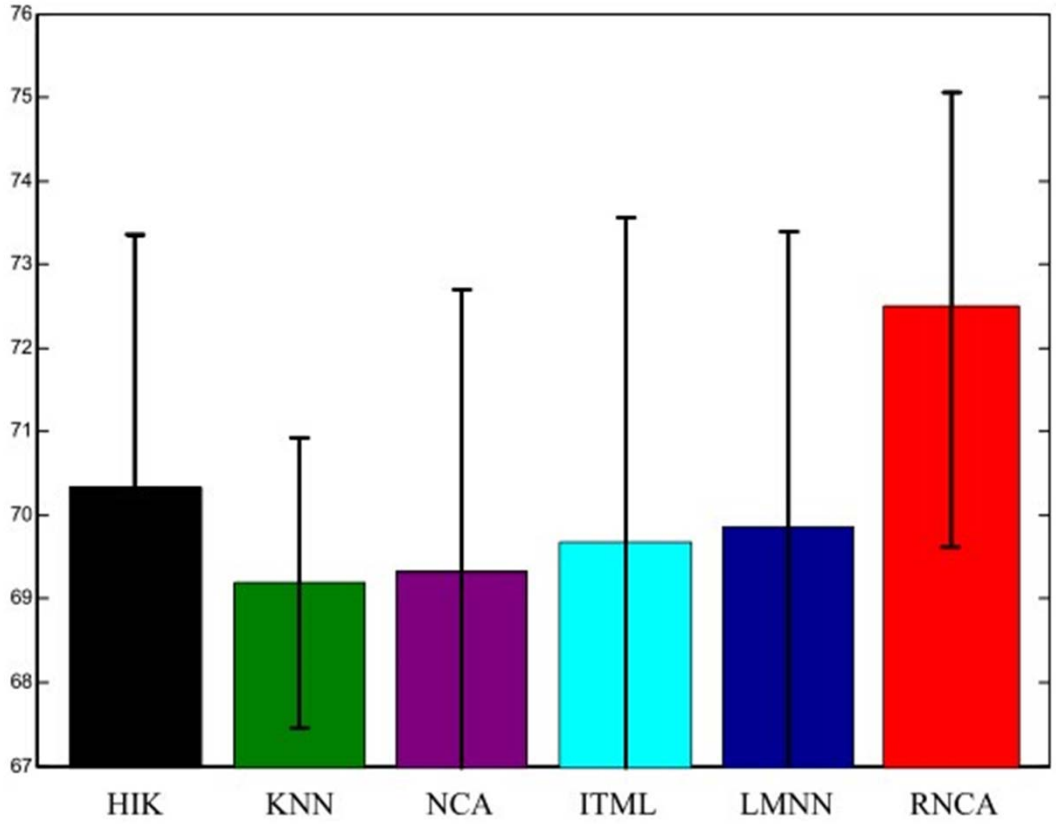


Figure 5: Illustration of typical images in the Horseface dataset harvested from the Web. (a) images in the positive category and (b) images with negative labels, where images with noisy labels are marked with a red square.

Experiments on Horseface Dataset



Comparison of classification performance of various algorithms on the Horseface dataset with label noise.

Conclusion

· We show that the NCA method is sensitive to label noise by inspecting the derivative of likelihood with respect to the transformation matrix A.

· To address this issue, we make a simple modification to NCA
1) a model that faithfully explains most observed labels
2) combine this model with the distribution of observed labels to infer the prior label of a label.

· Experiments on several UCI datasets and a real dataset with unknown noise patterns show that the proposed RNCA is more tolerant to class label noise than NCA.

Thanks