THE ROLE OF DIMENSIONALITY REDUCTION IN CLASSIFICATION

Weiran Wang and **Miguel Á. Carreira-Perpiñán**
EECS, University of California, Merced

1 Abstract

- Dimensionality reduction (DR) is often used as a preprocessing step in classification, but usually in a filter approach. Best performance would be obtained by optimizing the classification error jointly over a DR mapping $\mathbf{F}$ (into latent space $\mathbb{R}^L$) and classifier $\mathbf{g}$ in a wrapper approach, but this is a difficult nonconvex problem:

$$\min_{\mathbf{F},\mathbf{g},\boldsymbol{\xi}} \lambda R(\mathbf{F}) + \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad (1)$$
$$\text{s.t. } \left\{y_n(\mathbf{w}^T\mathbf{F}(\mathbf{x}_n)+b) \ge 1-\xi_n,\ \xi_n \ge 0\right\}_{n=1}^{N}$$

where here we use a linear SVM classifier $\mathbf{g}(\mathbf{F}(\mathbf{x})) = \mathbf{w}^T\mathbf{F}(\mathbf{x}) + b$. (With K classes, we use the one-vs-all scheme and train K binary linear SVMs, one for each class.)
- Using the **method of auxiliary coordinates**, we give a simple, efficient algorithm to train a combination of nonlinear DR and a classifier, and apply it to a RBF mapping with a linear SVM.
- The resulting nonlinear low-dimensional classifier achieves classification errors competitive with the state-of-the-art but is **fast at training and testing**, and allows the user to trade off runtime for classification accuracy easily.
- When trained jointly, the DR mapping takes an extreme role in eliminating variation: it tends to **collapse classes in latent space**, erasing all manifold structure, and lay out class centroids so they are linearly separable with maximum margin.

2 Optimization: method of auxiliary coordinates

Problem (1) can be significantly simplified with the method of auxiliary variables [1]. This breaks the nested functional dependence $\mathbf{g}(\mathbf{F}(\cdot))$ into simpler shallow mappings $\mathbf{g}(\mathbf{z})$ and $\mathbf{F}(\cdot)$, by introducing an auxiliary vector $\mathbf{z}_n \in \mathbb{R}^L$ per input pattern and defining the equivalent problem

$$\min_{\mathbf{F},\mathbf{g},\boldsymbol{\xi},\mathbf{Z}} \lambda R(\mathbf{F}) + \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad (2)$$
$$\text{s.t. } \left\{y_n(\mathbf{w}^T\mathbf{z}_n+b) \ge 1-\xi_n,\ \xi_n \ge 0,\ \mathbf{z}_n = \mathbf{F}(\mathbf{x}_n)\right\}_{n=1}^{N}.$$

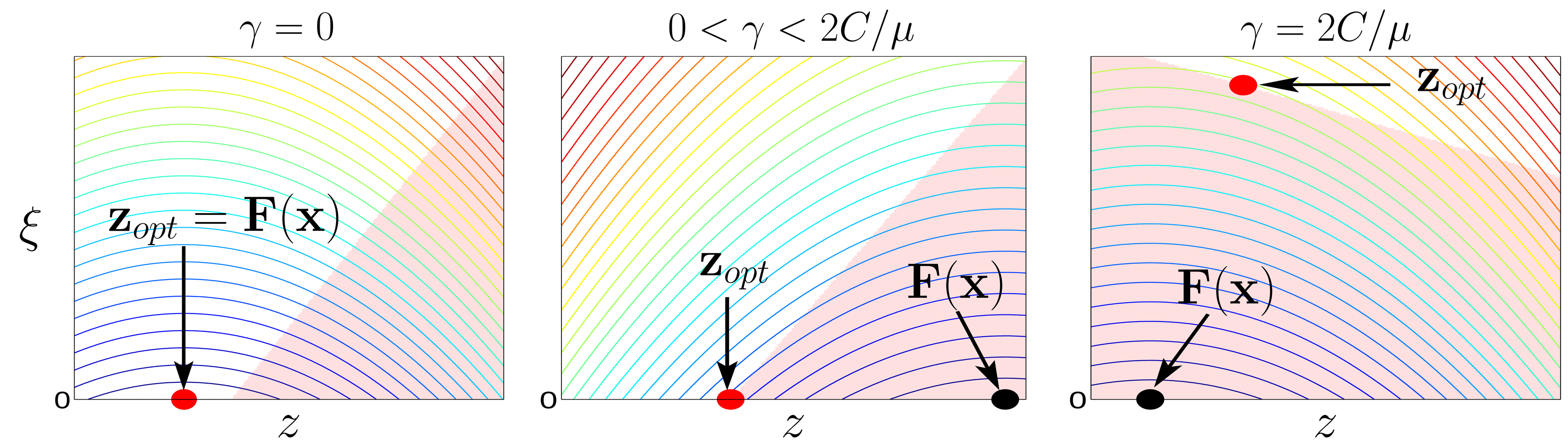
We solve (2) with the **quadratic-penalty method**. We optimize the following problem for fixed penalty parameter $\mu > 0$ and drive $\mu \to \infty$:

$$\min_{\mathbf{F},\mathbf{g},\boldsymbol{\xi},\mathbf{Z}} \lambda R(\mathbf{F}) + \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n + \frac{\mu}{2}\sum_{n=1}^{N}\|\mathbf{z}_n - \mathbf{F}(\mathbf{x}_n)\|^2 \quad (3)$$
$$\text{s.t. } \left\{y_n(\mathbf{w}^T\mathbf{z}_n+b) \ge 1-\xi_n,\ \xi_n \ge 0\right\}_{n=1}^{N}.$$

Alternating optimization for (3): $(\mathbf{F},\mathbf{g})$ step is a usual regression and linear SVM classification done independently from each other (reusing existing algorithms); optimizing over $\mathbf{Z}$ decouples on each n and solves

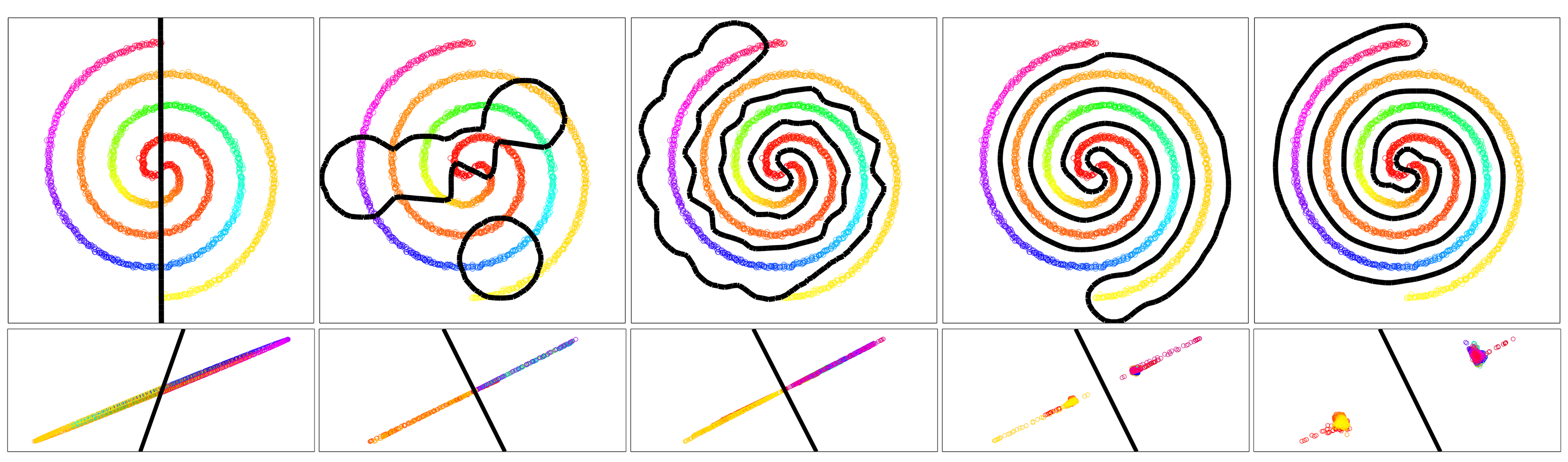
$$\min_{\mathbf{z},\xi} \|\mathbf{z}-\mathbf{F}(\mathbf{x})\|^2 + 2C/\mu\xi \quad \text{s.t. } y(\mathbf{w}^T\mathbf{z}+b) \ge 1-\xi, \quad \xi \ge 0,$$

a convex quadratic program with solution $\mathbf{z}_{opt} = \mathbf{F}(\mathbf{x}) + \gamma y \mathbf{w}$.

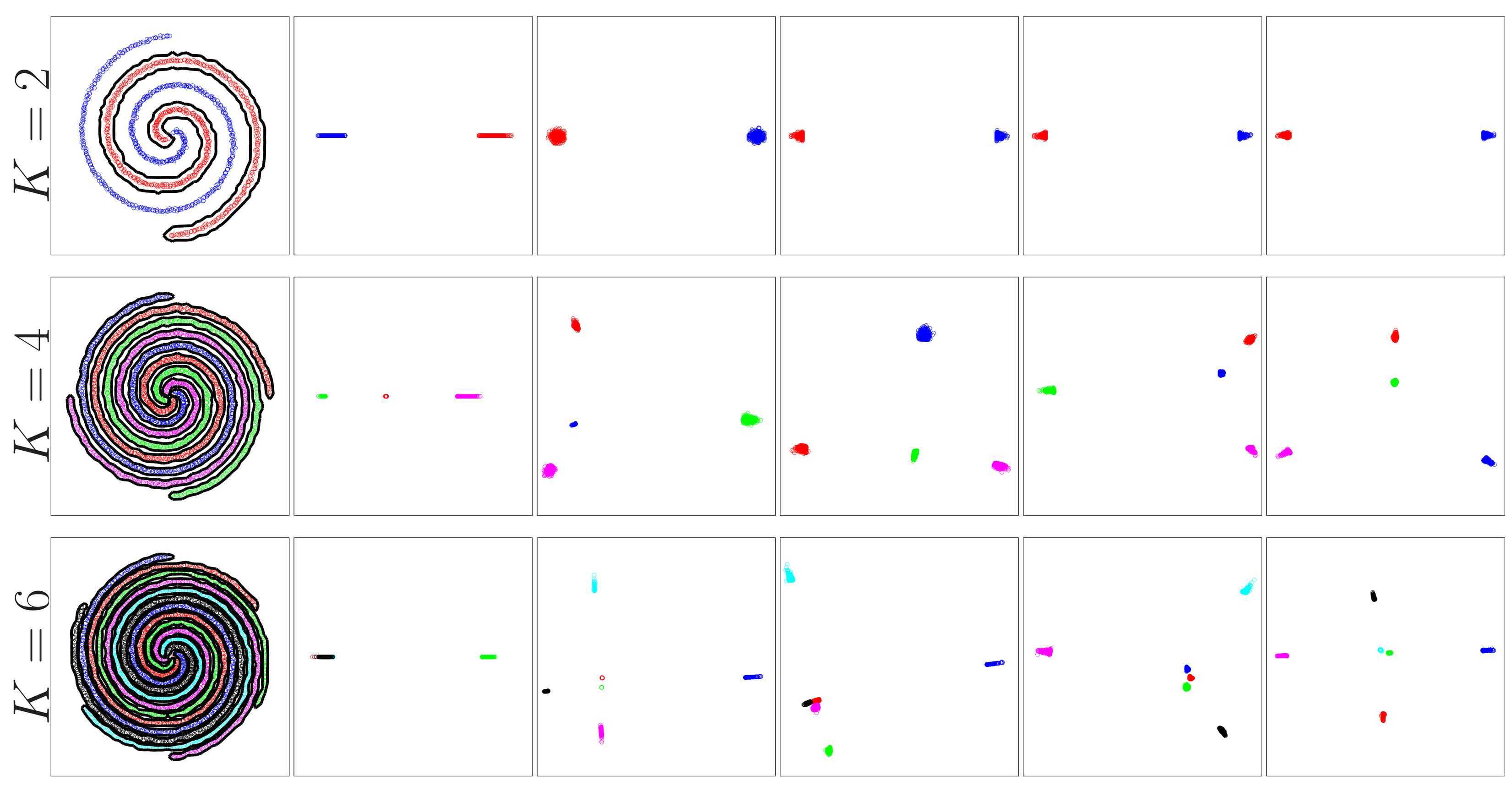


3 Role of dimension reduction in classification

- Formulation (1) does not explicitly seek to collapse classes, but this behavior emerges anyway from the assumption of low-dimensional representation, if trained jointly with the classifier.



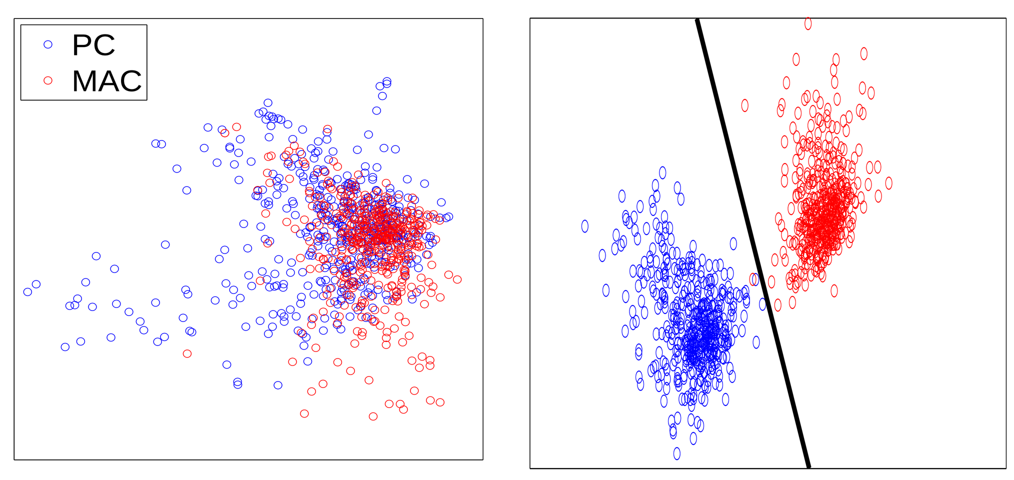
- For K-class problems, the classification performance improves drastically as the latent dimensionality L increases in the beginning, and then stabilizes after some critical L.
- Typically with $L = K-1$ dimensions, the classes form point-like clusters that approximately lie on the vertices of a regular simplex.



[1] Miguel Á. Carreira-Perpiñán and Weiran Wang. Distributed optimization of deeply nested systems. AISTATS 2014.

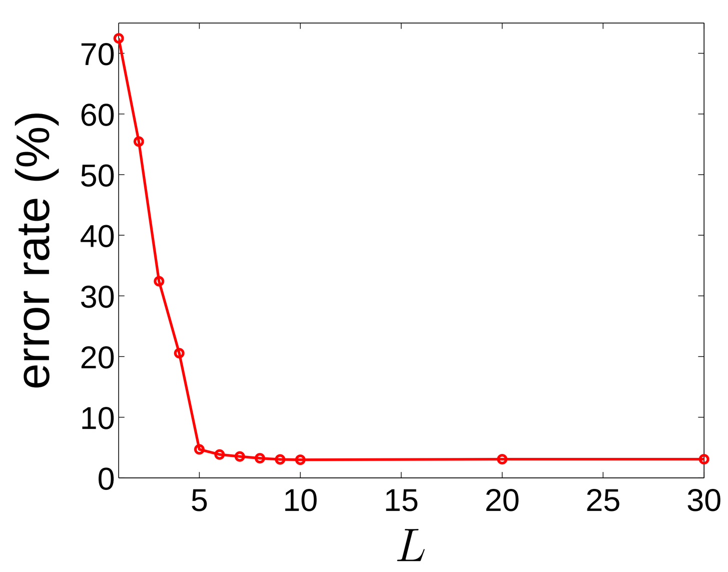
4 Experimental results

Methods	Error (%)
NN	19.16 (0.74)
Linear SVM	13.5 (0.72)
PCA ($L=2$)	42.10 (1.22)
LDA ($L=1$)	14.21 (1.63)
Ours ($L=1$)	**13.12 (0.67)**
Ours ($L=2$)	**12.94 (0.82)**
Ours ($L=20$)	**12.76 (0.81)**

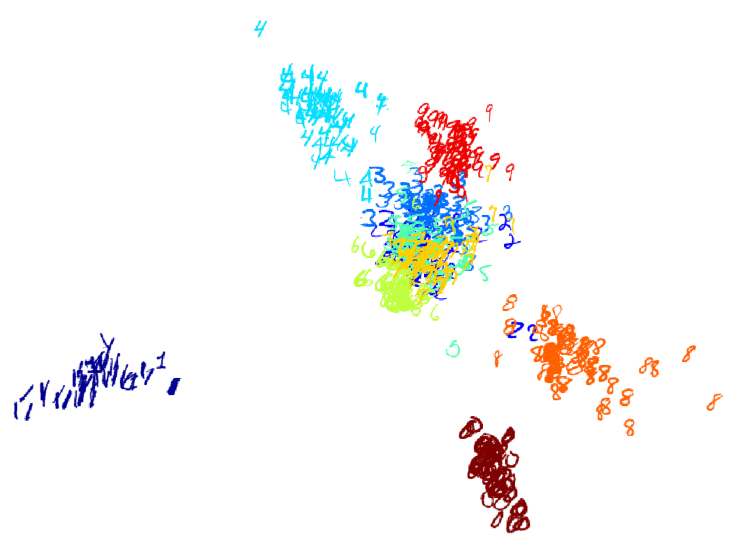


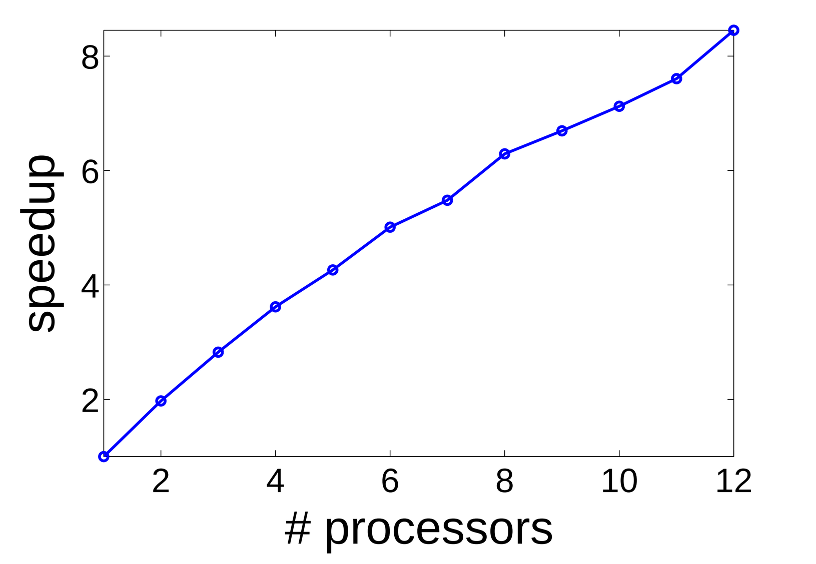
Binary classification results on the PC/MAC subset of 20 newsgroups.

Method	Error	# BFs
Nearest Neighbor	5.34	10 000
Linear SVM	9.20	–
Gaussian SVM	2.93	13 827
LDA (9) + Gaussian SVM	10.67	8 740
PCA (10) + Gaussian SVM	7.44	5 894
PCA (40) + Gaussian SVM	2.58	12 549
Ours (10, 18)	**2.99**	**2 500**
PCA (40) **+ Ours** (10, 17)	**2.60**	**2 500**



Test error rates (%) and number of basis functions used on MNIST.





Embedding of our algorithm on MNIST and speedups obtained with the Matlab Parallel Processing Toolbox.