Tokyo Research Laboratory

Large Margin Component Analysis

by Lorenzo Torresani, Kuang-chih Lee

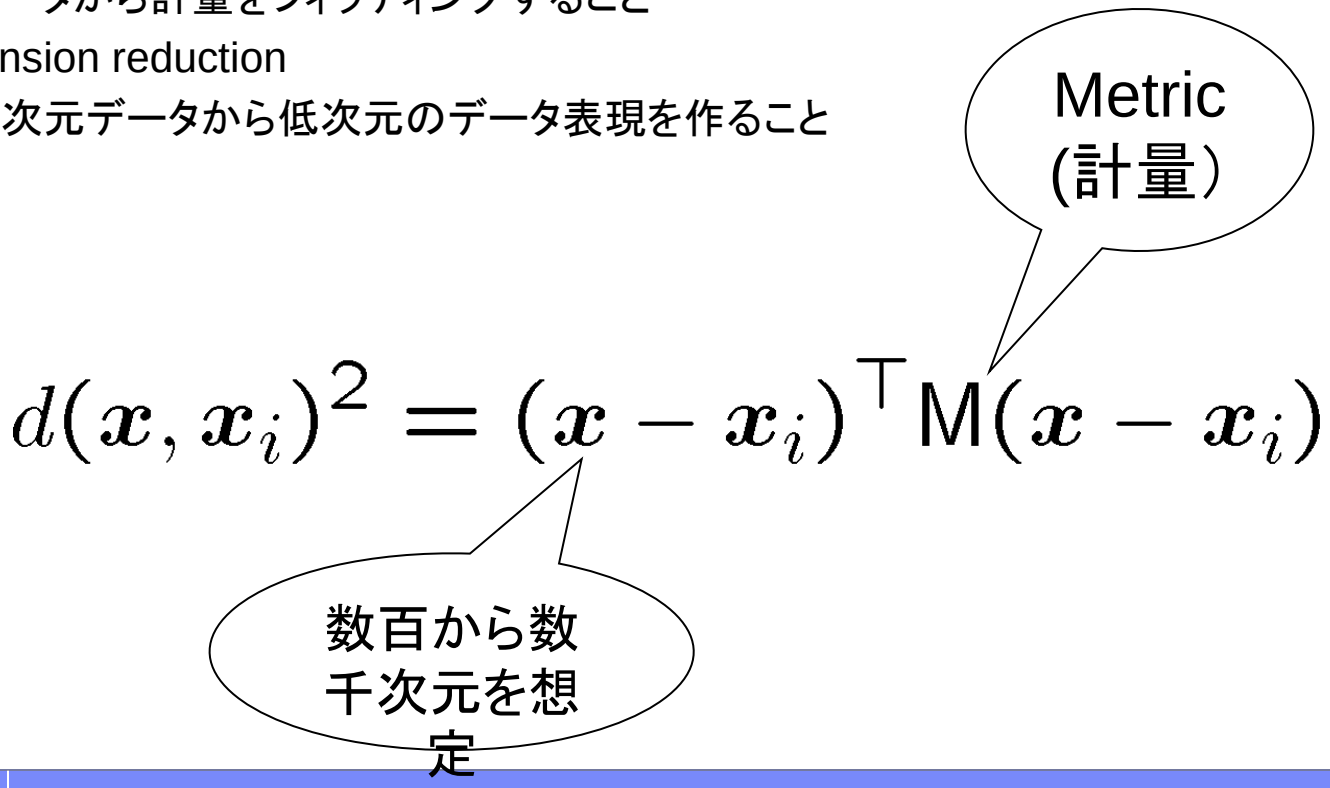
IBM東京基礎研究所
井手剛

目次

- 論文の概要
- なぜこの論文を面白そうだと思ったか
- 論文の紹介
  - はじめに
  - 最大マージンk近傍分類のための次元削減
  - 最大マージン分類のための非線形特徴抽出
  - 実験結果
  - 多手法との関連
- 感想

論文の概要: この論文は、k-NN分類器の改良に関する論文。改良のしどころは、「距離」の計算式。

- キーワード
  - k-NN
    - k最近傍分類法
  - Metric learning
    - データから計量をフィッティングすること
  - Dimension reduction
    - 高次元データから低次元のデータ表現を作ること



なぜこの論文を面白そうだと思ったか

- Large Margin Component Analysis というタイトルが意味深でいい感じだった
- 次元削減とメトリック学習を一緒にやる、というのが気に入った
- 実験結果のグラフをチラと見るとなかなかの性能向上が得られていた

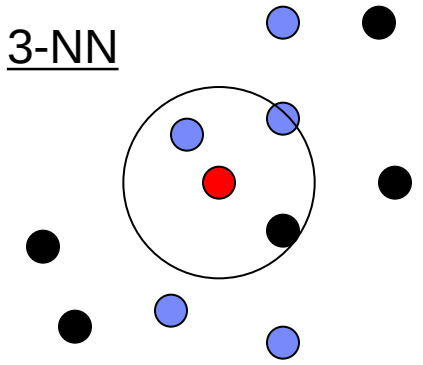
Introduction
はじめに

おさらい
k-Nearest Neighbor とは？

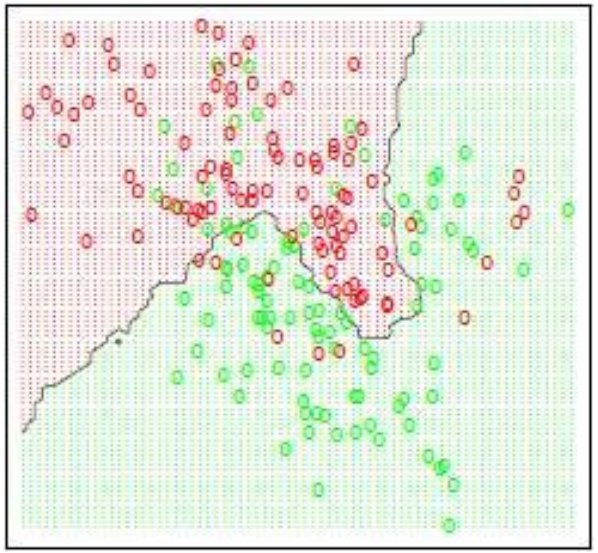
- **k個の近傍点の平均**

$$Y(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

  - $N_k(x)$ はxのk個の近傍
  - 「近傍」を決めるときに、距離の測り方を与える必要



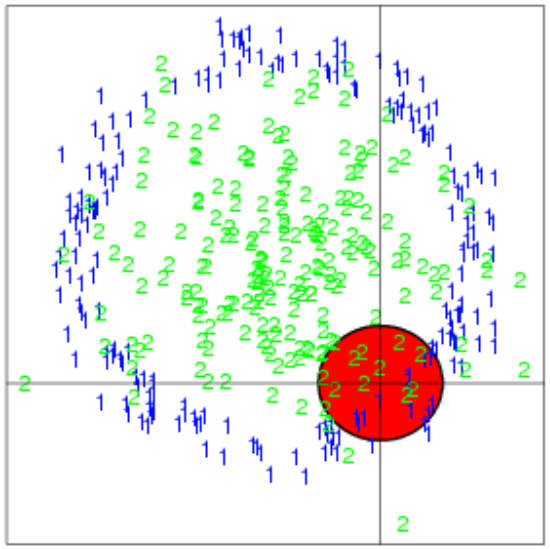
- **単純だけど意外によい**
  - 実装簡単
  - 非線形な識別境界を実現できる
- **でも. . .**
  - 一般には、訓練データをすべて覚えていないといけない
  - 距離の測り方に任意性

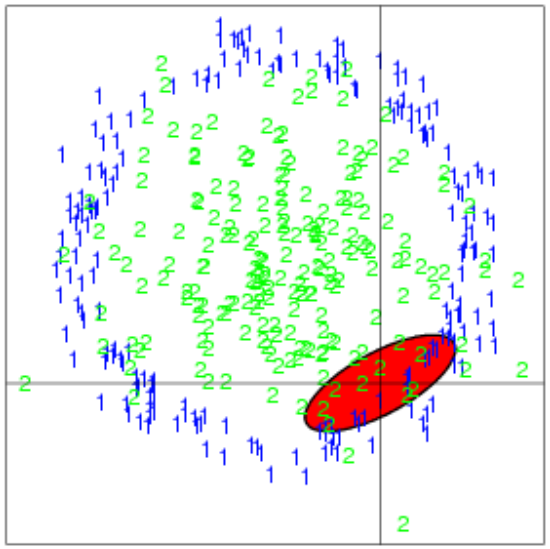


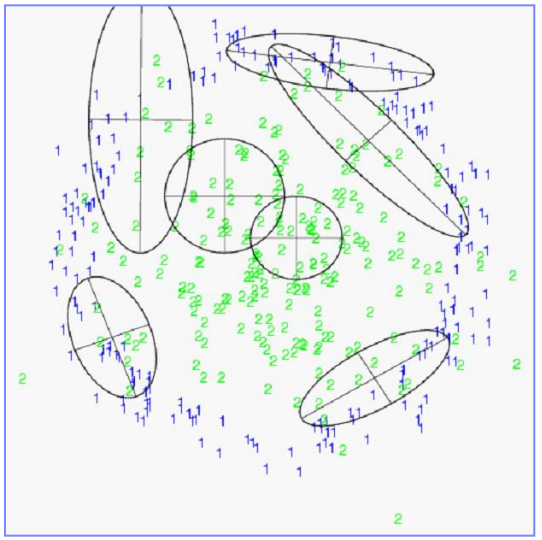
k-NNで距離尺度をいじるとなぜいいのか: DANNアルゴリズム (Hastie-Tibshirani, 1996)

- **“Discriminant adaptive nearest neighbor classification”**
  - メトリック学習のさきがけ（と思われているようだ）
  - 識別境界に沿って空間を圧縮するように距離を測る
  - 行列計算によって、計量Mを訓練データから学習する方法を提案

$$d(\boldsymbol{x}, \boldsymbol{x}_i)^2 = (\boldsymbol{x} - \boldsymbol{x}_i)^\top \mathrm{M} (\boldsymbol{x} - \boldsymbol{x}_i)$$

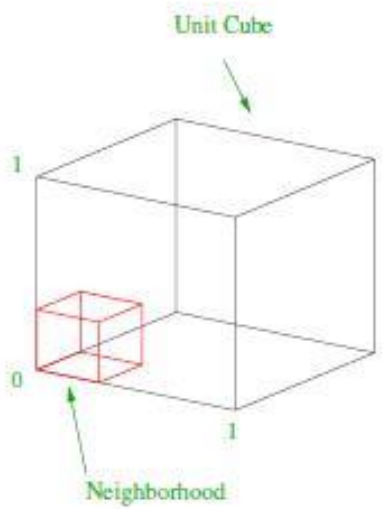


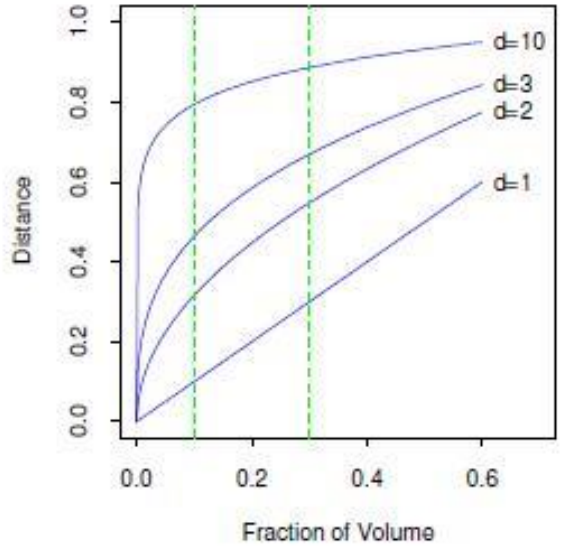




高次元のデータはいろいろ苦しい：記憶容量的に。次元の呪い的に。

- **何が難しいか**
  - 訓練データを全部ためておくのはスペース的に厳しい
  - 計算量も馬鹿にならない
  - 次元に呪われる：近傍の概念がほとんど意味を失ってしまう
    - 単位サイコロの体積の10%を占める辺の長さ
      - 0.46 (3次元)
      - 0.79（10次元）





See “Elements”

Weinberger et al. との差分：（1）計量の学習と次元削減を統合する。（2）計量の学習におけるカーネル化の処方箋を与える

- **この論文はWeinberger et al. [NIPS 2005]の改良である**
  - Objective（後述）はほぼ同一

- **しかし二つの点で本質的な差分がある**
  - 1．次元削減機能を計量の学習方法に取り込んだ
    - Weinberger et al.では前処理としてPCAを行っていた
  - 2．計量の学習のカーネル化をしてみせた
    - Weinberger et al.ではfuture workに挙げられていた

$$d(\boldsymbol{x}, \boldsymbol{x}_i)^2 = (\boldsymbol{x} - \boldsymbol{x}_i)^\top \mathrm{M} (\boldsymbol{x} - \boldsymbol{x}_i)$$

Linear Dimensionality Reduction for Large Margin kNN Classification
最大マージンk近傍分類のための次元削減

WeinbergerのLMNN (Large-Margin Nearest Neighbor, 1/3)
目的関数の第1項は同一ラベルのサンプルの広がりの度合いを表す

- **Weinberger et al.[NIPS 2005]**
- **訓練データ { (x_i, y_i) }**
- **正方・フルランクのD次元行列による1次変換** $x'_i = \mathrm{L}x_i$
- **異なるラベルがうまく分離してくれるようにLを学習したい**

- **Objectiveは二つの項からなる**

$$\epsilon(\mathrm{L}) = \epsilon_1(\mathrm{L}) + \epsilon_2(\mathrm{L})$$

- **第1項は同じラベルのサンプルの広がりの度合い**

$$\epsilon_1(\mathrm{L}) = \sum_{i,j} \eta_{i,j} {d'_{ij}}^2$$

$${d'_{ij}}^2 \equiv ||\mathrm{L}x_i - \mathrm{L}x_j||^2$$

- xi と同一ラベルでk近傍の時に1
- 同一ラベルを持たないと0
- 同じラベルでも、k近傍に入っていないと0

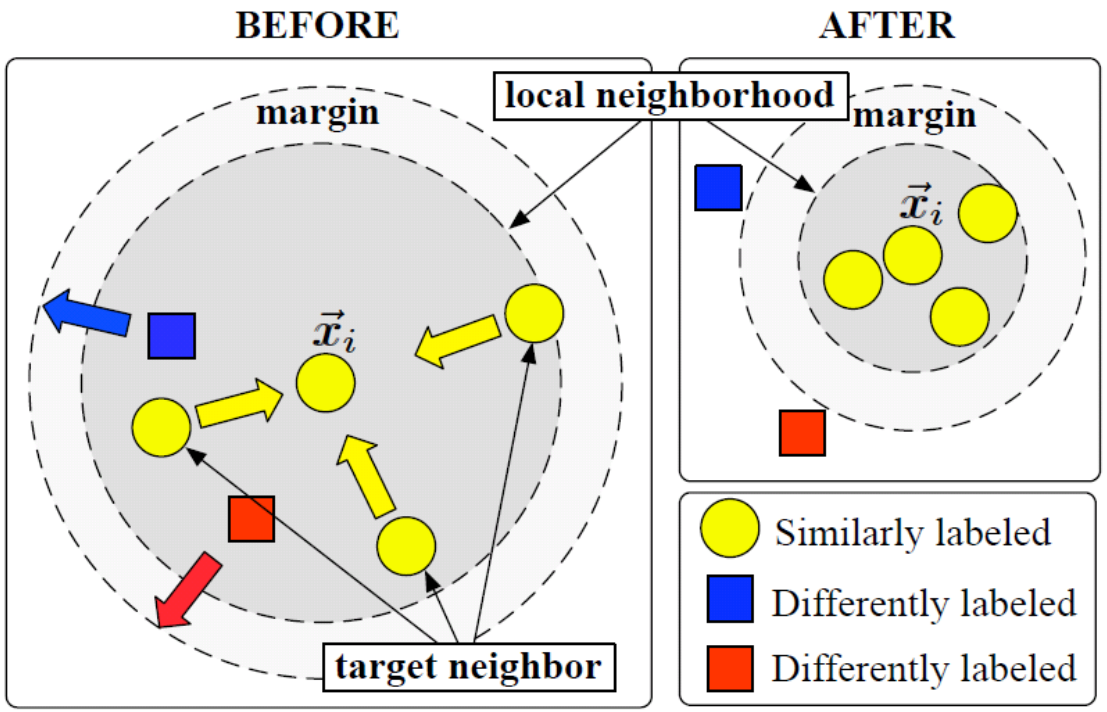
WeinbergerのLMNN (Large-Margin Nearest Neighbor, 2/3)
目的関数の第2項がマージン最大化を目指す損失項

- **Objectiveの第2項は、近寄りすぎの異ラベルサンプルを罰する**

$$\epsilon_2(\mathsf{L}) = c \sum_{i,j,l} \eta_{i,j}(1 - y_{il}) h(d'_{ij}{}^2 - d'_{il}{}^2 + 1)$$

hinge loss

- **絵で理解しよう**



WeinbergerのLMNN (Large-Margin Nearest Neighbor, 3/3) M≡L^TL についてSDPとして解く

- **Hinge lossにスラック変数を使って書き換える**
  - M≡L^TLは常に正定だから半正定値計画問題となる

$$\textbf{Minimize } \sum_{ij} \eta_{ij} (\vec{x}_i - \vec{x}_j)^\top \mathbf{M} (\vec{x}_i - \vec{x}_j) + c \sum_{ij} \eta_{ij} (1 - y_{il}) \xi_{ijl} \textbf{ subject to:}$$

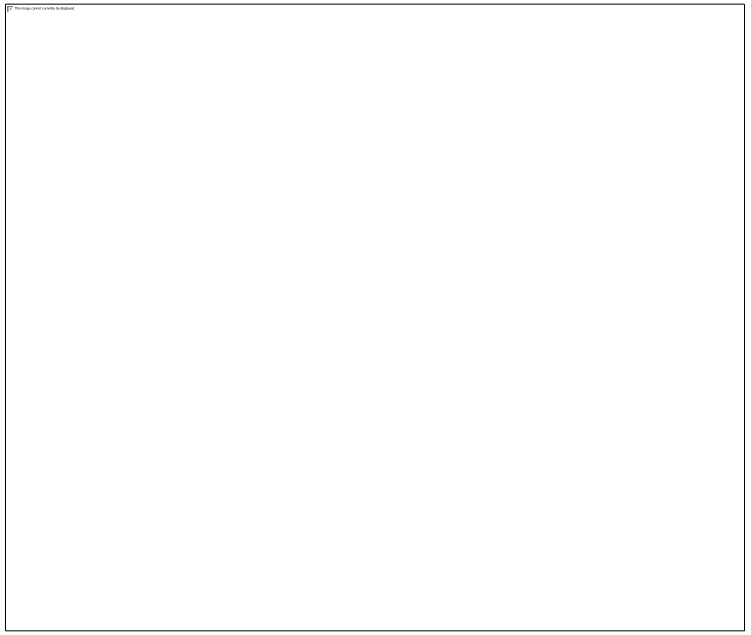
$$\textbf{(1)} \ (\vec{x}_i - \vec{x}_l)^\top \mathbf{M} (\vec{x}_i - \vec{x}_l) - (\vec{x}_i - \vec{x}_j)^\top \mathbf{M} (\vec{x}_i - \vec{x}_j) \ \geq \ 1 - \xi_{ijl}$$

$$\textbf{(2)} \ \xi_{ijl} \geq 0$$

$$\textbf{(3)} \ \mathbf{M} \succeq 0.$$

  - 汎用のSDPソルバは遅すぎるので、勾配法を使ったらしい

- **PCAで次元削減してからLMNNをやると、k-NNよりはよい結果を出す**
  - しかし多クラスSVMにはなかなか勝てない



Large Margin Component Analysis ——変換行列Lを長方形行列に限ることで、低ランク化を行う。凸じゃなくなるが、気にしない

d次元 (<100)

$$\boldsymbol{x}_i' = \mathsf{L}\boldsymbol{x}_i$$

D次元 (~1000)

"Although the objective optimized by our method is also **not convex**, we **experimentally demonstrate** that our solution converges consistently to better metrics than those computed via the application of PCA followed by subspace distance learning (see Section 4)."

- **深刻に考えずに勾配法で解く**

$$\begin{aligned}\frac{\partial \epsilon(\mathbf{L})}{\partial \mathbf{L}} = \; & 2\mathbf{L}\sum_{ij}\eta_{ij}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^T + \\ & 2c\mathbf{L}\sum_{ijl}\eta_{ij}(1-y_{il})\left[(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^T-(\mathbf{x}_i-\mathbf{x}_l)(\mathbf{x}_i-\mathbf{x}_l)^T\right] \\ & h'(||\mathbf{L}(\mathbf{x}_i-\mathbf{x}_j)||^2-||\mathbf{L}(\mathbf{x}_i-\mathbf{x}_l)||^2+1)\end{aligned}$$

Nonlinear Feature Extraction for Large Margin kNN Classification
最大マージン分類のための非線形特徴抽出

この計量学習の勾配法をカーネル化したい（誰もやった人はいないので）。素朴に x →φとするだけでは内積だけでは書けない

- 素朴に x → φとした時の勾配の式

The gradient in feature space can now be written as:

$$\frac{\partial \epsilon(\mathbf{L})}{\partial \mathbf{L}} = 2\sum_{ij} \eta_{ij}\mathbf{L}(\phi_i - \phi_j)(\phi_i - \phi_j)^T + 2c\sum_{ijl} \eta_{ij}(1 - y_{il})h'(s_{ijl})\mathbf{L}\left[(\phi_i - \phi_j)(\phi_i - \phi_j)^T - (\phi_i - \phi_l)(\phi_i - \phi_l)^T\right]$$

Lについては何もしなくてよいのか？？

カーネルPCAでの変換行列をヒントに、L=ΩΦ の形を仮定するのがキモ

- カーネルPCAでの主成分は、マップされたサンプルの線形結合で仮定される
  - $$\boldsymbol{u}^{(i)} = \sum_{l=1}^{n} \alpha_l^{(i)} \boldsymbol{\phi}(\boldsymbol{x}_l)$$
- したがって、主成分で張られる空間への射影行列は、こんな形となる
  - $$\mathrm{L} = \left[\sum_{l=1}^{n} \alpha_l^{(1)} \boldsymbol{\phi}(\boldsymbol{x}_l), \sum_{l=1}^{n} \alpha_l^{(2)} \boldsymbol{\phi}(\boldsymbol{x}_l), \ldots \right]^\top$$
  - 書き換えると
    $$\mathrm{L} = \Omega\Phi \qquad \Phi^\top = [\boldsymbol{\phi}(\boldsymbol{x}_1), \boldsymbol{\phi}(\boldsymbol{x}_2), \ldots]$$
- だからここでも L=ΩΦの形にLを仮定するのが自然。

勾配法による更新則は無事カーネル化された

- 計算に難しいところは全然ない

$$\mathbf{L}_{new} = \mathbf{L}_{old} - \lambda \left. \frac{\partial \epsilon(\mathbf{L})}{\partial \mathbf{L}} \right|_{\mathbf{L}=\mathbf{L}_{old}} = \left[\Omega_{old} - \lambda \Gamma_{old}\right] \Phi = \Omega_{new} \Phi$$

Experimental results
実験結果

実験その1: 高次元データでの実験
実験条件

- **LMCAは勾配法で解くから、初期値が必要**
  - PCAで初期値を与える
- **カーネル版のLMCAはガウシアンカーネルを使う**
  - KLMCAと表記
- **k（k-NNのkのこと）、c（ロス項の係数）、カーネルパラメターはCVで決める**

- **データ**
  - Isolet
    - 6238訓練データ
    - D=617次元
    - 26クラス
  - AT&T Faces
    - 400サンプル
    - D=1178次元
    - 40クラス
  - StarPlus fMRI

実験その1: 高次元データでの実験
WeinbergerらのLMNNを系統的に上回っている。

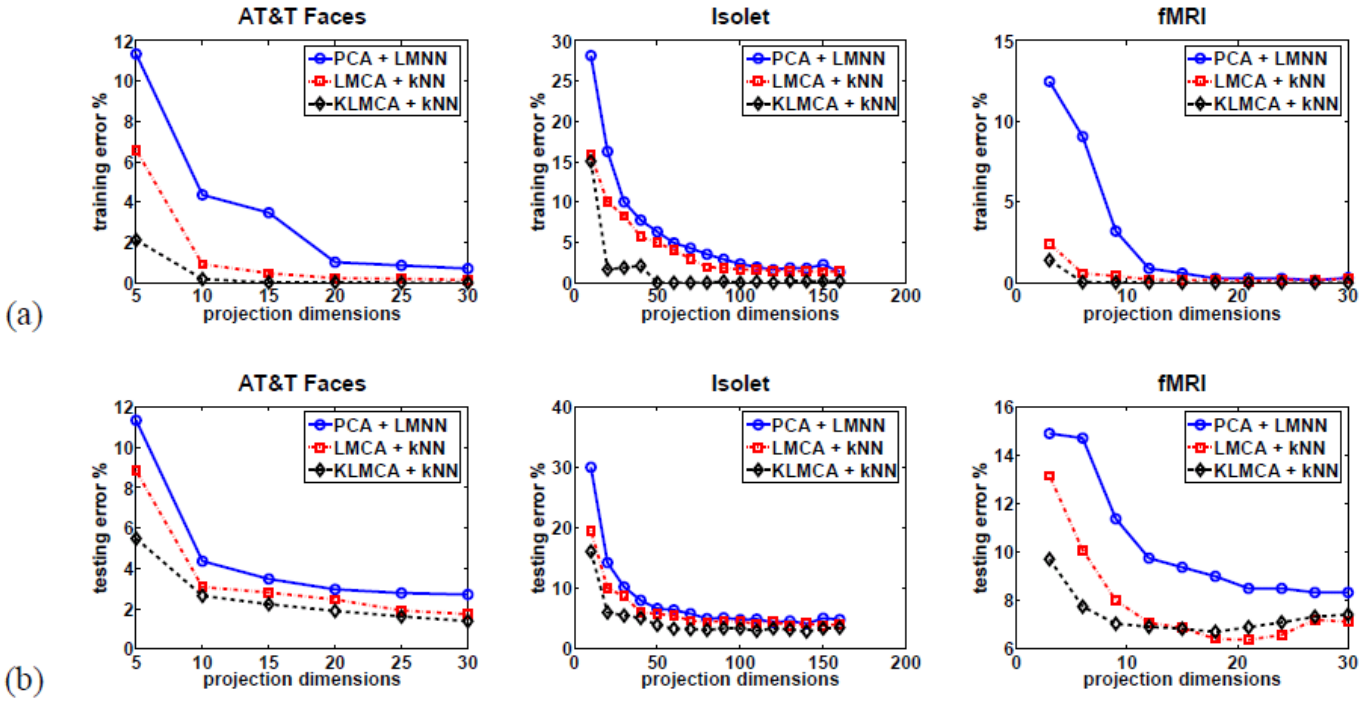


Figure 1: Classification error rates on the high-dimensional datasets Isolet, AT&T Faces and StarPlus fMRI for different projection dimensions. (a) Training error. (b) Testing error.

- **計算時間（AT&T Faces, d=10, k=3）**
  - LMNN 5 sec
  - LMCA 21 sec (k-LMCA 25 sec)

実験その2: 比較的次元が低いデータでの分類精度 多クラスSVMに一応勝っている

- 次元は4から34
- LMCAでは次元削減はしないことにし(→LMNNと同一になる)、KLMCAとLMNNを比べる
  - つまりカーネル化のご利益を調べる

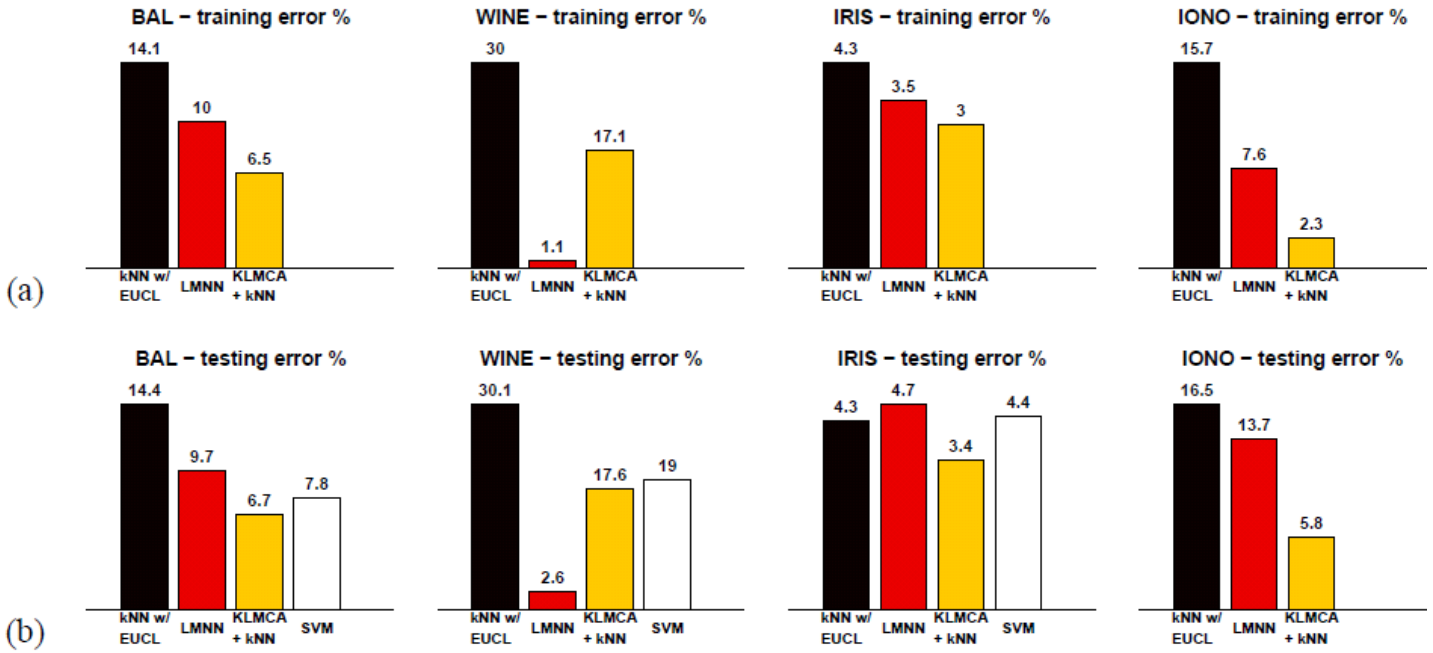


Figure 3: kNN classification accuracy on low-dimensional datasets: Bal, Wine, Iris, and Ionosphere. (a) Training error. (b) Testing error. Algorithms are kNN using Euclidean distance, LMNN [9], kNN in the nonlinear feature space computed by our KLMCA algorithm, and multiclass SVM.

感想

感想

- 個人的には、「空間の曲がり方を学習」、というのは素敵だと思う。
- ただし、メトリック学習についてのおいしいところは大方持って行かれてしまった気がする。
- SDPと聞いてひるんでいた時期がしばらくあったが、勾配法で何とかなるのなら、びびらなくてもいいのかもしれない。
- 凸じゃなくなるけど気にしません、という前向きな姿勢に感銘を受けた。
  - 最適解の理論的保証がない割に、かなりよい性能を出している。