# Fast low-rank metric learning

Dan Oneaţă

## 1 Introduction

- Motivation.
- Thesis structure. Briefly describe how the dissertation is organized.
- Explain the mathematical notation used through-out the thesis.

## 2 Background

### 2.1 Theoretical background

- Metrics. What is a distance metric? Properties.
- Mahalanobis-like metric. What are its advantages? Present equivalence between learning a Mahalanobis-like metric and learning a linear projection or doing feature extraction.

### 2.2 Related methods

- Introduce some of the most representative methods for metric learning: Mahalanobis metric for clustering (Xing et al., 2003), relevant component analysis (Shental et al., 2002).
- Present methods inspired by NCA: metric learning by class collapsing (Globerson and Roweis, 2006), non-linear NCA (Salakhutdinov and Hinton, 2007), largest margin nearest neighbour (Weinberger and Saul, 2009).
- Present work done for fast metric learning (Weinberger and Tesauro, 2007; Weinberger and Saul, 2009).
- Refer to some papers where NCA was used for practical applications: *e.g.*, (Keller et al., 2006; Singh-Miller, 2010).

# 3 Neighbourhood component analysis

## 3.1 General presentation

- Describe and interpret NCA equations.
- Advantages: improves accuracy, useful for dimensionality reduction, assumption-free.
- Drawbacks: non-convexity, gradient evaluation is expensive $\mathcal{O}(N^2D^2)$, how to classify a given point (use $k$NN or NCA objective function).

## 3.2 Practical issues

- Objective function is not convex: how can we avoid local optima? Try different initializations (random, PCA, LDA, RCA). Use different optimization methods (gradient descent, conjugate gradients). Try annealing the dimensionality gradually.

## 3.3 NCA as KDE

- Present NCA as a class-conditional kernel density estimation problem.

# 4 Reducing the computational cost

## 4.1 Sub-sampling

- Naïve idea. Easy to implement: use a random subset $\mathcal{D}_n \subseteq \mathcal{D}$ to learn the projection matrix $\mathbf{A}$.
- For classification we can use the entire data set.
- Reduces the computational cost of the gradient to $\mathcal{O}(n^2D^2)$.
- Drawbacks: does not use the whole information available. Also the final result is affected by the thinner distribution of the points that results by randomly sub-sampling points.

## 4.2 Mini-batches

- Again this is motivated by the fact that the gradient complexity is $\mathcal{O}(n^2D^2)$. This time we use the entire data set: we iteratively use different subsets from $\mathcal{D}$.
- Different possibilities:
    1. Randomly selected mini-batches. These need to be quite large to ensure convergence.

2. Batches selected by clustering: farthest point clustering, recursive projection clustering, agglomerative clustering using $k$-d trees.

### 4.3 Stochastic learning

- Instead of updating the projection matrix $\mathbf{A}$ after one sweep of the entire data set, we can update it gradually after seeing fewer points: we need only a general direction in the parameter space.
- This can be used for on-line learning.

### 4.4 Approximate computations

- Motivated by the fact that only a few contributions are significant. Brute pruning (using only points for whose contribution is greater than a certain threshold $p_{ij} > \epsilon$ + using only the first $k$ neighbours for each point) can be useful.
- In a more principled manner, we can use the KDE interpretation and $k$-d trees.
- This can be further accelerated by training in a stochastic fashion as described previously (subsection 4.3).

### 4.5 Exact computations

- In the model, we can replace the squared exponential kernel with a compact support kernel. In this manner we will have exact computations. We could use $k$-d trees to quickly do range searches.
- We can use the simplest polynomial kernel that satisfies differentiability, although other kernels are probably very similar.
- Must take care to start with the points positioned in such a way that each point has at least one neighbour within its range.
- Again we can use the idea from subsection 4.3.

### 4.6 NCA with compact support kernels and background distribution

- An extended case that takes care of the scenario when a point remains unallocated by using a Gaussian background distribution for each class. This can be interpreted in a generative model.

# 5 Evaluation

# 6 Conclusions

## Bibliography


Globerson, A. and Roweis, S. (2006). Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, volume 18, page 451.

Keller, P., Mannor, S., and Precup, D. (2006). Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 449–456. ACM.

Salakhutdinov, R. and Hinton, G. (2007). Learning a nonlinear embedding by preserving class neighbourhood structure. In *AI and Statistics*, volume 1.

Shental, N., Hertz, T., Weinshall, D., and Pavel, M. (2002). Adjustment learning and relevant component analysis. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 776–792. Springer.

Singh-Miller, N. (2010). *Neighborhood Analysis Methods in Acoustic Modeling for Automatic Speech Recognition*. PhD thesis, Massachusetts Institute of Technology.

Weinberger, K. and Saul, L. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244.

Weinberger, K. and Tesauro, G. (2007). Metric learning for kernel regression. In *Eleventh international conference on artificial intelligence and statistics*, pages 608–615.

Xing, E., Ng, A., Jordan, M., and Russell, S. (2003). Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, pages 521–528.


## Appendix

This section will consists of further results (tables, figures, graphs) and mathematical derivations that were not included in the main text.