# Convergence of Multi-Pass Large Margin Nearest Neighbor Metric Learning

Christina Göpfert Benjamin Paassen
Barbara Hammer *

CITEC center of excellence
Bielefeld University - Germany

*(This is a preprint of the publication [7], as provided by the authors.)*

**Abstract**

Large margin nearest neighbor classification (LMNN) is a popular technique to learn a metric that improves the accuracy of a simple k-nearest neighbor classifier via a convex optimization scheme. However, the optimization problem is convex only under the assumption that the nearest neighbors within classes remain constant. In this contribution we show that an iterated LMNN scheme (multi-pass LMNN) is a valid optimization technique for the original LMNN cost function without this assumption. We further provide an empirical evaluation of multi-pass LMNN, demonstrating that multi-pass LMNN can lead to notable improvements in classification accuracy for some datasets and does not necessarily show strong overfitting tendencies as reported before.

## 1 Introduction

Metric learning is concerned with inferring a metric from data that supports further processing of said data. The most common application of metric learning is the support of classification schemes. In simple terms this can be described as a distance that makes data points from the same class look more similar and data points from different classes look more dissimilar. Large margin nearest neighbor classification (LMNN) is one of the most popular techniques in the metric learning zoo [12, 1, 9], which specifically aims to improve the accuracy of a $k$-nearest neighbor classifier. It has been sucessfully applied in pattern recognition tasks such as pedestrian recognition [4], face identification [6] and movement classification [8].

As most other metric learning approaches, LMNN introduces a positive semidefinite matrix $M$ to the standard Euclidean metric and optimizes this matrix according to a cost function that models the $k$-nearest neighbor classification error. This optimization is an instance of *semidefinite programming*,

*Funding by the DFG under grant number HA 2719/6-2 and the CITEC center of excellence (EXC 277) is gratefully acknowledged.

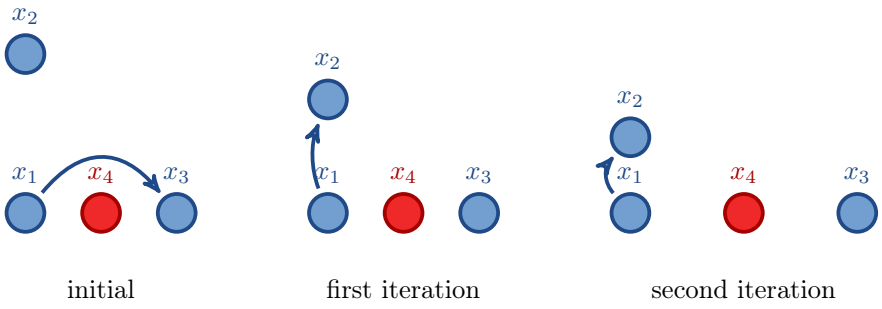


Figure 1: A schematic illustration of a scenario where changes in the target neighborhood make the LMNN optimization easier. Left: The initial configuration where the data point $x_1$ is closest to $x_3$ within the same class. Middle: After a first metric learning step, $x_2$ becomes the target neighbor. $x_1$ would still not be correctly classified, because $x_4$ is closer to $x_1$ than $x_2$. Right: Another metric learning step can now transform the space such that $x_1$ and $x_2$ are close but $x_1$ and $x_4$ are far apart.

which implies that a global optimum can be found [12, 2]. However, this desirable property only holds under the assumption that the closest $k$ neighbors from the same class - the so-called *target neighbors* - remain constant. It is easy to imagine a setting where this assumption is violated. Consider Figure 1 (left and middle), for example. Here, the optimization of the convex problem does not find the global optimum in the LMNN cost function but a local one. The global optimum can only be found if neighborhood changes induced by the metric change are taken into account. This gives reason to suspect that classic LMNN might fail for data sets where changes in the neighborhood are likely to occur. Therefore it seems worthwhile to investigate the theoretical validity of LMNN in more detail.

In this contribution we show that the constant neighborhood assumption leads to an *overestimation* of the LMNN cost function, which implies that an update of the target neighborhood leads to an improvement in the cost function value. After updating the target neighbors, another LMNN run can be applied, resulting in a multi-pass LMNN scheme, converging to a local optimum (Section 5). We also demonstrate that such an iterative scheme does indeed improve the classification accuracy on artificial data (Section 6), and does not show strong overfitting tendencies on real data, that have been reported before [12].

## 2 Related Work

Several properties of large margin nearest neighbor classification (LMNN) have been investigated in the literature. For example, Do and colleagues have shown that LMNN can be regarded as learning a set of local SVM variants in a quadratic space [5]. Further, Ying and Li have reformulated LMNN as an Eigenvalue optimization problem [13]. Finally, several extensions of the original LMNN approach have been proposed, such as varied cost functions that support faster optimization [11], hierarchical LMNN [3], multi-task LMNN [10] and

several more [1, 9]. However, these extensions still assume a constant target neighborhood. To our knowledge, only Weinberger and Saul have attempted to adapt the target neighborhood in a multi-pass LMNN scheme [12]. However, they do not provide theoretical justification for this approach.

# 3 Quadratic Form Distances

Most metric learning schemes - LMNN among them - focus on a so-called *Mahalanobis metric* [9, 1]. More precisely, assume that we have $N$ data points $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^n$. We define $d_M$ as a binary function

$$d_M(x_i, x_j) := \sqrt{(x_i - x_j)^T \cdot M \cdot (x_i - x_j)} \tag{1}$$

Note that $d_M$ is a metric iff $M \in \mathbb{R}^{n \times n}$ is positive semidefinite. If $M$ is the $n$-dimensional identity matrix, this is the standard Euclidean distance. Interestingly, positive-semi-definiteness of $M$ also implies that $M$ can be refactored into a product $M = L^T \cdot L$ for some matrix $L \in \mathbb{R}^{n \times n}$. $L$ can then be interpreted as a linear transformation to a space, where $d_M$ corresponds to the Euclidean metric. The challenge of a metric learning algorithm is to adapt $M$, such that the target task - e.g. classification - becomes simpler.

# 4 Large Margin Nearest Neighbor Classification

The aim of large margin nearest neighbor classification (LMNN) is to ensure good classification accuracy of a $k$-nearest neighbor classifier. A $k$-nearest neighbor classifier assigns the class label of the majority of the $k$ nearest neighbors. Thus, to guarantee correct classification for each point, it has to be ensured that the majority of the $k$ nearest neighbors belong to the correct class. LMNN formalizes this objective in a cost function with two parts: the first ensures that certain data points from the same class are close together, the second ensures that data points from different classes are *not* close together.

More precisely, given a data set $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^n$ with the respective class labels $y_i$, the LMNN cost function $E$ is given as [12]:

$$E(M) := \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_M^k(i)} d_M^2(x_i, x_j) + \sum_{l=1}^{N} (1 - y_i \cdot y_l) \cdot \Big[ d_M^2(x_i, x_j) + \gamma^2 - d_M^2(x_i, x_l) \Big]_+ \tag{2}$$

where $\gamma$ is a positive real number called the *margin*; $[\cdot]_+$ denotes the hinge-loss defined as $[r]_+ := \max\{0, r\}$; and $\mathcal{N}_M^k(i)$ are the indices of the $k$ nearest neighbors (regarding $d_M$) of point $x_i$ that belong to the same class. $\mathcal{N}_M^k(i)$ is also called the *target neighborhood* of $x_i$.

Note that $\mathcal{N}_M^k$ depends on $M$. Therefore, a direct minimization of $E$ by adapting $M$ is infeasible. However, if the target neighborhood is fixed, a semidefinite program results, which can be solved efficiently [12, 2]. We call this the *constant target neighborhood assumption*. It can be formalized as the minimiza-

tion of $\tilde{E}$, where

$$\tilde{E}(M,\mathcal{N}^k) := \sum_{i=1}^{N} \sum_{j\in\mathcal{N}^k(i)} d_M^2(x_i,x_j) + \sum_{l=1}^{N} (1-y_i\cdot y_l)\cdot \Big[d_M^2(x_i,x_j)+\gamma^2-d_M^2(x_i,x_l)\Big]_+ . \tag{3}$$

and the second argument is fixed to some assignment of $k$ target neighbors for each point. Note that $\tilde{E}(M,\mathcal{N}_M^k) = E(M)$.

# 5 Multi-Pass LMNN

We intend to show that an indirect minimization of $E$ is possible using an alternating optimization scheme. We proceed in two steps: First we prove that the classic LMNN solution *overestimates* $E$. Then we provide a convergence proof for our proposed alternating scheme.

**Theorem 1.** *Let $M$ and $M'$ be positive-semidefinite $n \times n$ matrices. Then it holds:*

$$\mathcal{N}_M^k = \mathcal{N}_{M'}^k \Rightarrow \tilde{E}(M',\mathcal{N}_M^k) = \tilde{E}(M',\mathcal{N}_{M'}^k) \tag{4}$$

$$\mathcal{N}_M^k \neq \mathcal{N}_{M'}^k \Rightarrow \tilde{E}(M',\mathcal{N}_M^k) > \tilde{E}(M',\mathcal{N}_{M'}^k) \tag{5}$$

*Proof.* If $\mathcal{N}_M^k = \mathcal{N}_{M'}^k$, then $\tilde{E}(M',\mathcal{N}_M^k) = \tilde{E}(M',\mathcal{N}_{M'}^k) = E(M')$ and the assertion in Equation 4 is clear.

If $\mathcal{N}_M^k(i) \neq \mathcal{N}_{M'}^k(i)$ for some $i \in \{1,\dots,N\}$, then for each $j \in \mathcal{N}_M^k(i) \setminus \mathcal{N}_{M'}^k(i)$, $j' \in \mathcal{N}_{M'}^k(i)$, and $l \in \{1,\dots,N\}$, we have

$$d_{M'}(x_i,x_{j'}) < d_{M'}(x_i,x_j) \tag{6}$$

and

$$\Big[d_{M'}^2(x_i,x_{j'}) + \gamma^2 - d_{M'}^2(x_i,x_l)\Big]_+ \leq \Big[d_{M'}^2(x_i,x_j) + \gamma^2 - d_{M'}^2(x_i,x_l)\Big]_+ \tag{7}$$

Thus, the summand for $i$ of $\tilde{E}(M',\mathcal{N}_M^k)$ is strictly larger than the corresponding summand of $\tilde{E}(M',\mathcal{N}_{M'}^k)$. As every other summand is either equal to or larger than the corresponding one in $\tilde{E}(M',\mathcal{N}_{M'}^k)$, the assertion in Equation 5 follows. □

If the constant target neighborhood assumption is guaranteed to lead to an overestimation of the actual cost function value, a minimization of $\tilde{E}$ under constant neighborhood assumption also decreases $E$. This suggests an alternating optimization scheme as shown in Algorithm 1, which is equivalent to multi-pass LMNN as proposed by Weinberger and Saul [12]. We optimize $M$ w.r.t. $\tilde{E}$, then update the target neighborhoods. If at least one target neighborhood changes, we continue, otherwise the algorithm has converged.

**Theorem 2.** *Algorithm 1 is guaranteed to converge to a local optimum after a finite number of steps.*

**Algorithm 1** An alternating optimization scheme for the LMNN cost function shown in Equation 2.

```
Initialize M ← I^n.
converged ← false
while ¬converged do
    Optimize M w.r.t. Ẽ(M, N_M^k) via classic LMNN techniques.
    converged ← true
    for i ∈ {1, ..., N} do
        Update N_M^k(i).
        if N_M^k(i) has changed then
            converged ← false.
        end if
    end for
end while
return M.
```

*Proof.* Let $(M_t)_t$ be a sequence of matrices produced by a run of Algorithm 1. Then we know that $\tilde{E}(M_{t+1}, \mathcal{N}^k_{M_t}) \leq \tilde{E}(M_t, \mathcal{N}^k_{M_t})$ due to the convex optimization step and $\tilde{E}(M_{t+1}, \mathcal{N}^k_{M_t+1}) \leq \tilde{E}(M_{t+1}, \mathcal{N}^k_{M_t})$ due to Theorem 1. Thus, $E(M_{t+1}) \leq E(M_t)$ for all $t$.

If the algorithm terminates after $T$ steps, then $\mathcal{N}^k_{M_T} = \mathcal{N}^k_{M_{T-1}}$. This implies that $\tilde{E}$ reached a local optimum because no change in the matrix can be made anymore that would decrease the value - otherwise it would have been chosen in the last step. This, in turn, implies a local optimum of $E$. Therefore, the stopping criterion of Algorithm 1 corresponds to a local optimum.

Now, assume that the algorithm does not stop. Since there is only a finite number of target neighborhoods to choose from, there must be $t, t'$ with $t' > t$, such that $\mathcal{N}^k_{M_t} = \mathcal{N}^k_{M_{t'}}$. Since the optimization step of the algorithm finds a *global* optimum w.r.t. the current neighborhood it has to hold $\tilde{E}(M_{t'+1}, \mathcal{N}^k_{M_{t'}}) = \tilde{E}(M_{t+1}, \mathcal{N}^k_{M_t})$. Because $\tilde{E}$ decreases monotonously, $\tilde{E}$ has to be constant for all iterations between $t$ and $t'$. No two successive neighborhoods of $\mathcal{N}_{M_t}, \ldots, \mathcal{N}_{Mt'}$ are the same, otherwise the algorithm would stop. But according to Theorem 1, $\tilde{E}$ decreases strictly whenever the target neighborhood changes.

Therefore, we conclude that algorithm 1 searches through the possible target neighborhoods without repetition, until a local optimum is achieved. As only a finite number of target neighborhoods exist, convergence is achieved after a finite number of steps. □

# 6 Experiments

In order to assess multi-pass LMNN experimentally, we applied the current version (3.0) of the LMNN toolbox provided by Weinberger [12] in several iterative runs. Note that this recent version is a gradient-boosted variant of the optimization, unlike the original suggestion. As in the original paper, we set the neighborhood parameter to $k = 3$ for LMNN, and evaluated the performance of a $k$-nearest neighbor classifier on the learned metric after each iteration in a 10-fold cross-validation. For the sake of practicality, we did not run the algorithm

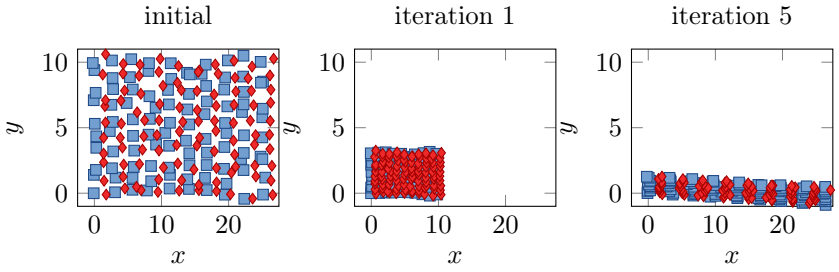


Figure 2: The initial zebra stripes dataset, as well as the projected data points $L^T \cdot x_i$ after the first iteration and the last iteration.

| dataset | $N$ | $n$ | train error | std. | test error | std. |
|---|---|---|---|---|---|---|
| zebra | 200 | 2 | 0.019 | 0.004 | 0.015 | 0.023 |
| iris | 128 | 4 | 0.024 | 0.008 | 0.040 | 0.053 |
| wine | 152 | 13 | 0.000 | 0.000 | 0.021 | 0.028 |
| bal | 535 | 4 | 0.063 | 0.019 | 0.073 | 0.036 |
| isolet | 7,797 | 617 | 0.000 | 0.000 | 0.030 | 0.003 |
| letters | 20,000 | 16 | 0.002 | 0.000 | 0.027 | 0.005 |

Table 1: The number of data points $N$, the number of features/dimensions $n$, and the resulting classification error for each of the experimental data sets. The classification error is given for training and test set respectively, with standard deviation.

until convergence but stopped after 5 iterations.

**Artificial Data:** To illustrate a typical situation where multi-pass LMNN is superior to single-pass LMNN we use a two-dimensional dataset suggested in Weinberger and Sauls original paper, namely a zebra-striped pattern, where stripes of points of the first and the second class alternate [12] (see Figure 2, left). Such a dataset does not only highlight the value of a localized cost function, it also illustrates the importance of updating the target neighborhood. In the initial configuration, some of the target neighbors belong not to the same stripe, but to a different stripe, which makes the LMNN cost function under constant neighborhood assumption hard to optimize. However, after a first pass of LMNN metric learning, we expect that the learned metric "shrinks" the y dimension of the dataset, such that points in the same stripe move closer together. Thereby, more target neighbors belong to the same stripe and the LMNN cost function becomes easier to optimize.

Indeed, we observe this effect in the experimental evaluation. In each successive pass the y dimension shrinks, thereby increasing the accuracy of a $k$-NN classifier. In Figure 2 we show the data as projected by the matrix $L$ after each iteration. Figure 3 (left) displays the training and test error versus LMNN iteration, averaged in a 10-fold cross-validation.

**Real datasets:** In order to assess the performance on real data we also repeated most of the experiments with multi-pass LMNN reported in [12]. In particular, we experimented on the USPS letter dataset, the isolet dataset, the iris

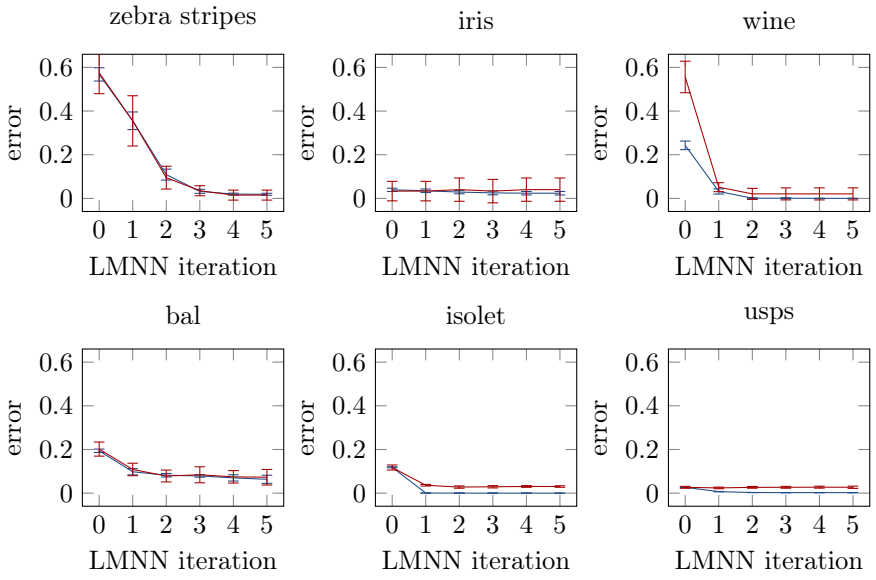


Figure 3: The classification error on the training (blue) and on the test set (red) plotted for all datasets, averaged over 10 cross-validation trials. The x-axis shows the current LMNN iteration. The error bars signify the standard deviation across trials.

dataset, the bal dataset and the wine dataset. Statistics regarding the datasets as well as the final classification error are shown in Table 1. The development of the classification error over time is displayed in Figure 3.

All in all, we observe no strong benefit of multi-pass LMNN over 1-pass LMNN. However, we also did not observe noticeable over-fitting effects as reported by [12], which is likely due to relatively early stopping with five iterations.

# 7 Conclusion

We have shown that local optima of the LMNN cost function can be found using multi-pass LMNN. We have also demonstrated that data sets, for which an adapted metric changes the structure of the target neighborhood, can profit noticeably from multiple passes of LMNN metric learning. As a simple formula, multi-pass LMNN can be considered to be beneficial if the ideal target neighborhood is not obvious to the original metric. Interestingly, this benefit seems to be rather minor in the tested real datasets. Also, we did not notice (strong) over-fitting effects as reported by [12].

Overall, we conclude that multi-pass LMNN is a relatively risk-free and easy-to-use extension of classic LMNN approach that can be easily combined with other extensions of choice and comes with a theoretical convergence guarantee, which the original LMNN approach does not provide. Additionally, it might lead to noticeable performance improvements in datasets, where the initial target neighborhood leads to suboptimal learning impulses.

**Acknowledgments.**

Funding by the DFG under grant number HA 2719/6-2 and the CITEC center of excellence (EXC 277) is gratefully acknowledged.

# References

[1] Bellet, A., Habrard, A., Sebban, M.: A survey on metric learning for feature vectors and structured data. ArXiv e-prints (2013), http://arxiv.org/abs/1306.6709

[2] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York, NY, USA (2004)

[3] Chen, Q., Sun, S.: Hierarchical large margin nearest neighbor classification. In: Pattern Recognition (ICPR), 2010 20th International Conference on. pp. 906–909 (Aug 2010)

[4] Dikmen, M., Akbas, E., Huang, T.S., Ahuja, N.: Pedestrian recognition with a learned metric. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) Computer Vision – ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part IV. pp. 501–512. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[5] Do, H., Kalousis, A., Wang, J., Woznica, A.: A metric learning perspective of svm: on the relation of lmnn and svm. Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) pp. 308–317 (2012)

[6] Guillaumin, M., Verbeek, J., Schmid, C.: Is that you? metric learning approaches for face identification. In: Computer Vision, 2009 IEEE 12th International Conference on. pp. 498–505 (Sept 2009)

[7] Göpfert, C., Paaßen, B., Hammer, B.: Convergence of multi-pass large margin nearest neighbor metric learning. In: 25th International Conference on Artificial Neural Networks (ICANN). pp. 510–517. Springer Nature (2016)

[8] Hosseini, B., Hammer, B.: Efficient metric learning for the analysis of motion data. In: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on. pp. 1–10 (Oct 2015)

[9] Kulis, B.: Metric learning: A survey. Foundations and Trends in Machine Learning 5(4), 287–364 (2013)

[10] Parameswaran, S., Weinberger, K.Q.: Large margin multi-task metric learning. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 1867–1875. Curran Associates, Inc. (2010), http://papers.nips.cc/paper/3935-large-margin-multi-task-metric-learning.pdf

[11] Park, K., Shen, C., Hao, Z., Kim, J.: Efficiently learning a distance metric for large margin nearest neighbor classification. Proceedings of the AAAI Conference on Artificial Intelligence (2011), http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3571

[12] Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (2009), http://dl.acm.org/citation.cfm?id=1577069.1577078

[13] Ying, Y., Li, P.: Distance metric learning with eigenvalue optimization. J. Mach. Learn. Res. 13(1), 1–26 (Jan 2012), http://dl.acm.org/citation.cfm?id=2503308.2188386