The following scientific article was officially published in the journal *IEEE Signal Processing Letters*. This article’s citation is as follows:


doi: [10.1109/LSP.2014.2301037](http://dx.doi.org/10.1109/LSP.2014.2301037)

The manuscript, as accepted by the publisher, is reproduced here, as it appears in the first author’s Ph.D. thesis, entitled *Modélisation physique des tissus mous du tronc scoliotique pour la simulation de l'apparence post-chirurgicale*. The thesis citation is as follows:


© 2014 Kondo Claude Assi. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit: [http://creativecommons.org/licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)
6.1 Abstract

The main objective of this letter is to formulate a new approach of learning a Mahalanobis distance metric for nearest neighbor regression from a training sample set. We propose a modified version of the large margin nearest neighbor metric learning method to deal with regression problems. As an application, the prediction of post-operative trunk 3D shapes in scoliosis surgery using nearest neighbor regression is described. Accuracy of the proposed method is quantitatively evaluated through experiments on real medical data.

6.2 Introduction

The k-nearest neighbors (k-NN) rule [70] is one of the oldest and simplest methods in statistical prediction. Nearest neighbor regression consists in assigning to a new data point the response of the most similar in a dataset [73]. In k-NN regression, the output variable is predicted as a weighted average of the k nearest observations in a dataset, where the neighborhood is defined in terms of a chosen distance metric. Applications of k-NN methods range from computer vision, image retrieval and classification, to face recognition [108], speech recognition [109], human activity recognition and pose estimation, text analysis, and wireless sensor networks [110].

One of the key point in nearest neighbor based methods is to define a distance measure in the input space to identify nearest neighbors, and this mostly depends on the domain application. The default distance metric often used is the Euclidean distance. However ideally, each application requires a specific adapted distance metric since nearest neighbor methods have been demonstrated to have improved performance when used with a learned appropriate distance metric from a sample examples. One of the most learned metric is the Mahalanobis distance, and one of the most widely used Mahalanobis distance learning methods for k-NN
classification is the large margin nearest neighbor (LMNN) proposed by Weinberger et al. [75, 77]. Other metric learning methods for classification have also been proposed by different researchers such as adaptive metric nearest neighbor (ADAMENN) [78], and discriminant adaptive nearest neighbor (DANN) [79]. However these approaches have the disadvantage of requiring more than one parameter to be tuned, which make them less attractive compared to LMNN. The LMNN method has since been extended to other distances, for example in the $\chi^2$-LMNN [111] where the $\chi^2$ histogram distance is used in place of the Mahalanobis distance. Unfortunately most of the metric learning for k-NN approaches are essentially designed for classification problems [112]. Although nearest neighbors regression play an important role in statistical prediction [72], to the best of our knowledge, very few metric learning methods were proposed for nearest neighbor regression problems. The k-NN regression gives fairly similar performance as linear regression with respect to the average RMSEs in some applications, and it would be interesting to design an appropriate metric learning algorithm for k-NN regression.

In this letter, we propose a metric learning method for k-NN regression. We extend the LMNN method proposed in [75, 77] to the case of nearest neighbor regression. Although based on the same framework, however, our method introduces new features to deal with the specific case of regression, which otherwise can not be addressed in the classical LMNN. We then apply our model to the nearest neighbor prediction of postoperative 3D trunk shapes of scoliotic patients.

6.3 Modified large margin nearest neighbor metric learning

Let $D_n = \{z_i = (x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, ..., n\}$ be a dataset where $\mathcal{X}$ and $\mathcal{Y}$ are some metric spaces. We will refer to $\mathcal{X}$ and $\mathcal{Y}$ as the input and response space, respectively. The desired properties of distance metrics for regression are expressed as follows. Intuitively, in order to reduce regression prediction error, one may wish that two inputs $x_i$ and $x_j$ are close one to another in the input space if their respective responses $y_i$ and $y_j$ are also close one to another in the response space. More specifically, we would like, for any triplet of pairs $(x_i, y_i), (x_j, y_j), (x_l, y_l)$, if $x_i$ is much closer to $x_j$ than to $x_l$ with respect to a distance $\delta_{\mathcal{X}}$ defined on $\mathcal{X}$ then it is likely $y_i$ is much closer to $y_j$ than to $y_l$ with respect to a chosen error distance $\delta_{\mathcal{Y}}$ in the response space. In this case, the proximity order of the triplet is preserved.

6.3.1 Intuition of our modified LMNN

We base our model on the following intuitions to insure an accurate nearest neighbor regression: (i) each training input $x_i$ and its $k$ nearest neighbors should preserve proximity order,
Figure 6.1 Illustration of the intuition behind the modified LMNN metric learning for regression. The point \( x_j \) is referred to as the target neighbor. The point \( x_l \) is referred to as an impostor since it violates the proximity order preservation (in this case). The training consists in finding a learned metric \( \delta_X \) such that: (1) the target neighbor is pushed closer to the input query point within a smaller radius after training, (2) impostor is pushed outside the smaller radius domain by a finite margin.

(ii) for a given triplet \((x_i, x_j, x_l)\), training inputs \( x_i \) that violate proximity order should be widely separated from \( x_i \) in such a way that proximity order is restored. Borrowing from the same terminology as in [75], for an input \( x_i \) with response \( y_i \) and target neighbor \( x_j \), we call an impostor any input \( x_l \) with response \( y_l \) such that

\[
\begin{align*}
\delta_Y(y_i, y_l) &> \delta_Y(y_i, y_j) \\
\delta_X(x_i, x_l) &\leq \delta_X(x_i, x_j) + \epsilon,
\end{align*}
\]

where \( \epsilon > 0 \) is the margin. Specifically, an impostor \( x_l \) is any input violating proximity order and that invades the perimeter within a \( \epsilon \)-margin defined by any target neighbor \( x_j \) of the input \( x_i \). We aim to learn a linear transformation of the input space such that the training inputs satisfy the above mentioned properties. Figure 6.1 illustrates the main idea behind our modified LMNN metric learning for regression. This is a regression oriented adaptation of the idealized error reduction scenario of the classical LMNN [75]. It shows how nearest neighbor regression errors in the original input space are corrected by learning an appropriate linear transformation. Before learning, a training input has both target neighbor and impostor in its local neighborhood. During learning, the impostor is pushed outside the perimeter established by the target neighbor. After learning, the mapped inputs points are such that there exists a finite margin between the perimeter and the impostor, and proximity order in both input and response spaces is restored. From the way they are presented, these

---

\[\text{Target neighbors are selected by using prior knowledge (if available) or by simply computing the k nearest neighbors using Euclidean distance [75].}\]
ideas can be cast into the framework of the large margin nearest neighbor and be stated as two competing terms in our model's loss function, where one term penalizes large distances between nearby inputs that preserve proximity order, while the other term penalizes small distances between inputs which violate proximity order.

6.3.2 Proximity order preservation indicator

The key point of our metric learning approach for regression problems is the introduction of the proximity order preservation concept. Let us define the proximity order function $F_{M,\delta_M}$ for a metric space $M$ equipped with the distance $\delta_M$, as for $\tau_{ijl} = (u_i, u_j, u_l) \in M^3$,

$$F_{M,\delta_M}(\tau_{ijl}) = \delta_M(u_i, u_j) - \delta_M(u_i, u_l).$$

For the triplets of pairs of input and response $((x_i, y_i), (x_j, y_j), (x_l, y_l))$, we denote $t_{ijl} = (x_i, x_j, x_l)$ and $t^*_{ijl} = (y_i, y_j, y_l)$ as the triplets in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let us define the triplets labeling function $\Pi$ as:

$$\Pi_{Y,\delta_Y}(t) := \begin{cases} 1 & \text{if } F_{Y,\delta_Y}(t^*) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.2)$$

The function $\Pi_{Y,\delta_Y}$ assigns 0/1 labels to triplets $(x_i, x_j, x_l)$ based on a chosen distance $\delta_Y$ used for measuring errors in the response space $\mathcal{Y}$. Let $C_{\delta_X}$ be a function on triplets of $\mathcal{X}$ associated to distance $\delta_X$ defined as :

$$C_{\delta_X}(t) := \begin{cases} 1 & \text{if } F_{X,\delta_X}(t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.3)$$

We introduce the proximity order preservation indicator $\nu_{ijl}$, as

$$\nu_{ijl} = \nu(t_{ijl}) = C_{\delta_X}(t_{ijl})\Pi_{Y,\delta_Y}(t_{ijl}). \quad (6.4)$$

Note that $\nu_{ijl}$ is equal to 1 only if the proximity order of the triplet is preserved. A k-NN regression using a distance $\delta_X$ with a lower rate of proximity order violation is likely to have a better accuracy performance (See Appendix, Proposition [ ] and Corrolary [ ]). We use this to define a cost term that penalizes small distances between input points which violate the proximity order.
6.3.3 Loss function

Given a training set of features $x_i$ along with their response $y_i$ with $(x_i, y_i) \in \mathbb{R}^{d_{\text{dim}_1}} \times \mathbb{R}^{d_{\text{dim}_2}}$, $(i = 1, \ldots, n)$, we are interested in learning a Mahalanobis distance metric parameterized by a linear transformation $L$, i.e.

$$D_L(x_i, x_j) = \|L(x_i - x_j)\|^2 = (x_i - x_j)^\top L^\top L(x_i - x_j),$$

(6.5)

which allows an accurate nearest neighbor regression. The parameter matrix $L$ is to be chosen such as to minimize the distance between the mappings of a vector and its $k$ nearest neighbors

$$\mathcal{E}_1(L) = \sum_{ij} \eta_{ij} \|L(x_i - x_j)\|^2,$$

(6.6)

where $\eta_{ij}$ is given by

$$\eta_{ij} = \begin{cases} 1 & \text{if } j \text{ is a target neighbor of } i \\ 0 & \text{otherwise.} \end{cases}$$

The parameter $L$ should also allow that the distance of $x_i$ from a target neighbor $x_j$ be less than its distance from an incorrect neighbor $x_l$ (referred to as an impostor). To this end, we use $\nu_{ijl}$ in (6.4) and consider minimizing a hinge loss over triplets of input vectors

$$\mathcal{E}_2(L) = \sum_{ijl} \eta_{ij}(1 - \nu_{ijl}) \left[\epsilon + \|L(x_i - x_j)\|^2 - \|L(x_i - x_l)\|^2\right]_+,$$

(6.7)

where $\left[z\right]_+ = \max\{z, 0\}$, and $\epsilon$ is the margin. We end up with a total cost function $\mathcal{E}(L)$ that combines the two competing terms $\mathcal{E}_1$ and $\mathcal{E}_2$ using a weight parameter $\mu \in (0, 1)$:

$$\mathcal{E}(L) = (1 - \mu)\mathcal{E}_1(L) + \mu\mathcal{E}_2(L).$$

(6.8)

Since $M = L^\top L \succeq 0$, and by introducing a nonnegative slack variables $\xi_{ijl}$ for each triplet, the minimization of the cost function $\mathcal{E}(L)$ can be formulated as a convex semidefinite pro-
The SDP problem \((6.9)\) resulting from our formulation has the same form as the one proposed by Weinberger et al. [75] for their LMNN. However there are some differences in the terms involved, in particular the introduction of the proximity order preservation indicator \(\nu_{ijl}(6.4)\) in the loss function. Equation \((6.9)\) addresses nearest neighbor regression problems, whereas [75] aimed to solve only nearest neighbor classification problems.

### 6.4 Results

Applications to prediction of scoliotic trunk 3D shapes after spine surgery using nearest neighbor (NN) regression were conducted. For the experiments, some characteristic feature curves of the human scoliotic trunk surface topography were considered. The so-called back valley curves were extracted on preoperative and postoperative trunk surfaces. The shape of this feature curve almost follows that of the spine, but is also influenced by the muscle surrounding the spine and supporting the trunk in the upright posture. The upper and lower end points of the curve are anatomical landmarks corresponding to the spinous process of C7 (seventh cervical vertebra) and L5 (the fifth lumbar vertebra), respectively. This feature curve has the advantage of capturing the changing taking place in the back from the preoperative to the postoperative state. It is indeed the feature curve on the trunk surface whose shape change is the most directly influenced by a spine surgery instrumentation. Figure 6.2 (Left) shows the back valley curve along the spinous processes of a scoliotic patient in the preoperative (red) and the postoperative (blue) status overlayed on the patient postoperative trunk mesh. The associated deformation field is depicted on Figure 6.2 (Right). A dataset of 141 pairs of scoliotic shapes data, from teenagers aged between 11 – 18 years old, is considered. Our proposed Mahalanobis distance metric learning is evaluated using the leave-one-out cross-validation procedure. Each sample point is composed of a pair (the preoperative and postoperative shapes). The predicted curve can then be compared with
Figure 6.2 Left: Overlayed back valley curves (preoperative (blue) and postoperative (red)) on a patient surface mesh. Right: Displacement vector field along the back valley curve from preoperative to postoperative state.

The actual postoperative feature curve and a shape prediction error can be computed. To evaluate our results, we compute the prediction error in terms of the normalized root mean square prediction error, which has the advantage of allowing prediction error to be measured on the same scale for all observations. Quarter polar plots are used to visualize pointwise prediction errors along the back valley curve. The radius represents errors values while the angles $\theta$ (or the points along the arc) correspond each to the location of points along the back valley curve, where $\theta = 0$ is the bottom endpoint and $\theta = 90$ is the upper endpoint. The error unit is relative to the span of the spine, which is set to 1 after a common rigid registration of all trunk shapes data. Distances with better performances have errors graph closer to the origin point of the polar plot.

The proposed learned metric is compared with two other metrics: the default Euclidean distance and an arbitrary Mahalanobis metric which is defined by a random semidefinite positive matrix. The quarter polar plots of the mean errors are presented in Figure 6.3. The mean errors for the learned Mahalanobis are less than the ones for the Euclidean and arbitrary Mahalanobis metrics, all along the back valley, with a maximum mean error (0.043) attained around the mid-level of the trunk for the learned Mahalanobis and a maximum of 0.055, 0.056 for the Euclidean and arbitrary Mahalanobis respectively attained around the mid-lumbar level. A maximum error difference between the learned Mahalanobis and the Euclidean distance is found around the mid-lumbar level (MLL). The mean prediction error at the MLL level decreased significantly (0.016) between NN prediction using the Euclidean metric (Mean ± SD: 0.0531 ± 0.037, $N = 141$) and prediction using learned Mahalanobis metric (Mean ± SD: 0.0371 ± 0.0213, $N = 141$), (two-sample t-test assuming equal variance [113], p < 0.001). It appears that the NN regression using our learned Mahalanobis metric...
Figure 6.3 Quarter polar plots of the mean pointwise prediction errors distribution along the back valley curve of \( N = 141 \) scoliotic patients. (a) Red: Learned Mahalanobis metric, (b) Green: Arbitrary Mahalanobis metric, (c) Black: Euclidean metric.

has the lower errors, on average, for the nearest neighbor regression prediction.

We cast our metric learning for nearest neighbor regression as the SDP problem (6.9), which has the same form as the classical LMNN formulation. However, it is important to mention the main differences. Weinberger et al.’s LMNN was aimed to solve k-NN classification problems and for that the second term of their loss function contains an indicator \( y_{il} \) which expresses whether or not target neighbors have the same label (i.e \( y_{il} = 1 \) if and only if \( y_i = y_l \) and, \( y_{il} = 0 \) otherwise). This indicator is no longer relevant in the case of a k-NN regression problem since the response space is continuous, and had to be replaced by some other appropriate relevant indicator. Here in our formulation, we make use of a triplet indicator \( \nu_{ijl} \) which expresses whether or not proximity order, under the distance used, between target neighbors is preserved from the input space to the response space. This indicator is fixed during the learning process for a given training set, and has the advantage of allowing us to keep the same form of the metric learning SDP as LMNN in [75], while at the same time allowing to deal with regression. Our proposed modified LMNN metric learning method produces a Mahalanobis metric that outperforms the Euclidean metric for nearest neighbor regression on scoliotic trunk 3D shapes data.

6.5 Conclusion

In this letter, we have presented a new metric learning method for regression. The proposed method is an extension of the large margin nearest neighbor metric learning method to tackle nearest neighbor regression problems. It has been successfully applied to the prediction of
APPENDIX: Risk of a distance metric for nearest neighbor regression

In nearest neighbor regression, the response estimate \( \hat{y} \in \mathcal{Y} \) of a new data point \( x \in \mathcal{X} \), is given by

\[
\hat{y} = r_{\text{NN}_{\delta_x}}(x) = y_{k: x_k = \text{NN}_{\delta_x}(x)},
\]

where \( \text{NN}_{\delta_x}(x) \) denotes the nearest neighbor of \( x \) in \( D^X_n = \{ x_i : (x_i, y_i) \in D, i = 1, \ldots, n \} \) with respect to the distance \( \delta_x \). Let \( (X_0, Y_0) \in D \) such that \( X_0 \) is the true nearest neighbor of \( X \) in \( D^X_n \) with respect to \( \delta_x \). We define the error \( \text{err}_{\text{NN}_{\delta_x}} (X, Y) = \delta_Y(Y, r_{\text{NN}_{\delta_x}}(X)) = \|Y - r_{\text{NN}_{\delta_x}}(X)\| \). We are interested in selecting a distance \( \delta_x \) to reduce the expected error \( \mathbb{E}[\text{err}_{\text{NN}_{\delta_x}}] \). From the initial dataset \( D_n \), one can derive a dataset \( \mathcal{D} \) of \( \{0, 1\} \)-labeled triplets

\[
\mathcal{D} = \{ s = (t_{ijl}, \lambda_{ijl}) \in \mathcal{X}^3 \times \{0, 1\}, \quad t_{ijl} = (x_i, x_j, x_l), \lambda_{ijl} = \Pi_{Y, \delta_Y}(t_{ijl}) \}.
\]

When \( C_{\delta_x}(t) \neq \lambda \), a violation of the proximity order occurs. The loss function for the distance \( \delta_x \) is defined for \( s = (t, \lambda) \in \mathcal{X}^3 \times \{0, 1\} \) as

\[
\ell(\delta_x, s) = 1_{\{C_{\delta_x}(t) \neq \lambda\}} = 1 - \nu_t,
\]

where \( \nu_t = C_{\delta_x}(t) \Pi_{Y, \delta_Y}(t) \) is obtained from \[6.4\]. The risk of a distance \( \delta_x \) is then defined as the expected loss:

\[
R(\delta_x) = \mathbb{E}[\ell] = \mathbb{P}_{s = (t, \lambda) \sim P^s} \{ C_{\delta_x}(t) \neq \lambda \}.
\]

The key property of distances supporting our metric learning approach is: the lower the proximity order violation rate, the better the nearest neighbor regression. Let us illustrate this idea on a single triplet. Consider the three points sample set \( D_3 = \{ (x_0, y_0), (x_1, y_1), (x_2, y_2) \} \), and let \( s_0 = (t_{012}, \lambda_{012}) \), with \( t_{012} = (x_0, x_1, x_2) \) and \( \lambda_{012} = 1 \). Without loss of generality, let us choose \( (x_0, y_0) \) as the test point. Suppose that \( \delta_{\mathcal{X}}^{(1)} \) and \( \delta_{\mathcal{X}}^{(2)} \) are two distances in \( \mathcal{X} \) such that \( \ell(\delta_{\mathcal{X}}^{(1)}, s_0) = 0 \) and \( \ell(\delta_{\mathcal{X}}^{(2)}, s_0) = 1 \), i.e., \( \ell(\delta_{\mathcal{X}}^{(1)}, s_0) < \ell(\delta_{\mathcal{X}}^{(2)}, s_0) \). In this case, \( r_{\text{NN}_{\delta_x}^{(1)}}(x_0) = y_1 \), \( r_{\text{NN}_{\delta_x}^{(2)}}(x_0) = y_2 \) and \( \text{err}_{\text{NN}_{\delta_x}^{(1)}}(x_0, y_0) = \|y_0 - y_1\| < \|y_0 - y_2\| = \text{err}_{\text{NN}_{\delta_x}^{(2)}}(x_0, y_0) \). This property is formally stated as:

**Proposition 1.** Let \( \delta_{\mathcal{X}}^{(1)} \) and \( \delta_{\mathcal{X}}^{(2)} \) be two distances in \( \mathcal{X} \) with risk \( R(\delta_{\mathcal{X}}^{(1)}) \) and \( R(\delta_{\mathcal{X}}^{(2)}) \) respec-
tively, such that \( R(\delta_{\mathcal{X}}^{(1)}) < R(\delta_{\mathcal{X}}^{(2)}) \). Then

\[
\mathbb{E}[\text{err}_{\text{NN}_{\delta_{\mathcal{X}}^{(1)}}}] < \mathbb{E}[\text{err}_{\text{NN}_{\delta_{\mathcal{X}}^{(2)}}}].
\] (6.14)

The best distance \( \delta_{\mathcal{X}} \) in \( \mathcal{X} \) is the one with minimum risk. The minimum risk distance \( \delta^* \) is

\[
\delta^* = \arg \inf_{\delta_{\mathcal{X}}} R(\delta_{\mathcal{X}}).
\] (6.15)

and

**Corollary 1.** For all \( \delta_{\mathcal{X}} \in D \),

\[
\mathbb{E}[\text{err}_{\text{NN}_{\delta^*}}] \leq \mathbb{E}[\text{err}_{\text{NN}_{\delta_{\mathcal{X}}}}].
\] (6.16)

Minimizing the distance risk may allow us to design a metric learning method for regression.
RÉFÉRENCES


Journal of Educational Psychology, 24:417—441, 1933.


Press.

Proceedings of IEEE International Conference on Computer Vision, volume 2, pages 

[104] H. Wold. Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least 
Squares Approach. Perspectives in Probability and Statistics. Papers in Honour of M. 
S. Bartlett, 1975.

[105] S. Wold, H. Ruhe, H. Wold, and W.J. Dunn III. The collinearity problem in linear 
regression. the partial least squares (PLS) approach to generalized inverse. SIAM 

[106] V. Pazos, F. Cheriet, L. Song, H. Labelle, and J. Dansereau. Accuracy assessment of 
human trunk surface 3d reconstructions from an optical digitizing system. Medical and 


projection error for face recognition. IEEE signal processing letters, 20(5):443–446, 
2013.

in nearest-neighbor based search for continuous speech recognition. IEEE Signal Pro-

[110] S. Marano, V. Matta, and P. Willett. Nearest-neighbor distributed learning by ordered 

[111] Dor Kedem, Stephen Tyree, Kilian Weinberger, Fei Sha, and Gert Lanckriet. Non-linear 


