ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Adaptive distance measures for sequential data

Bassam Mokbel, Benjamin Paassen, and Barbara Hammer *

CITEC centre of excellence, Bielefeld University, Germany

Abstract. Recent extensions of learning vector quantization (LVQ) to general (dis-)similarity data have paved the way towards LVQ classifiers for possibly discrete, structured objects such as sequences addressed by classical alignment. In this contribution, we propose a metric learning scheme based on this framework which allows for autonomous learning of the underlying scoring matrix according to a given discriminative task. Besides facilitating the often crucial and problematic choice of the scoring matrix in applications, this extension offers an increased interpretability of the results by pointing out structural invariances for the given task.

1 Introduction

Because of the intuitive representation of models in terms of prototypical representatives, prototype-based methods such as learning vector quantization (LVQ) enjoy a wide popularity in application domains, particularly if human inspection and interaction are necessary or life-long model adaptation is considered [10, 7, 6]. Modern LVQ schemes are accompanied by mathematical guarantees about their convergence behavior and generalization ability [11, 12]. One success story of LVQ is connected to powerful metric adaptation schemes which enable an autonomous identification of relevant dimensions and their correlations. This not only enhances their representational power but also facilitates interpretability by means of an attention focus regarding the input features [11].

Most classical LVQ approaches can process vectorial data only, limiting the suitability of these methods in the context of complex data structures such as sequences, trees or graph structures, for which a direct vectorial representation is often not available. Recent developments offer possible extensions of LVQ towards more general data which are represented in terms of (dis)similarities only: kernel LVQ, relational LVQ, or generalizations thereof [4]. Although this opens the way towards complex data structures, it is not clear how to transfer powerful metric adaptation to this setting.

In this contribution, we focus on classification tasks for sequence data. Alignment plays a major role in this context which can efficiently be computed based on dynamic programming. Alignment similarities heavily depend on the underlying scoring function which assigns scores to local (symbolic) comparisons. In bioinformatics, these scores are typically inferred from evolutionary models [13]. However, if semantic information is missing, scoring choices are often ad hoc. A few promising approaches for inverse alignment, i.e. to infer local scores from example alignments, have been proposed [3, 14, 1, 2]. None of them, however, addresses the problem to infer scores directly from a classification task, i.e. to simultaneously build a classification model, data alignments, and an underlying scoring matrix from labeled training data only. We propose an extension of LVQ schemes which enables the autonomous inference in classification tasks.

*Funding by the DFG under grant number HA 2719/6-1 and by the CITEC centre of excellence is gratefully acknowledged.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

2 Sequence alignment

Assume an alphabet Σ is given together with a pairwise dissimilarity measure $d_\lambda(x_i, x_j) = \lambda_{ij}$. We denote sequences over Σ as $\bar{a} = (a_1, \ldots, a_I, \ldots, a_{|\bar{a}|})$ with length $|\bar{a}|$. A (global) *alignment* of sequences $\bar{a}$ and $\bar{b}$ consists of extensions $(\bar{a}^*, \bar{b}^*) \in (\Sigma \cup \{-\})^2$ of the sequences by gaps such that the results have equal length. Extending the dissimilarity d_λ to gap costs, we can define *alignment costs* as

$$d^*(\bar{a}, \bar{b}) = \min \left\{ \sum_{i=1}^{|\bar{a}*|} d_\lambda(a_i^*, b_i^*) \;\middle|\; (\bar{a}^*, \bar{b}^*) \text{ is alignment of } (\bar{a}, \bar{b}) \right\}$$

We assume the normalization $d_\lambda(a_i, a_i) = 0$ and $d_\lambda(a_i, b_j) \geq 0$ for $a_i \neq b_j \in \Sigma \cup \{-\}$. Alignment of gaps is forbidden, which can be realized by setting $d_\lambda(-, -) = \infty$. Denoting $\bar{a}(I) = (a_1, \ldots, a_I)$ and $\bar{b}(J) = (b, \ldots, b_J)$, alignment costs can be computed by dynamic programming based on the Bellman equation

$$\begin{array}{lcl} d^*(\bar{a}(0), \bar{b}(0)) & = & 0 \\ d^*(\bar{a}(I+1), \bar{b}(J+1)) & = & \min\{d^*(\bar{a}(I), \bar{b}(J)) + d_\lambda(a_{I+1}, b_{J+1}), \\ & & \quad d^*(\bar{a}(I+1), \bar{b}(J)) + d_\lambda(-, b_{J+1}), \\ & & \quad d^*(\bar{a}(I), \bar{b}(J+1)) + d_\lambda(a_{I+1}, -)\} \end{array}$$

To obtain a differentiable dissimilarity we approximate min by the function $\text{softmin}(x_1, \ldots, x_n) = \sum_i x_i \cdot \exp(-\beta x_i) / \sum_j \exp(-\beta x_j)$ for $\beta \to \infty$ with the derivative $\text{softmin}'(x_i) = (1 - \beta \cdot (x_i - \text{softmin})) \cdot \exp(-\beta x_i) / \sum_j \exp(-\beta x_j)$.

3 Learning vector quantization for dissimilarities

Generalized LVQ (GLVQ) represents vectors $\vec{a}^i$ by typical prototypes $\vec{w}^j$ with labels $c(\vec{w}^j)$ [11]. Classification uses a winner-takes-all rule, i.e. a data point is mapped to the class of its closest prototype. Training optimizes the costs

$$\sum_{i=1}^{N} \Phi \left(\frac{d^+(\vec{a}^i) - d^-(\vec{a}^i)}{d^+(\vec{a}^i) + d^-(\vec{a}^i)} \right)$$

provided labeled data $(\vec{a}^i, c(\vec{a}^i))$ are given. Φ is monotonic, e.g. the sigmoidal, d^+ refers to the squared distance of $\vec{a}^i$ to the closest prototype with a matching label, d^- refers to the closest prototype with a non-matching label.

For non-vectorial data described by a symmetric matrix D of pairwise dissimilarities, an extension of LVQ schemes to so-called relational LVQ is possible [9, 4]: a pseudo-euclidean vector space with a bilinear form exists where (unknown) vectors $\vec{a}^i$ give rise to the dissimilarities d [9]. Prototypes are implicitly represented as convex combinations $\vec{w}^j = \sum_i \alpha_i^j \vec{a}^i$ with $\sum_i \alpha_i^j = 1$ inducing

$$d(\vec{a}^i, \vec{w}^j) = \sum_l \alpha_l^j d_{il} - 0.5 \sum_{ll'} \alpha_l^j \alpha_{l'}^j d_{ll'}$$

which can be computed based on the coefficients $\vec{\alpha}^j$ and the dissimilarities D only [4]. This allows for an adaptation of the coefficients $\vec{\alpha}^+$ or $\vec{\alpha}^-$ of the closest

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

correct or incorrect prototype, given a sequence index i, without an explicit reference to the implicit embedding $\vec{a}^i$:

$$\begin{aligned}\Delta\alpha_l^+ &\sim -\Phi' \cdot \tfrac{2d^-(\vec{a}^i)}{(d^+(\vec{a}^i)+d^-(\vec{a}^i))^2} \cdot \left(d_{il} - \textstyle\sum_{l'} \alpha_l^+ d_{ll'}\right)\\ \Delta\vec{\alpha}_l^- &\sim +\Phi' \cdot \tfrac{2d^+(\vec{a}^i)}{(d^+(\vec{a}^i)+d^-(\vec{a}^i))^2} \cdot \left(d_{il} - \textstyle\sum_{l'} \alpha_l^- d_{ll'}\right)\end{aligned}$$

If D holds pairwise alignment distances $d^*(\bar{a}, \bar{b})$, this yields LVQ for sequences.

4 Adaptive scoring for alignments

Since sequence alignment crucially depends on a correct choice of the scoring matrix λ, it seems beneficial to adjust the alignment parameters λ together with the prototypes. The LVQ framework enables a simple learning scheme: we can adapt scoring parameters (including gap costs) by gradient techniques. The derivative of the GLVQ costs for a sequence $\bar{a}^i$ with respect to λ_{km} yields

$$\Phi' \cdot \frac{2d^-(\bar{a}^i)}{(d^+(\bar{a}^i) + d^-(\bar{a}^i))^2} \partial d^+(\bar{a}^i)/\partial\lambda_{km} - \Phi' \cdot \frac{2d^+(\bar{a}^i)}{(d^+(\bar{a}^i) + d^-(\bar{a}^i))^2} \partial d^-(\bar{a}^i)/\partial\lambda_{km}$$

with

$$\partial d(\bar{a}^i, \bar{w}^j)/\partial\lambda_{km} = \sum_l \alpha_l^j \partial d_{il}/\partial\lambda_{km} - 0.5 \sum_{ll'} \alpha_l^j \alpha_{l'}^j \partial d_{ll'}/\partial\lambda_{km}$$

where the derivative can be computed based on the same dynamic programming scheme as the alignment itself. Denote the three possible alignment choices as

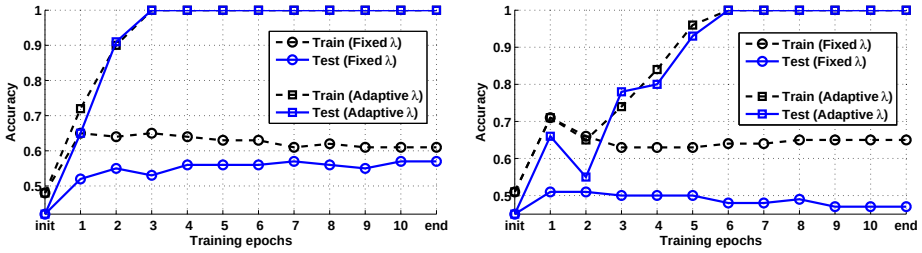
$$\begin{aligned}A_1 &= d^*(\bar{a}(I), \bar{b}(J)) + d_\lambda(a_{I+1}, b_{J+1})\\ A_2 &= d^*(\bar{a}(I+1), \bar{b}(J)) + d_\lambda(-, b_{J+1})\\ A_3 &= d^*(\bar{a}(I), \bar{b}(J+1)) + d_\lambda(a_{I+1}, -)\end{aligned}$$

Then

$$\begin{aligned}\tfrac{\partial d^*(\bar{a}(I+1),\bar{b}(J+1))}{\partial\lambda_{km}} &= \mathrm{softmin}'(A_1) \cdot \left(\tfrac{\partial d^*(\bar{a}(I),\bar{b}(J))}{\partial\lambda_{km}} + \delta_k(a_{I+1})\delta_m(b_{J+1})\right)\\ &+ \mathrm{softmin}'(A_2) \cdot \left(\tfrac{\partial d^*(\bar{a}(I+1),\bar{b}(J))}{\partial\lambda_{km}} + \delta_k(-)\delta_m(b_{J+1})\right)\\ &+ \mathrm{softmin}'(A_3) \cdot \left(\tfrac{\partial d^*(\bar{a}(I),\bar{b}(J+1))}{\partial\lambda_{km}} + \delta_k(a_{I+1})\delta_m(-)\right)\end{aligned}$$

where $\delta_k(a_i)$ tests whether the symbol a_i is element k in Σ. For $\beta \to \infty$, i.e. when the softmin function behaves like a crisp minimum, the derivative with respect to λ_{km} converges to the number of times symbols a_k and a_m are paired in the actual alignment. If we assume that prototypes are close to a data point, one entry α_l^j dominates $\vec{\alpha}^j$, and we can approximate the prototype $\vec{w}^j$ by the corresponding sequence $\bar{a}^l$. Then, $\partial d(\bar{a}^i, \bar{w}^j)/\partial\lambda_{km}$ becomes $\partial d_{il}/\partial\lambda_{km}$. We arrive at an update which (i) decreases costs λ_{km} according to the number of times the pairing $(k, m) \in (\Sigma \cup \{-\})^2$ is observed in an alignment of the sequence $\bar{a}^i$ to the closest prototype sequence with the same label, and (ii) increases λ_{km} according to the number of times (k, m) appears in an alignment of $\bar{a}^i$ and the closest prototype sequence with an incorrect labeling. Thus, the adaptation of the scoring matrix strongly resembles a standard Hebb rule. This approximation also reduces the computational complexity for $\partial d(\bar{a}^i, \bar{w}^j)/\partial\lambda_{km}$ from $\mathcal{O}(N^2 \cdot |\bar{a}| \cdot |\bar{b}| \cdot |\Sigma|^2)$ to $\mathcal{O}(|\bar{a}| \cdot |\bar{b}|)$, N being the number of sequences.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.



(a) Accuracies on *Replacement* data set (b) Accuracies on *Gap* data set

Fig. 1: Classification performance over 10 epochs of training with relational LVQ on the artificial data sets, comparing the use of a fixed choice of a scoring matrix λ with the adaptation of λ between prototype updates in every 3rd epoch.

5 Experiments

We compare the performance of relational LVQ for pairwise sequence alignments D based on a fixed scoring matrix λ vs. the novel scheme as introduced above, incorporating adaptive alignment costs[1]. We use 3 data sets. The class structures in two artificial data sets are designed to demonstrate the abilities to adequately adapt (i) replacement costs and (ii) gap costs. Both sets contain random sequences which follow deliberate structural patterns, allowing class separation for correctly parameterized scoring, but weak separation for other scoring.

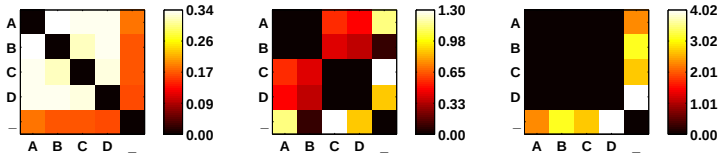
Replacement data: Strings have 12 symbols, which are randomly generated from the 4-letter alphabet $\Sigma = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}$ according to the regular expressions $(\mathsf{A}|\mathsf{B})^5$ $(\mathsf{A}|\mathsf{B})$ $(\mathsf{C}|\mathsf{D})$ $(\mathsf{C}|\mathsf{D})^5$ for the first class, and $(\mathsf{A}|\mathsf{B})^5$ $(\mathsf{C}|\mathsf{D})$ $(\mathsf{A}|\mathsf{B})$ $(\mathsf{C}|\mathsf{D})^5$ for the second. Hence, replacements of A or B by C or D are discriminative, while replacements A with B, and C with D are not. We expect positive gap costs, since gaps could circumvent the alignment of the discriminative middle parts.

Gap data: The second data set focuses on gap scoring. Strings in the first class are random sequences $\bar{a}^k \in \Sigma^{10}$ of length 10, whereas strings $\bar{a}^l \in \Sigma^{12}$ in the second class have length 12. Therefore, replacements of letters are not discriminative, while the introduction of gaps hinders discrimination. Thus, gap costs should be high, while symbol replacements should cost less.

For both data sets, we create 100 training and test data each. The training and test accuracy is evaluated over 10 epochs of relational LVQ training with one prototype per class. The learning rate for the adaptation of λ_{km} is $\eta_{rep} = 0.003$ for replacement scores and $\eta_{gap} = \eta_{rep} \cdot |\Sigma|$ for gap scores, to ensure the same adaptation rate for all parameters. As initialization, we use a standard choice of $\lambda_{km} = 1/(|\Sigma|-1)\ \forall\,(k,m) \in \Sigma^2, k \neq m$, and accordingly, $\lambda_{k\_} = \lambda_{\_k} = 1/(2(|\Sigma|-1))\ \forall\, k \in \Sigma$, adding some noise to break ties in the initial alignments. All self-replacement scores remain fixed $\lambda_{kk} = 0$. During the adaptation, negative values in λ are reset to 0 to keep D non-negative. The experimental results in Fig. 1 show that the classification accuracy is close to random for the initial λ, whereas

[1]The implementation of the alignment was kindly provided by `http://gapc.eu`.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.



(a) Standard/init. λ (b) *Replacement* data (c) *Gap* data

Fig. 2: Visualizations of the scoring matrix λ, where color/intensity encodes the values. On the left is the standard choice (also initial state before adaptation), the middle and right show the final state of λ after adaptation for two data sets.

the adaptation of λ leads to a perfect training and test set accuracy after a few learning epochs. In Fig. 2, the learned λ display ideal scoring matrices.

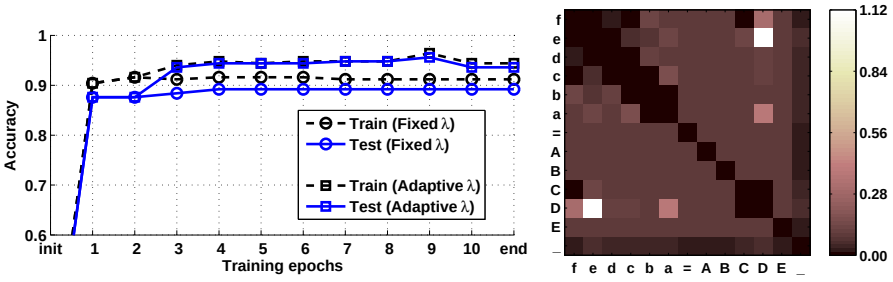
Chromosomes data: The sequences in the set represent band patterns from the Copenhagen Chromosomes database [8]. Every sequence encodes the differential succession of density levels observed in gray-scale images of a human chromosome. Since 7 levels of density are distinguished, a 13-letter alphabet $\Sigma = \{\mathsf{a}, \ldots, \mathsf{f}, =, \mathsf{A}, \ldots, \mathsf{F}\}$ represents a difference coding of successive positions, where upper and lower case letters mark positive and negative changes respectively, and "=" represents no change[2]. Here we restrict to a 3-class problem, using 165 sequences for each of these three chromosomes. (To account for the complexity of the full data set, a local scoring matrix λ_c for every class c would be necessary. This could be handled in a similiar way and is subject of ongoing work.) The meta-parameters and initial setup are the same as described above. The results shown in Fig. 3 demonstrate a clear improvement of the test accuracy by 5% after adaptation of the scoring matrix. Interestingly, λ shows a semantically meaningful pattern, with rather low values in the 1st and 2nd off-diagonals, which resembles the fact that density differences on neighboring scales are exchangeable within classes[3].

6 Discussion

We have introduced a generic scheme for supervised metric adaptation in sequence alignments to facilitate prototype-based classifier training for structures, and we have presented first promising experiments of the technique which demonstrate that relevance learning of scoring parameters for structural metrics is possible. Note that, unlike [5], we do not assume differentiability of the dissimilarity with respect to the data structures itself but differentiability with respect to the adaptive metric parameters only. Thus this approach opens the way towards efficient metric adaptation schemes for distance based approached in discrete structure spaces such as sequences or, as a generalization, trees or graph structures. Besides an improved classification accuracy and generalization ability of the models, this method can enhance the interpretability of the resulting classifiers by pointing out the relevance of structural edit operations.

[2]See details about the data here: `http://algoval.essex.ac.uk/data/sequence/copchrom/`
[3]Note that the symbol F did not occur in our selection and was therefore not considered.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.



(a) Classification accuracies (b) Adapted λ

Fig. 3: Results for the *Chromosomes* data set, where the (semantically sound) adaptation of λ (right) yield an improvement of 5% in test accuracy (left).

So far, we have considered an approximation of the exact gradients by Hebb terms due to the greatly reduced computational complexity; investigating the robustness and accuracy of learning rules based on the exact formulas and better approximations thereof are the subject of ongoing work. First tests using cross-validation on the above data sets confirm the promising results presented here.

References

[1] M. Bernard, L. Boyer, A. Habrard, and M. Sebban. Learning probabilistic models of tree edit distance. *Pattern Recognition*, 41(8):2611–2629, 2008.

[2] L. Boyer, Y. Esposito, A. Habrard, J. Oncina, and M. Sebban. Sedil: Software for edit distance learning. *Lecture Notes in Computer Science*, 5212 LNAI(2):672–677, 2008.

[3] A. Habrard, J. Iñesta, D. Rizo, and M. Sebban. Melody recognition with learned edit distances. *Lecture Notes in Computer Science*, 5342 LNCS:86–96, 2008.

[4] B. Hammer, D. Hofmann, F.-M. Schleif, and X. Zhu. Learning vector quantization for (dis-)similarities. *Neurocomputing*, 2013. In Press.

[5] M. Kästner, D. Nebel, M. Riedel, M. Biehl, and T. Villmann. Differentiable kernels in generalized matrix learning vector quantization. In *ICMLA (1)*, pages 132–137. IEEE, 2012.

[6] S. Kirstein, A. Denecke, S. Hasler, H. Wersing, H.-M. Gross, and E. Körner. A vision architecture for unconstrained and incremental learning of multiple categories. *Memetic Computing*, 1(4):291–304, 2009.

[7] T. Kohonen. *Self-organizing maps*. Springer, 1995.

[8] C. Lundsteen, J. Phillip, and E. Granum. Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.

[9] E. Pekalska and B. Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, 2005.

[10] F.-M. Schleif, B. Hammer, M. Kostrzewa, and T. Villmann. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[11] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[12] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.

[13] V. Sperschneider. *Bioinformatics*. Springer, 2008.

[14] A. Takasu, D. Fukagawa, and T. Akutsu. Statistical learning algorithm for tree similarity. In *Proc. of IEEE Int. Conf. on Data Mining, ICDM*, pages 667–672, Omaha, NE, 2007.