Editorial of the Special issue on Neural Networks and Kernel Methods for Structured Domains

Barbara Hammer [a] Craig Saunders [b] Alessandro Sperduti [c]

[a] *Institute of Computer Science*
Clausthal University of Technology, Germany

[b] *School of Electronics and Computer Science*
University of Southampton, U.K.

[c] *Dipartimento di Matematica Pura ed Applicata*
Università di Padova, Italy

1 Introduction

In a variety of interesting application areas, data can naturally be represented in a structured form such as sequences, trees, or graphs: parse trees in natural language processing, object graphs in scene analysis, text sequences, DNA strings, chemical formulas, 3D protein structures, metabolic or regulatory networks, to name just a few. Thereby, the amount of raw data has increased dramatically during past years, such that machine learning became an important issue for automatic data analysis in these domains. Learning directly in structured domains has traditionally been considered difficult. The only notable exceptions have been time series recognition and forecast, and treatment of relational data in symbolic domains (Inductive Logic Programming).

For large-scale noisy domains, statistical learning techniques such as the support vector machine or neural networks have achieved remarkable results, however, the applicability of such techniques is usually restricted to standard vector spaces. Because of that, data originating from structured domains involving numerical attributes and noise, have usually been coded using a "flat" representation, i.e. vectors of real numbers representing structural features extracted by a preprocessing stage, and then fed as input to one of the methods developed for flat data. The reason for such strategy is in part related to the amenability of vectorial representations to mathematical analysis and exploitation, which is not the case for a fully structured representation.

Although good results have been obtained following this strategy, both computational and generalization concerns have motivated some researchers to develop new techniques to directly deal with structured information. From the computational side, it is not practical to represent all possible structural information into a flat representation, since this would lead to huge real-valued vectors. Moreover, in several structured domains, variance in size of the input structures implies the construction of a general representation scheme that besides being able to cope with average size structures, should also be able to cope with few large structures, introducing input dimensions sparsely populated and thus difficult to evaluate from a learning perspective.

The dilemma is between preserving universality of representation, paying this choice with high space/time complexity and almost sure overfitting, and dropping structural details in the representations, with the potential risk to incur in severe underfitting through loss of information. Thus, flat representations are to be used when there is enough knowledge about the computational task and involved structured domain: only relevant structural features are extracted a priori and efficiently processed by vector-based techniques. In this case, it should not be difficult to define effective feature reduction/selection procedures to reduce the dimensionality of the vector encoding the structural information. It is not currently clear, however, which family of reduction/selection procedures are the most effective for structured domains. When there is no a priori knowledge, however, an approach which tries to preserve as much structural information as possible in the representations, and develops suitable efficient procedures to process these representations, seems to be more sensible.

In this special issue mainly two different but highly interrelated streams of research are explored which share, more or less consciously, this philosophy: Recurrent/Recursive Neural Networks (see for example, (Kolen and Kremer, 2001)) and Kernels for Structures (see (Gartner, 2003) for a short survey). Recurrent/Recursive Neural Networks are based on the following strategy: temporal/structural relationships are represented explicitly and concisely according to the current input structure, although with some limitations in the case of structural data; then an internal task-dependent and task-efficient representation is developed via learning, and concurrently used, in supervised learning, to learn the classifier/regressor of interest. This is obtained by jointly training an encoding function for the structural data, and an output function for classification and/or regression. The problem of variance in size of input structures is solved by weights sharing. Very interesting results from the computational point of view have already emerged for this approach, both concerning supervised and unsupervised learning (e.g. (Hammer, 2000; Hammer et al., 2004)). There is still the need, however, to develop learning procedures which guarantee, at least in probability, the generalization error to be below a specified threshold.

Kernels for Structures, on the other hand, try to exploit the variety and success of kernel methods such as SVMs and use the kernel trick to avoid an explicit representation of the structural features into a vectorial form: since only comparisons among structures is actually required, string/structure matching procedures are directly defined in the structured input domain, without explicitly constructing the (often large) vectorial feature space. A difficulty of this approach is the a priori definition of the kernel so to fit the application domain: for many domains a structured kernel cannot preserve all structural information unless solving NP-hard problems (see (Ramon and Gärtner, 2003)). There have been several approaches which address the problem of designing domain-specific structure kernels. Fisher kernels developed by (Jaakkola et al., 2000) use the Fisher-score vectors of Markov-model parameters as their feature space. Convolution kernels for discrete structures were introduced in (Haussler, 1999), where kernels are based in turn on smaller kernels which compare specific structure parts. At the same time, Watkins (Watkins, 1999) independently proposed a kernel for strings based on comparing all (possibly non-contiguous) k-length subsequences for two input strings. The connection between Fisher kernels and other discrete kernels was highlighted by Saunders et. al (Saunders et al., 2003), where it was shown that string-type kernels have a probabilistic interpretation and equivalent Fisher kernels for the resulting HMMs can be defined. There now exist several general frameworks for building kernels for discrete structures, most notably rational kernels (Cortes et al., 2003) and probability product kernels (Jebara et al., 2004). Perhaps the most successful applications of structure kernels has been in the field of bioinformatics, where several structure kernels such as profile kernels (Kuang et al., 2004), mismatch kernels (Leslie et al., 2004) and local-alignment kernels (Saigo et al., 2004) have been shown to achieve state-of-the-art performance on tasks such as protein-homology detection.

Kernel-based approaches for other types of structures (rather than individually structured training examples) have also been developed. These include diffusion kernels (Kondor and Lafferty, 2002), for when the training examples themselves form part of a structure (e.g. web pages are often related by an ontology). Recently, proposals on generating structured outputs rather than a single label have also been presented and have received a great deal of interest (e.g. (Altun et al., 2003; Tsochantaridis et al., 2004; Taskar et al., 2003)).

2 Scanning the Issue

The special issue is opened by an example of how, knowing a priori the structural information of interest, allows vector-based approaches to be effective. Specifically, Zhao et al. show how a vector-based representation of proteins based on motif content and protein composition turns out to be effective for

protein classification tasks. Each protein is represented by a vector just encoding structural information related to motif content and amino acids statistics. The obtained representations are then processed to extract their principal components, and a subset of them are selected by a wrapper method based on a Genetic Algorithm in combination with a Support Vector Machine: subsets of principal components are selected by a Genetic Algorithm and evaluated on the classification task by a Support Vector Machine. The Genetic Algorithm also provides the "optimal" values for the Support Vector Machine's hyperparameters. The proposed approach has been experimented on classification tasks defined on proteins from the PIR and SCOPE databases and shown to outperform several other approaches, such as decision trees, PSI-BLAST, HRRer, and SVM-pairwise.

The Recurrent/Recursive Neural Networks approach is represented by three papers that propose new models, and two additional papers studying the loading problem for this family of networks, and their relationships with probabilistic graphical models, respectively. Ceroni et al. address the supervised problem of learning protein secondary structure by exploiting both sequential and relational data. The neural model they propose consists of a recursive and bi-directional neural network that takes as input a sequence along with an associated interaction graph, which models a priori knowledge about long-range dependency relations. In their experimental setting, the interaction graph is actually derived from knowledge of protein contact maps at the residue level. The results obtained with this approach show that exploitation of interaction graphs in input can actually and significantly boost the prediction accuracy. Bianchini et al. observe that many problems in pattern recognition require invariance or symmetry. Thus, in the context of a pattern recognition system for object detection in images, they propose a recursive neural network model able to process directed acyclic graphs with labelled edges. Specifically, the encoding function of the proposed recursive network is implemented by a state transition function which considers the edge labels and is independent both from the number and the order of the children of each node. Although the definition of this state transition function may appear to be too specific, they show that the universal approximation capability results obtained for recursive neural networks still hold. Thus the advantage of the proposed model over a standard recursive model is in putting a stronger bias on invariance and symmetry. Experimental results on a task involving face detection in images acquired by an indoor camera show very promising results. Voegtlin proposes a recurrent linear network, trained by Oja's constrained Hebbian learning rule, to represent the temporal context associated to input sequences. The result of training is a generalization of Principal Components Analysis (PCA) to time-series, called Recursive PCA. The author shows that this functionality is supported via a neural implementation of a logical stack. An interesting feature of this network is that sequences stored in the network may be retrieved explicitly. Gori and Sperduti investigate the relationships between the

difficulty of a given learning task and the chosen recursive neural network architecture. They show that, in the case of structured data with categorical labels, and under the assumption that a solution with zero error exists, it is actually possible to define a non trivial upper bound on the number of hidden units to use in order to avoid the presence of local minima. They also stress that this is possible because both the topology of the input structures and the network architecture impose quite informative constraints on the gradient of the error function. Finally, Baldi and Zvi, in a short but informative paper, explain to readers not expert in probabilistic approaches, the formal relationship between Recursive Neural Networks and Probabilistic Directed Graphical Models, including Bayesian Networks. Specifically, the former can be seen as limits, both in distribution and probability, of the latter with local conditional distributions that have vanishing covariance matrices and converge to delta functions. They also derive conditions for uniform convergence and analyze the behavior and exactness of Belief Propagation (BP) in "deterministic" Bayesian Networks.

Kernels Methods for structured data are covered by three papers. Carrozza and Rampone consider regression problems involving graph structured data and propose an incremental supervised learning algorithm for network-based estimators using diffusion kernels. The basic idea is to iteratively add diffusion kernel nodes according to an empirical risk driven rule based on an extended chained version of the Nadaraja-Watson estimator. A genetic-like optimization technique is used as well in this paper to determine the "optimal" values for the hyper-parameters, i.e., the diffusion parameters. Experimental results on classification problems are reported to demonstrate the effectiveness of the proposed approach. Ralaivola et al. recognize that Chemistry is a natural source of structured information and that machine learning can contribute to develop reliable, fast, and non-expensive methods to automatically extract knowledge and meaning from large chemical compound datasets. They focus on kernel methods and derive three new kernels for structures inspired from the work on molecular fingerprinting and based on depth-first searches in graphs. The experimental evaluation reported for the proposed kernels show that they achieve performances at least comparable or superior to those reported in the relevant literature. In the last paper of the special issue, Cuturi and Vert show how ideas and techniques from information theory and data compression can be used to develop a new kernel for strings based on probabilistic suffix trees. A nice feature of the proposed kernel is that it can be computed in linear time and space just using the information contained in the spectrum of the strings to be compared. Promising experimental results for a standard protein homology detection experiment are reported. The authors stress that the proposed kernel performs well with respect to other state-of-the-art methods while using no biological prior knowledge.

3 Future Challenges

The collection of interesting contributions contained in this volume covers relevant theoretical aspects, further developments, and applications of recursive processing of structures, on the one side, and kernels for structures on the other, which constitute two of the most promising current learning paradigms for structures. The articles further the state-of-the-art in these areas and, moreover, they point the way towards future challenging research directions.

While the amount of available data in application areas becomes larger and larger, the development of learning algorithms which are efficient for both training and analysis, becomes ever more important. As discussed in this volume, structural features might simplify training and the loading problem since structural information might account for appropriate presentation of relevant information or it might allow efficient sharing of identical subparts. However, in particular for kernel-based algorithms, principled limitations exist, and the development of problem-specific informative kernel metrics which are computationally efficient enough to be applicable in practice remains a challenge. Hence, an important future direction will rely on the design of fast computation schemes, efficient navigation in the search space e.g. in active or transductive learning, and the automatic design of kernels based on the given task, as already investigated in e.g. (Leslie and Kuang, 2004) and several contributions of this volume. A possible approach, especially for neural networks, could be the exploitation of incremental/constructive learning procedures able to find, on demand, the required tradeoff between computational complexity and accuracy. A potential problem with this approach, however, could be the control of overfitting.

A second important research direction is closely connected to efficient classifier design: since structures often cover only a small part of the input space, a correct bias of learning and appropriate shaping of the search space become crucial for valid generalization. Regularization might be implicit in the architecture or learning scheme, as pointed out e.g. in (Hammer and Tino, 2003), or it can be integrated explicitely in the architecture or kernel e.g. by incorporating appropriate invariances. In this volume, a method to incorporate invariance with respect to permutation of subtrees into recursive networks which preserves universal approximation capability has been proposed. However, the automatic identification of relevant symmetries from data and the integration of often complicated structural invariances into machine learning approaches remains a subject of future research. Note that inference of relevant information and similarities from data as proposed in this volume offer one starting point to infer relevant structural invariances of given data.

A third very important line of research is given by the task to learn structured

outputs. This is the case e.g. for set-valued functions, structural transduction, or structure inference, as occur in various areas such as inference of protein structures or parse trees. Depending on the exact formulation, principled problems and difficulties might occur and only first proposals for such problems can be found in the literature (Hammer, 2000; Hammer et al., 2005; Tsochantaridis et al., 2004; Taskar et al., 2003; Rousu et al., 2005). An adequate modeling of causalities underlying the data is vital in structural transduction, as also demonstrated in this volume for the task of protein structure prediction.

Finally, it is very important to try to exploit, as much as possible, mathematical tools especially devised for structured domains and involving vectorial spaces, such as (Friedman and Tillich, 2004).

References

Altun, Y., Tsochantaridis, I., Hofmann, T., 2003. Hidden markov support vector machines. In: ICML'03. pp. 3–10.

Cortes, C., Haffner, P., Mohri, M., 2003. Rational kernels. In: Becker, S., Thrun, S., Obermayer, A. (Eds.), Advances in Neural Information Processing Systems 15.

Friedman, J., Tillich, J.-P., 2004. Calculus on graphs. CoRR cs.DM/0408028.

Gartner, T., 2003. A survay of kernels for structured data. ACM SIGKDD Explorations Newsletter 5 (1), 49–58.

Hammer, B., 2000. Learning with Recurrent Neural Networks. Vol. 254 of Springer Lecture Notes in Control and Information Sciences. Springer-Verlag.

Hammer, B., Micheli, A., Sperduti, A., 2005. Universal approximation capability of cascade correlation for structures. Neural Computation 17, 1109–1159.

Hammer, B., Micheli, A., Sperduti, A., Strickert, M., 2004. Recursive self-organizing network models. Neural Networks 17 (8-9), 1061–1085.

Hammer, B., Tino, P., 2003. Recurrent neural networks with small weights implement definite memory machines 15 (8), 1897–1929.

Haussler, D., July 1999. Convolution kernels on discrete structures. Tech. Rep. UCSC-CRL-99-10, University of California, Santa Cruz.

Jaakkola, T., Diekhans, M., Haussler, D., 2000. A discriminative framework for detecting remote protein homologies. Journal of Computational Biology 7 (1,2), 95–114.

Jebara, T., Kondor, R., Howard, A., 2004. Probability product kernels. Journal of Machine Learning Research 5, 819–844.

Kolen, J., Kremer, S. (Eds.), 2001. A Field Guide to Dynamical Recurrent Networks. IEEE Press, Inc., New York.

Kondor, R., Lafferty, J., 2002. Diffusion kernels on graphs and other discrete input spaces.

Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., Leslie, C. S., 2004. Profile-based string kernels for remote homology detection and motif extraction. In: 3rd International IEEE Computer Society Computational Systems Bioinformatics Conference (CSB 2004). pp. 152–160.

Leslie, C., Eskin, E., Cohen, A., Weston, J., Noble, W. S., 2004. Mismatch string kernels for discriminative protein classification. Bioinformatics 20 (4), 467–76.

Leslie, C., Kuang, R., 2004. Fast string kernels using inexact matching for protein sequences. Journal of Machine Learning Research 5, 1435–1455.

Ramon, J., Gärtner, T., September 2003. Expressivity versus efficiency of graph kernels. ECML/PKDD'03 workshop proceedings, pp. 65–74.

Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J., 2005. Learning hierarchical multi-category text classification models. In: Proceedings of the 22nd International Conference on Machine Learning (ICML 2005).

Saigo, H., Vert, J., Akutsu, T., Ueda, N., 2004. Protein homology detection using string alignment kernels. Bioinformatics 20, 1682–1689.

Saunders, C., Shawe-Taylor, J., Vinokourov, A., 2003. String Kernels, Fisher Kernels and Finite State Automata. In: Becker, S., Thrun, S., Obermayer, A. (Eds.), Advances in Neural Information Processing Systems 15.

Taskar, B., Guestrin, C., Koller, D., 2003. Max-margin markov networks. In: Neural Information Processing Systems.

Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y., 2004. Support vector machine learning for interdependent and structured output spaces. In: Brodley, C. E. (Ed.), ICML '04: Twenty-first international conference on Machine learning. ACM Press, New York, NY, USA.

Watkins, C., January 1999. Dynamic alignment kernels. Tech. Rep. CSD-TR-98-11, Royal Holloway, University of London.