

Evolutionary kernel machines

Oliver Kramer · Christian Igel · Günter Rudolph

Published online: 1 May 2012
© Springer-Verlag 2012

This special issue on *evolutionary kernel machines* is dedicated to the latest developments in combining kernel-based algorithms and evolutionary computation. Kernel-based methods are among the state-of-the-art approaches in machine learning and data mining (e.g., [3, 4]). The idea of kernel machines is to change the representation of the input data by mapping the input space to some metric feature space. The kernel is a real-valued function of two input space elements that corresponds to a scalar product of its arguments mapped to the feature space [1]. In algorithms using the ‘kernel trick’, all operations in the feature space can be expressed by scalar products—and the kernel replaces all these operations. What are the advantages of using a kernel? The kernel renders an explicit mapping of input data to the feature space unnecessary, and a kernel function can usually be computed much more efficiently than mapping two data points to the feature space and computing their scalar product in the feature space representation. Thus a kernel allows to work in very high-dimensional feature spaces. Further, kernel-based algorithms are very modular and flexible. Apart from hyperparameters (such as learning rates and regularization parameters) only the kernel function has to be changed when the problem domain changes. The kernel defines the feature space and

the algorithm’s notion of similarity between input objects. The clear interface makes handling of non-standard data (e.g., biological sequences) comparatively easy. Finally, the convenient mathematical structure of the kernel-induced feature and hypothesis spaces supports design and analysis of the adaptive system.

The basic optimization problems arising when applying kernel methods—for example, training a support vector machine—are usually solved using mathematical programming. However, in some scenarios, there are good reasons to employ stochastic optimization techniques due to the need to solve problems with local optima, high dimensionality, discontinuities, multiple conflicting objectives, and noise. Examples are the fundamental task of finding a proper kernel function for an application (e.g., [2]), feature selection, or label assignment in semi-supervised learning. Here, evolutionary algorithms come into play, which are applicable to objective functions suffering from the aforementioned characteristics. In addition, they are embarrassingly parallelizable.

This special issue comprises three original research papers. The article by Koch, Bischl, Flasch, Bartz-Beielstein and Koenig studies the problem of automated tuning of support vector machines. It shows how kernels can be evolved by genetic programming, and how sophisticated parameter tuning methods can help to find appropriate hyperparameters. Glasmachers, Koutník and Schmidhuber introduce a kernel-based evolutionary framework for learning continuous functions. Within this framework, nested function spaces of increasing complexity are used to incrementally determine an appropriate level of complexity. Finally, Gieseke, Kramer, Airola and Pahikkala employ stochastic local search schemes to address the combinatorial problem of assigning labels to unlabeled data in semi- and unsupervised kernel-based classification.

O. Kramer (✉)
Oldenburg, Germany
e-mail: oliver.kramer@uni-oldenburg.de

C. Igel
Department of Computer Science,
University of Copenhagen, 2100 Copenhagen, Denmark
e-mail: igel@diku.dk

G. Rudolph
Fakultät für Informatik, TU Dortmund, Dortmund, Germany
e-mail: Guenter.Rudolph@tu-dortmund.de

We would like to thank all authors and reviewers for their great contributions to this special issue, and also thank Larry Bull for his support. Enjoy reading!

References

1. Aronszajn N (1950) Theory of reproducing kernels. *Trans Am Math Soc* 68(3):337–404
2. Igel C (2010) Evolutionary kernel learning. In: Sammut C, Webb GI (eds) *Encyclopedia of machine learning*. Springer, Berlin
3. Schölkopf B, Smola AJ (2002) *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, USA
4. Shawe-Taylor J, Cristianini N (2004) *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge