Evolutionary kernel learning

Christian Igel
Institut für Neuroinformatik
Ruhr-Universität Bochum
D-44780 Bochum, Germany

Definition

Evolutionary kernel learning stands for using evolutionary algorithms to optimize the kernel function for a kernel-based learning machine.

Motivation and Background

In kernel-based learning algorithms the kernel function determines the scalar product and thereby the metric in the feature space in which the learning algorithm operates. The kernel is usually not adapted by the kernel method itself. Choosing the right kernel function is crucial for the training accuracy and generalization capabilities of the learning machine. It may also influence the runtime and storage complexity during learning and application.

Finding an appropriate kernel is a model selection problem. The kernel function is selected from an a priori fixed class. When a parameterized family of kernel functions is considered, kernel adaptation reduces to finding an appropriate parameter vector. In practice, the most frequently used method to determine these values is grid search. In simple grid search the parameters are varied independently with a fixed step-size through a range of values and the performance of every combination is measured. Because of its computational complexity, grid search is only suitable for the adjustment of a few parameters. Further, the choice of the discretization of the search space may be crucial. Gradient-based approaches are perhaps the most elaborate techniques for adapting real-valued kernel parameters, see [1, 2] and references therein. To use these methods, however, the class of kernel functions must have a differentiable structure. They are also not directly applicable if the score function for assessing the parameter performance is not differentiable. This excludes some reasonable performance measures. Evolutionary kernel learning does not suffer from these limitations. Additionally it allows for

```
initialize parent population of individuals,
each encoding kernel and perhaps additional parameters
while termination criterion is not met
    create offspring individuals from parents
    using variation operators
        train and evaluate kernel machine encoded by individuals
        using sample data
    select new parent population based on evaluation
```

Figure 1: Canonical evolutionary kernel learning algorithm.

multi-objective optimization (MOO).

Structure of Learning System

Canonical evolutionary kernel learning can be described as an evolutionary algorithm (EA) in which the individuals encode kernel functions, see Figure 1. These individuals are evaluated by determining the task-specific performance of the kernel they represent. Two special aspects must be considered when designing an EA for kernel learning. First, one must decide how to assess the performance (i.e., the fitness) of a particular kernel. That is, model selection criteria have to be defined depending on the problem at hand. Second, one must also specify the subset of possible kernel functions in which the EA should search. This leads to the questions of how to encode these kernels and which variation operators to employ.

Assessing fitness: Model selection criteria

The following presents some performance indices that have been considered for kernel selection. They can be used alone or in linear combination for single-objective optimization. In MOO a subset of these criteria can be used as different objectives.

It is important to note that, although many of these measures are designed to improve generalization, kernel learning can lead to overfitting if only limited data is used in the model selection process (e.g., in every generation the same small data sets are used to assess performance). Regularization (e.g., in a Bayesian framework) can be used to prevent overfitting. If enough data are available, it is advisable to monitor the generalization behavior of kernel learning using independent data. For example, external data can be used for the early-stopping of evolutionary kernel learning.

Accuracy on sample data The most straightforward way to evaluate a model is to consider its performance on sample data. The empirical risk given by the error on the training data could be considered, but it does not measure generalization. To estimate the generalization performance, the accuracy on data not used for training is evaluated. In the simplest case, the available data is split into a training and validation set, with the first used for learning and the second for subsequent performance assessment. A theoretically sound and simple method is cross-validation (CV). Cross-validation makes better use of the data, but it is more computationally demanding. In practice, it yields very good results.

If classification is considered, it may be reasonable to split the classification error into false negative and false positive rates and to view sensitivity and specificity as two separate objectives [3].

Measures derived from bounds on the generalization performance Statistical learning theory allows one to compute estimates of and bounds on the expected generalization error of learning machines. These values can be utilized as criteria for model selection, although then the assumptions of the underlying theorems from statistical learning theory are typically violated and the terms "bound" and "unbiased estimate" become misleading.

An example where radius-margin bounds are used to evolve kernels for support vector machines (SVMs) is given in [4]. For hard-margin SVMs, the number of support vectors (SVs) is an upper bound on the expected number of errors made by the leave-one-out procedure (e.g., see [1]). It was optimized in combination with the empirical risk for example in [4].

Number of input variables Variable selection refers to the feature selection problem of choosing input variables that are best suited for the learning task. Masking a subset of variables can be viewed as modifying the kernel. By considering only a subset of feature dimensions the computational complexity of the learning machine decreases. When deteriorating feature dimensions are removed, the overall performance may increase. Reducing the number of input variables is therefore a common objective, which can be achieved using single-objective [5, 6, 7, 8] or multi-objective [9, 10] evolutionary kernel learning.

Space and time complexity of the classifier In some applications, it can be desirable to have fast kernel methods (e.g., for meeting real-time constraints). Thus, execution time may be considered in the performance assessment during evolutionary kernel learning.

The space and time complexity of SVMs scales with the number of SVs. This is an additional reason to consider minimization of the number of SVs as an objective in evolutionary model selection for SVMs [4, 3].

Multi-objective optimization The design of a learning machine is usually a MOO problem. For example, accuracy and complexity can be viewed as multiple, and probably conflicting, objectives. The goal of MOO is to approximate a diverse set of Pareto-optimal solutions (i.e., solutions that cannot be improved in one objective without getting worse in another one), which provide insights into the trade-offs between the objectives. Evolutionary multi-objective algorithms have become popular for MOO. Applications of multi-objective evolutionary kernel learning combining some of these performance measures listed above can be found in [4, 9, 10].

Encoding and variation operators

The sheer complexity of the space of possible kernel functions makes it necessary to restrict the search to a particular class of kernel functions. This restriction essentially determines the representation and the operators used in evolutionary kernel learning.

When a parameterized family of mappings is considered, the kernel parameters can be encoded more or less directly in a real-valued EA. This is a frequently used representation, for example for Gaussian kernel functions.

For variable selection a binary encoding can be appropriate. One can fix a kernel $k : X \times X \to \mathbb{R}$ where $k(x, z)$ solely depends on some distance measure between $x, z \in X$. In the binary encoding each bit then indicates whether a particular input variable is considered when computing the distance [9, 10].

Kernels can be built from other kernels. For example, if k_1 and k_2 are kernel functions on X then $ak_1(x, z) + bk_2(x, z)$ or $a \exp(-bk_1(x, z))$ for $x, z \in$

X, $a, b \in \mathbb{R}^+$ are also kernels on X. This suggests a representation in which the individuals encode expressions that evaluate to kernel functions.

Given these different search spaces, it is not surprising that aspects of all major branches of evolutionary computation have been used in evolutionary kernel learning: genetic algorithms [6], genetic programming [11], evolution strategies [4], and evolutionary programming [12].

In general, kernel methods assume that the kernel (or at least the Gram matrix in the training process) is positive semi-definite (psd). Therefore, it is advisable to restrict the search space such that only psd functions evolve. Other ways of dealing with the problem of ensuring positive semi-definiteness are to ignore it [11] or to construct a psd Gram matrix from the matrix M induced by the training data and a non-psd "kernel" function. The latter can be achieved by subtracting the smallest eigenvalue of M from its diagonal entries.

Gaussian kernels Gaussian kernel functions are prevalent. Their general form is $k(x, z) := \exp\left(-(x-z)^{\mathrm{T}} A(x-z)\right)$ for $x, z \in \mathbb{R}^n$ and symmetric positive definite (pd) matrix $A \in \mathbb{R}^{n \times n}$. When adapting A, the issue of ensuring that the optimization algorithm generates only pd matrices A arises. This can be achieved by an appropriate parametrization of A. Often the search is restricted to matrices of the form γI, where I is the unit matrix and $\gamma \in \mathbb{R}^+$ is the only adjustable parameter. However, allowing more flexibility has proven to be beneficial in certain applications (e.g., see [1, 13, 2]). It is straightforward to consider diagonal matrices with positive elements to allow for independent scaling factors weighting the input components. However, only by dropping this restriction one can achieve invariance against both rotation and scaling of the input space. A real-valued encoding that maps onto the set of all symmetric pd matrices can be used such that all modifications of the parameters result in feasible kernels, see [13, 2, 3] for different parametrizations.

Optimizing additional hyperparameters One of the advantages of evolutionary kernel learning is that it can be easily augmented with an optimization of additional hyperparameters of the kernel method. The most prominent example is to encode not only the kernel but also the regularization parameter when doing model selection for SVMs.

Application Example

Notable applications of evolutionary kernel learning include the design of classifiers in bioinformatics [10, 9, 14]. Let us consider [14] as an instructive example. Here, the parameters of a sequence kernel are evolved to improve the prediction of gene starts in DNA sequences. The kernel can be viewed as a weighted sum of 64 kernels, each measuring similarity with respect to a particular tri-nucleotide sequence (codon). The 64 weights $w_1, \ldots, w_{64}$ are optimized together with an additional global kernel parameter σ and a regularization parameter C for the SVM. Each individual stores $x \in \mathbb{R}^{66}$, where $(w_1, \ldots, w_{64}, \sigma, C)^{\mathrm{T}} = (\exp(x_1), \ldots, \exp(x_{64}), |x_{65}|, |x_{66}|)^{\mathrm{T}}$. An evolution strategy is applied, using additive multi-variate Gaussian mutation and weighted global recombination for variation and rank-based selection. The fitness is determined by 5-fold cross-validation. The evolved kernels lead to higher classification rates and the adapted weights reveal the importance of particular codons for the task at hand.

See also

Neuroevolution; Evolutionary Artificial Neural Networks

Recommended Reading

[1] Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46**(1) (2002) 131–159

[2] Glasmachers, T., Igel, C.: Gradient-based adaptation of general gaussian kernels. Neural Computation **17**(10) (2005) 2099–2105

[3] Suttorp, T., Igel, C.: Multi-objective optimization of support vector machines. In Jin, Y., ed.: Multi-objective Machine Learning. Volume 16 of Studies in Computational Intelligence. Springer-Verlag (2006) 199–220

[4] Igel, C.: Multi-objective model selection for support vector machines. In Coello Coello, C.A., Zitzler, E., Hernandez Aguirre, A., eds.: Proceedings of the Third International Conference on Evolutionary Multi-

Criterion Optimization (EMO 2005). Volume 3410 of LNCS., Springer-Verlag (2005) 534–546

[5] Eads, D.R., Hill, D., Davis, S., Perkins, S.J., Ma, J., Porter, R.B., Theiler, J.P.: Genetic algorithms and support vector machines for time series classification. In Bosacchi, B., Fogel, D.B., Bezdek, J.C., eds.: Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V. Volume 4787 of Proceedings of the SPIE. (2002) 74–85

[6] Fröhlich, H., Chapelle, O., Schölkopf, B.: Feature selection for support vector machines using genetic algorithms. International Journal on Artificial Intelligence Tools **13**(4) (2004) 791–800

[7] Jong, K., Marchiori, E., van der Vaart, A.: Analysis of proteomic pattern data for cancer detection. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: Applications of Evolutionary Computing. Volume 3005 of LNCS., Springer-Verlag (2004) 41–51

[8] Miller, M.T., Jerebko, A.K., Malley, J.D., Summers, R.M.: Feature selection for computer-aided polyp detection using genetic algorithms. In Clough, A.V., Amini, A.A., eds.: Medical Imaging 2003: Physiology and Function: Methods, Systems, and Applications. Volume 5031 of Proceedings of the SPIE. (2003) 102–110

[9] Pang, S., Kasabov, N.: Inductive vs. transductive inference, global vs. local models: SVM, TSVM, and SVMT for gene expression classification problems. In: International Joint Conference on Neual Networks (IJCNN 2004). Volume 2., IEEE Press (2004) 1197–1202

[10] Shi, S.Y.M., Suganthan, P.N., Deb, K.: Multi-class protein fold recognition using multi-objective evolutionary algorithms. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, IEEE Press (2004) 61–66

[11] Howley, T., Madden, M.: The Genetic Kernel Support Vector Machine: Description and Evaluation. Artificial Intelligence Review **24**(3) (2005) 379–395

[12] Runarsson, T.P., Sigurdsson, S.: Asynchronous parallel evolutionary model selection for support vector machines. Neural Information Processing – Letters and Reviews **3**(3) (2004) 59–68

[13] Friedrichs, F., Igel, C.: Evolutionary tuning of multiple SVM parameters. Neurocomputing **64**(C) (2005) 107–117

[14] Mersch, B., Glasmachers, T., Meinicke, P., Igel, C.: Evolutionary optimization of sequence kernels for detection of bacterial gene starts. International Journal of Neural Systems **17**(5) (2007) 369–381

Evolutionary algorithms

Synonyms

genetic and evolutionary algorithms, evolutionary computation, evolutionary computing

Definition

Generic term subsuming all machine learning and optimization methods inspired by neo-Darwinian evolution theory.

Kernel matrix

Synonyms

Gram matrix

Definition

Given a kernel function $k : X \times X \rightarrow \mathbb{C}$ and patterns $x_1, \ldots, x_m \in X$, the $m \times m$ matrix K with elements $K_{ij} := k(x_i, x_j)$ is called kernel matrix of k with respect to $x_1, \ldots, x_m$.

Leave-one-out error

Synonyms

hold-one-out error, LOO error

Definition

Given a data set of ℓ patterns, the LOO error is the ℓ-fold cross-validation error.

Model selection

Definition

Model selection is the process of choosing an appropriate mathematical model from a class of models.

Multi-objective optimization

Synonyms

vector optimization, multi-criteria optimization, MOO

Definition

Multi-criteria optimization is concerned with the optimization of a vector of objectives, which can be the subject of a number of constraints or bounds. The goal of multi-objective optimization is usually to find or to approximate the set of Pareto-optimal solutions. A solution is Pareto-optimal if it cannot be improved in one objective without getting worse in another one.

Positive semi-definite

Synonyms

positive definite

Definition

A symmetric $m \times m$ matrix K satisfying $\forall x \in \mathbb{C}^m : x^* K x \geq 0$ is called positive semi-definite. If the equality only holds for $x = \mathbf{0}$ the matrix is positive definite.

A function $k : X \times X \to \mathbb{C}$, $X \neq \varnothing$, is positive (semi-) definite if for all $m \in \mathbb{N}$ and all $x_1, \ldots, x_m \in X$ the $m \times m$ matrix K with elements $K_{ij} := k(x_i, x_j)$ is positive (semi-) definite.

Sometimes the term strictly positive definite is used instead of positive definite and positive definite refers then to positive semi-definiteness.