ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Multi-scale Support Vector Machine Optimization by Kernel Target-Alignment

M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez *

University of Córdoba, Dept. of Computer Science and Numerical Analysis
Rabanales Campus, Albert Einstein building, 14071 - Córdoba, Spain

Abstract. The problem considered is the optimization of a multi-scale kernel, where a different width is chosen for each feature. This idea has been barely studied in the literature, and through the use of evolutionary or gradient descent approaches, which explicitly train the learning machine and thereby incur high computacional cost. To cope with this limitation, the problem is explored by making use of an analytical methodology known as kernel-target alignment, where the kernel is optimized by aligning it to the so-called ideal kernel matrix. The results show that the proposal leads to better performance and simpler models at limited computational cost when applying the binary Support Vector Machine (SVM) paradigm.

1 Introduction

The crucial ingredient of kernel methodologies [1] is undoubtedly the application of the so-called kernel trick, a procedure which maps the data into a higher-dimensional, or even infinite, feature space. The data separation in this space is proved to be easier, and allows the formulation of nonlinear variants of any algorithm which can be cast in terms of the inner products between data points.

Furthermore, the kernel function implicitly determines the feature space $\mathcal{F}$ in such a way that the correct choice of this kernel function becomes an important issue. These choices are related to the definition of a metric between input patterns that fosters correct classification. Usually, a parametrized set of kernels are considered for this purpose, although it is still necessary to choose a performance measure and an optimization strategy. This optimization is often done by means of a grid-search or cross-validation procedure over a previously defined search space, which could be very time-consuming and tedious depending on the size of the dataset, and also computationally unaffordable when considering the multi-scale approach. However, other methodologies for optimizing kernel machine parameters have been studied in state-of-the-art literature, such as evolutionary algorithms [2], those based on the computation of the smallest ball enclosing data [3] or meta-learning approaches [4]. In all the three cases, a large amount of computation is required for optimization, since it involves training the learning machine several times and, in some cases, the solution of an additional optimization problem. In order to overcome this handicap, a different and simpler approach is used, known as kernel-target alignment [5, 6].

*This work has been partially subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain).

This method is independent of the learning algorithm and therefore avoids the expensive computational procedure of training a classifier. Furthermore, the resulting optimal solution for a problem can be plugged into different learning machines. Essentially, kernel-target alignment optimization aims at finding a kernel function k in a restricted family of kernels such that the induced Gram matrix presents the smallest distance to the ideal kernel matrix, which preserves perfectly all the training label structure (represented in this case by similarities between patterns). This methodology has been successfully applied to binary classification problems, as well as to multinomial and regression ones.

Our work exploits the potential advantage of developing a method for tuning the kernel parameters without the need to train the learning machine, which provides the opportunity to learn more complex kernels (i.e. with a multiple set of parameters). To do so, the multi-scale kernel is considered (also known as multi-parametric), where a different kernel parameter is chosen for each feature with a gradient descent approach that optimizes the kernel-target alignment. The main motivation for this contribution is that usually spherical kernels (same weight for each attribute) are used in many real-world applications, where the attributes present very different natures. These kernels have also been studied by means of evolutionary approaches [7, 8] or gradient based methods [9, 10], but almost always considering the training of the learning machine.

The paper is organized as follows: Section II shows a description of the methodology used; Section III describes the experimental study and analyses the results obtained; and Section IV outlines some conclusions and future work.

2 Methodology

Although the properties of the kernel function are important, often the kernel matrix plays a more important role. Since kernel functions allow access to the feature space only via input samples, the pairwise inner products between elements of a finite input set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ are the only information available on the geometry of the feature space. This information is embedded in the kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Gram matrices contain information about the similarity among patterns, thus, the idealized kernel K^* [5] will submit the structure:

$$k^*(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} +1 & \text{if } y_i = y_j \\ -1 & \text{otherwise,} \end{cases} \tag{1}$$

where y_i is the target of pattern $\mathbf{x}_i$. $\mathbf{K}^*$ will provide information about which patterns should be considered similar when performing a learning task.

2.1 Kernel-target alignment

Up to this point, let us suppose an ideal kernel matrix $\mathbf{K}^*$ and a real kernel matrix $\mathbf{K}$ computed for some kernel width α. The Frobenius inner product between them ($\langle \mathbf{K}^*, \mathbf{K} \rangle_{\mathrm{F}} = \sum_{i,j=1}^{N} k^*(\mathbf{x}_i, \mathbf{x}_j) \cdot k(\mathbf{x}_i, \mathbf{x}_j)$, being N the number of patterns) provides information about how *'well'* the patterns are classified in their category. The notion of alignment between $\mathbf{K}$ and $\mathbf{K}^*$ [5, 6] is defined as:

$$A(\mathbf{K},\mathbf{K}^*) = \frac{\langle \mathbf{K},\mathbf{K}^* \rangle_F}{\sqrt{\langle \mathbf{K},\mathbf{K} \rangle_F \langle \mathbf{K}^*,\mathbf{K}^* \rangle_F}}, \tag{2}$$

and this quantity is totally maximized when a kernel is able to reflect the discriminant properties of the dataset used to define the ideal kernel.

Far beyond this formulation, some works have noted several issues in kernel-target alignment for different pattern distribution [5]. This problem has recently been solved by the use of *centered kernel matrices* [6], a method that correlates better with performance than the original definition of kernel-alignment [5]. The centered kernel matrix can be written as: $\mathbf{K}_c = (\mathbf{Z} - \mathbf{Z}\mathbf{1}_{\frac{1}{N}})^\top(\mathbf{Z} - \mathbf{Z}\mathbf{1}_{\frac{1}{N}}) = \mathbf{K} - \mathbf{K}\mathbf{1}_{\frac{1}{N}} - \mathbf{1}_{\frac{1}{N}}\mathbf{K} + \mathbf{1}_{\frac{1}{N}}\mathbf{K}\mathbf{1}_{\frac{1}{N}}$, where $\mathbf{Z} = \begin{bmatrix}\Phi(\mathbf{x}_1) & \cdots & \Phi(\mathbf{x}_n)\end{bmatrix}$, $\Phi(\cdot)$ is the mapping from input patterns to the feature space, and $\mathbf{1}_{\frac{1}{N}}$ is a matrix with all elements equal to $\frac{1}{N}$. $\mathbf{K}_c$ will also be a positive semi-definite kernel matrix fulfilling $k(\mathbf{x},\mathbf{x}) \geq 0, \forall\, \mathbf{x} \in \mathcal{X}$, and symmetry.

2.2 Gradient descent methodology

Due to the differentiability of A with respect to kernel width α (or $\boldsymbol{\alpha}$), a gradient descent algorithm can be used to maximize the alignment between the kernel constructed and the ideal one as follows: $\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} A(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*)$. Since $\boldsymbol{\alpha}$ is a vector comprised of several variables, we will have a gradient vector composed of partial derivatives $\nabla A = \left[\frac{\partial A}{\partial \alpha_1}, \ldots, \frac{\partial A}{\partial \alpha_d}\right]$, where d is the data dimensionality. In this work, the iRprop$_+$ algorithm is used to optimize the aforementioned kernel-target alignment, due to its robustness [11]. Although the second partial derivatives can also be computed and used for optimization, they could actually make this process more computationally costly due to the complexity of this second derivative formula. The alignment derivative with respect to the kernel width α (Eq. 2, but considering the notion of centered kernel) is:

$$\frac{\partial A(\mathbf{K}_\mathrm{c}(\alpha),\mathbf{K}_\mathrm{c}^*)}{\partial \alpha} = \frac{1}{||\mathbf{K}_\mathrm{c}^*||_\mathrm{F}} \left[\frac{\left\langle \frac{\partial \mathbf{K}}{\partial \alpha}, \mathbf{K}_\mathrm{c}^* \right\rangle_\mathrm{F}}{||\mathbf{K}_\mathrm{c}||_\mathrm{F}} - \frac{\langle \mathbf{K},\mathbf{K}_\mathrm{c}^* \rangle_\mathrm{F} \cdot \left\langle \mathbf{K}_\mathrm{c}, \frac{\partial \mathbf{K}}{\partial \alpha} \right\rangle_\mathrm{F}}{||\mathbf{K}_\mathrm{c}||_\mathrm{F}^3} \right], \tag{3}$$

where, for matrices $\mathbf{K}^1$ and $\mathbf{K}^2$, $\left\langle \mathbf{K}_\mathrm{c}^1, \mathbf{K}_\mathrm{c}^2 \right\rangle_\mathrm{F} = \left\langle \mathbf{K}^1, \mathbf{K}_\mathrm{c}^2 \right\rangle_\mathrm{F} = \left\langle \mathbf{K}_\mathrm{c}^1, \mathbf{K}^2 \right\rangle_\mathrm{F}$ [6].

The kernel selected for that purpose is the well-known Gaussian one: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha^2}\right)$, whose derivative can be computed as:

$$\left(\frac{\partial k(\mathbf{x}_i,\mathbf{x}_j)}{\partial \alpha}\right) = \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\alpha^3} \cdot \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha^2}\right). \tag{4}$$

However, we also consider the case of the multi-scale Gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{z=1}^d \frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2}\right) = \prod_{z=1}^d \exp\left(-\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2}\right), \tag{5}$$

whose derivative is the following:

$$\left(\frac{\partial k(\mathbf{x}_i,\mathbf{x}_j)}{\partial \alpha_h}\right) = \frac{(x_{ih} - x_{jh})^2}{\alpha_h^3} \cdot \prod_{z=1}^d \exp\left(-\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2}\right). \tag{6}$$

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

To avoid including positivity constraints in the optimization problem, a logarithmic scale (base 10) is used for parametrization, which does indeed result in a more stable optimization. Due to the local optimality of the iRprop$_+$, several random or even fixed initial points can be considered; in this case, however, for the sake of simplicity, the same initial point was used for optimization (10^0, since it is the mean point of the selected kernel values). Furthermore, the suitable choice of these initial points is a determining factor in the suitability of the solutions.

3 Experimental results

Regarding the experimental setup, a stratified 10-fold cross-validation was applied to divide the data, using the same partitions for all the methods compared. The results are taken as the mean and standard deviation over each one of the 10 test sets. For model selection, a stratified nested 5-fold cross-validation was used on the training sets, with kernel width and C parameter of SVM selected within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

The results of the experiments are shown in Table 1, where the standard cross-validation (CV), kernel-target alignment (KTA) and multi-scale kernel-target alignment (MSKTA) were tested with the SVM paradigm for 7 binary datasets of the UCI. The Table shows the testing results (mean and standard deviation, $mean_{SD}$) in terms of accuracy (Acc), time needed for parameters' optimization (C and α, or C and $\boldsymbol{\alpha}$, for the MSKTA) in seconds (Op_time), cost parameter of the SVM (C), number of support vectors (SVs), and alignment for training (A_{tr}) and testing sets (A_{ts}). Note that to obtain the time results, the algorithms have been run and optimized under the same machine architecture.

Table 1: Results achieved by the three parameter optimization procedures, where p represents the number of patterns of the dataset and f the number of features.

Dataset $\{p, f\}$	Method	Acc(%)	Op_time	C	SVs	A_{tr}	A_{ts}
glassG2 {163,9}	CV	$\mathit{77.87}_{\mathit{8.56}}$	$12.7_{9.1}$	$5.500_{4.743}$	$\mathit{102.7}_{\mathit{9.1}}$	$0.12_{0.01}$	$0.09_{0.03}$
	KTA	$77.83_{9.66}$	$\mathbf{7.5}_{\mathbf{0.7}}$	$5.500_{4.743}$	$120.6_{2.9}$	$\mathit{0.14}_{\mathit{0.01}}$	$\mathit{0.11}_{\mathit{0.03}}$
	MSKTA	$\mathbf{81.54}_{\mathbf{11.67}}$	$\mathit{12.4}_{\mathit{1.4}}$	$2.710_{3.852}$	$\mathbf{96.5}_{\mathbf{5.4}}$	$\mathbf{0.24}_{\mathbf{0.01}}$	$\mathbf{0.19}_{0.06}$
breast-w {699,9}	CV	$\mathit{96.71}_{\mathit{2.43}}$	$\mathit{35.2}_{\mathit{3.3}}$	$1.900_{2.846}$	$\mathit{101.1}_{\mathit{13.6}}$	$0.80_{0.00}$	$0.80_{0.02}$
	KTA	$\mathbf{96.85}_{\mathbf{2.41}}$	$\mathbf{29.6}_{\mathbf{4.8}}$	$0.550_{0.474}$	$\mathit{101.1}_{\mathit{27.6}}$	$\mathit{0.83}_{\mathit{0.01}}$	$\mathit{0.83}_{\mathit{0.02}}$
	MSKTA	$96.57_{2.35}$	$55.3_{6.4}$	$0.550_{0.474}$	$\mathbf{94.6}_{\mathbf{28.5}}$	$\mathbf{0.85}_{\mathbf{0.01}}$	$\mathbf{0.85}_{0.02}$
breast {286,15}	CV	$\mathit{70.65}_{\mathit{5.36}}$	$\mathit{12.5}_{\mathit{1.7}}$	$127.000_{309.015}$	$\mathbf{157.9}_{\mathbf{1.6}}$	$\mathit{0.08}_{\mathit{0.01}}$	$\mathit{0.06}_{\mathit{0.03}}$
	KTA	$70.30_{1.43}$	$\mathbf{9.0}_{\mathbf{2.2}}$	$0.001_{0.000}$	$196.1_{16.2}$	$0.06_{0.00}$	$0.00_{0.01}$
	MSKTA	$\mathbf{72.03}_{\mathbf{3.66}}$	$24_{3.3}$	$13.600_{30.653}$	$\mathit{160.1}_{\mathit{10.5}}$	$\mathbf{0.10}_{\mathbf{0.01}}$	$\mathbf{0.07}_{0.04}$
heart-c {302,22}	CV	$84.09_{8.31}$	$\mathit{16.1}_{\mathit{1.7}}$	$321.400_{469.885}$	$\mathbf{131.3}_{\mathbf{22.8}}$	$0.35_{0.01}$	$0.34_{0.06}$
	KTA	$\mathit{84.42}_{\mathit{6.32}}$	$\mathbf{12.0}_{\mathbf{2.5}}$	$1.900_{2.846}$	$161.8_{21.2}$	$\mathit{0.35}_{\mathit{0.01}}$	$\mathit{0.34}_{\mathit{0.07}}$
	MSKTA	$\mathbf{84.73}_{\mathbf{7.62}}$	$95.9_{15.5}$	$1.630_{2.971}$	$\mathit{155.3}_{\mathit{47.1}}$	$\mathbf{0.46}_{\mathbf{0.01}}$	$\mathbf{0.44}_{\mathbf{0.06}}$
sick {3772,33}	CV	$97.08_{1.06}$	$\mathbf{1728.9}_{\mathbf{537.2}}$	$730.000_{434.741}$	$\mathit{299.4}_{\mathit{30.0}}$	$0.04_{0.00}$	$0.04_{0.00}$
	KTA	$\mathit{97.27}_{\mathit{0.95}}$	$\mathit{2065.9}_{\mathit{423.4}}$	$154.000_{300.599}$	$393.4_{119.5}$	$\mathit{0.05}_{\mathit{0.00}}$	$\mathit{0.05}_{\mathit{0.01}}$
	MSKTA	$\mathbf{97.88}_{\mathbf{1.08}}$	$4201.3_{358.7}$	$343.000_{455.071}$	$\mathbf{160.4}_{\mathbf{31.1}}$	$\mathbf{0.14}_{\mathbf{0.00}}$	$\mathbf{0.14}_{\mathbf{0.02}}$
card {690,51}	CV	$\mathit{84.93}_{\mathit{3.22}}$	$\mathit{74.1}_{\mathit{1.7}}$	$54.100_{48.457}$	$294.7_{75.7}$	$0.16_{0.00}$	$0.16_{0.03}$
	KTA	$\mathit{84.93}_{\mathit{3.63}}$	$\mathbf{53.0}_{\mathbf{10.0}}$	$22.600_{40.959}$	$\mathit{284.6}_{\mathit{40.6}}$	$\mathit{0.17}_{\mathit{0.00}}$	$\mathit{0.17}_{\mathit{0.02}}$
	MSKTA	$\mathbf{86.09}_{\mathbf{4.17}}$	$328.8_{36.9}$	$242.011_{402.010}$	$\mathbf{230.7}_{\mathbf{73.9}}$	$\mathbf{0.56}_{\mathbf{0.01}}$	$\mathbf{0.54}_{\mathbf{0.05}}$
promoters {106,114}	CV	$\mathit{86.09}_{\mathit{7.56}}$	$\mathbf{13.5}_{\mathbf{2.8}}$	$115.300_{312.261}$	$\mathit{83.4}_{\mathit{9.6}}$	$0.18_{0.01}$	$\mathit{0.17}_{\mathit{0.04}}$
	KTA	$83.91_{11.37}$	$\mathit{38.7}_{\mathit{11.0}}$	$14.100_{30.574}$	$\mathit{83.4}_{\mathit{14.1}}$	$\mathit{0.19}_{\mathit{0.01}}$	$0.17_{0.04}$
	MSKTA	$\mathbf{86.82}_{\mathbf{11.85}}$	$146.5_{15.8}$	$8.020_{4.174}$	$\mathbf{49.1}_{\mathbf{16.6}}$	$\mathbf{0.54}_{\mathbf{0.02}}$	$\mathbf{0.48}_{\mathbf{0.10}}$

The best method is in **bold** face and the second one in *italics*.

From these results, several conclusions can be drawn. Firstly, the good performance of the proposal can be seen by analysing Acc, which outperforms (for almost all datasets) the CV and KTA methods. Furthermore, the results achieved by these methodologies were very similar, which indicates that the goodness of MSKTA is due to the multi-scale choice. In terms of time, KTA achieved the best results, except when considering a high number of patterns or features (sick and promoters datasets).

The chosen C parameters for SVM were also reported in Table 1, since they seemed to decrease when kernel-target alignment was used. This cost parameter controlled the trade-off between allowing training errors and forcing rigid margins, in such a way that when $C \to \infty$ the SVM led to the hard-margin approach. Therefore, if C is too large, we will have a high penalty for non-separable points and could store many support vectors, thus leading to overfitting. Because of that, the number of support vectors per model (SVs) was also considered. From that result, it can be noticed that the models obtained by MSKTA seemed to be much simpler (i.e. sparser models) than the ones obtained by CV and KTA, which could be due to the use of a more complex mapping function, therefore leading to a more "ideal" transformation of the input space, using the term ideal in the sense of the kernel mapping leading to a perfectly linear separable set.

Finally, analysing the alignment results (A_{tr} and A_{ts}), there are several issues of note. First, the use of the multi-parametric approach leads to far better alignment. Secondly, training (A_{tr}) and testing alignment (A_{ts}) seem to be highly correlated, in such a way that high alignment values in the training set could be robust enough to determine optimal kernel width without any loss of generality. Last, but not least, similar alignment values were reported for CV and KTA (indeed, several times KTA values were lower than CV ones, which could be due to the local optimality of iRprop$_+$), thus showing the relation between alignment optimization (KTA) and performance optimization (CV).

The non-parametric Friedman's test [12] (with $\alpha = 0.1$) has been applied to the mean Acc rankings, rejecting the null-hypothesis that all algorithms perform similarly. The confidence interval was $C_0 = (0, F_{(\alpha=0.1)} = 2.81)$, and the corresponding F-value was $3.72 \notin C_0$. The Bonferroni-Dunn test has also been applied and the test concluded that there were statistically significant differences for $\alpha = 0.1$, when the MSKTA was selected as the control method, for CV and KTA (the Bonferroni-Dunn critical difference was 1.05 and that obtained for both methods was 1.07).

Not only can the proposed methodology be useful in many real-world applications that present very different attributes, but it also seems to outperform uni-scale approaches (in accuracy) and to obtain sparser models at a reasonable computational cost. Moreover, although omitted due to space restrictions, another advantage of the proposal is that it provides the opportunity to perform feature selection by filtering attributes with large α_i values. The reason is that when kernel width $\alpha_i \to \infty$, the kernel value for that feature remains invariant and tends to one, so that feature i is not used for kernel computation (see Eq. 5), which could be beneficial for model interpretability.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

4 Conclusions

This paper makes use of the kernel-target alignment concept in order to optimize a multi-scale kernel (considering a different width for each feature) for the SVM paradigm. The optimization of the kernel width has almost always been studied by means of the learning machine execution, by a simple "trial and error" procedure which may end up being computationally unaffordable for multiple kernel widths. The results obtained show that kernel-target alignment is highly correlated with performance, and that the optimization of a multi-scale kernel leads to far better results, also in terms of model complexity. As future work, the study could be significantly extended by considering a higher number of datasets. Also, other proposals in the literature can be used for comparison, such as evolutionary algorithms or gradient descent ones based on the radius/margin bound or span of SVMs. Finally, a study of the multi-class case can be carried out by considering kernel-target alignment for the optimization of every single SVM decomposition.

References

[1] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA, 2004.

[2] L. Diosan, A. Rogozan, and J. Pécuchet. Improving classification performance of support vector machine by genetically optimising kernel shape and hyper-parameters. *Appl. Intell.*, 36(2):280–294, 2012.

[3] H. Do, A. Kalousis, A. Woznica, and M. Hilario. Margin and radius based multiple kernel learning. In *Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases: Part I*, pages 330–343. Springer-Verlag, 2009.

[4] C. Soares and P.B. Brazdil. Selecting parameters of svm using meta-learning and kernel matrix-based meta-features. In *Proc. of the ACM symposium on Applied computing*, pages 564–568, 2006.

[5] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *Adv. in Neural Information Processing Systems 14*, pages 367–373. MIT Press, 2002.

[6] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13:795–828, 2012.

[7] J. Gascón-Moreno, E. G. Ortiz-García, S. Salcedo-Sanz, A. Paniagua-Tineo, B. Saavedra-Moreno, and J. A. Portilla-Figueras. Multi-parametric gaussian kernel function optimization for ε-svmr using a genetic algorithm. In *Proc. of the 11th Intern. Conf. on Artificial neural networks*, volume 2 of *IWANN'11*, pages 113–120. Springer-Verlag, 2011.

[8] T. Phienthrakul and B. Kijsirikul. Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In *Proc. of the 2005 Conf. on Genetic and evolutionary computation*, GECCO '05, pages 905–911, 2005.

[9] A. Shamsheyeva and A. Sowmya. The anisotropic gaussian kernel for svm classification of hrct images of the lung. In *Intelligent Sensors, Sensor Networks and Information Processing Conf., 2004. Proc. of the 2004*, pages 439 – 444, 2004.

[10] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

[11] C. Igel and M. Hüsken. Empirical evaluation of the improved rprop learning algorithms. *Neurocomputing*, 50:105–123, 2003.

[12] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.