NOTE

Gradient-based Adaptation of General Gaussian Kernels

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

Gradient-based optimizing of Gaussian kernel functions is considered. The gradient for the adaptation of scaling and rotation of the input space is computed to achieve invariance against linear transformations. This is done by using the exponential map as a parametrization of the kernel parameter manifold. By restricting the optimization to a constant trace subspace, the kernel size can be controlled. This is, for example, useful to prevent overfitting when minimizing radius-margin generalization performance measures. The concepts are demonstrated by training hard margin support vector machines on toy data.

1 Introduction

We consider hyperparameter selection for kernel methods using general Gaussian kernels[1]

$$K_B(x,z) = e^{-\frac{1}{2}(Bx-Bz)^T(Bx-Bz)} \quad , \tag{1}$$

where $x, z \in \mathbb{R}^n$ and B is a positive definite symmetric $n \times n$ matrix. The most elaborated methods for adjusting these kernels are gradient-based approaches (Chapelle et al., 2002; Keerthi, 2002; Chung et al., 2003; Gold and Sollich, 2003) that restrict B to diagonal matrices. However, only by dropping this restriction one can achieve invariance against linear transformations of the input space. A related geometrically inspired idea is carried out by Galleske and Castellanos (2002) for probabilistic neural networks. Indeed it has been shown empirically by direct search that adapting the rotation of Gaussian kernels improves the performance on benchmark problems (Friedrichs and Igel, 2005). Therefore, we compute the gradient for optimizing B in the manifold of positive definite symmetric matrices. It is shown how to decouple adaptation of shape and orientation from the size of the kernel, which becomes necessary, for example, to overcome inherent problems

[1]The more common notation $K(x,z) = e^{-\frac{1}{2}(x-z)^T Q (x-z)}$ is recovered by using the map $B \mapsto B^T B = Q$ which is a diffeomorphism on the manifold of symmetric positive definite matrices. Many implementations of kernel-based methods only support Gaussian kernel with $B^T B = \gamma I$, where I is the unit matrix. Transforming the input space according to B and using the standard Gaussian kernel for training is a simple way to overcome this restriction in practice.

Accepted 2005

when minimizing radius-margin generalization performance measures for support vector machines (SVMs).

2 Kernel Parameterization & Gradient

In order to ensure that gradient-based optimization does not lead to invalid matrices, we use a parameterization that maps symmetric to symmetric and positive definite matrices. Let $\mathfrak{m} := \{A \in \mathbb{R}^{n\times n} \,|\, A = A^T\}$ be the vector space of symmetric $n \times n$ matrices. The manifold $M := \{B \in \mathbb{R}^{n\times n} \,|\, \forall x \neq 0 : x^T Bx > 0 \wedge B = B^T\}$ of positive definite symmetric $n \times n$ matrices can be parameterized using a single map[2], the natural choice is

$$\exp : \mathfrak{m} \to M \qquad A \mapsto \sum_{i=0}^{\infty} \frac{A^i}{i!} \ . \tag{1}$$

It holds $\exp(0) = I$ and $\frac{\partial}{\partial a_{ij}}\Big|_{A=0} \exp(A) = \frac{\partial A}{\partial a_{ij}}$ for each of the $n(n+1)/2$ hyperparameters $a_{ij} = a_{ji}$. The idea is to use at each point B a parametrization that maps the origin $0 \in \mathfrak{m}$ to B. We define the map

$$M \to M \qquad H \mapsto HBH \ , \tag{2}$$

which is a diffeomorphism[3] mapping the unit matrix I to B. To compute the gradient of (1.1), we express B by $\exp(A)B\exp(A)$ with $A = 0$. The trick is that in $A = 0$ the partial derivatives of $\exp(A)$ can be computed easily. We consider $\frac{\partial}{\partial a_{ij}}\Big|_{A=0} K_{\exp(A)B\exp(A)}$ for each hyperparameter $a_{ij} = a_{ji}$:

$$\begin{aligned}
\xi_{ij} &:= \frac{\partial}{\partial a_{ij}}\bigg|_{A=0} K_{\exp(A)B\exp(A)}(x,z) \\
&= \frac{\partial}{\partial a_{ij}}\bigg|_{A=0} e^{-\frac{1}{2}\cdot(x-z)^T \exp(A)^T B^T \exp(A)^T \exp(A)B\exp(A)(x-z)} \\
&= -\frac{1}{2} K_B(x,z) \cdot (x-z)^T \frac{\partial}{\partial a_{ij}}\bigg|_{A=0} \Big(\exp(A)B(\exp(A))^2 B\exp(A)\Big)(x-z) \\
&= -\frac{1}{2} K_B(x,z) \cdot (x-z)^T \left(\frac{\partial A}{\partial a_{ij}}B^2 + 2B\frac{\partial A}{\partial a_{ij}}B + B^2\frac{\partial A}{\partial a_{ij}}\right)(x-z)
\end{aligned}$$

setting $S := \dfrac{\partial A}{\partial a_{ij}}B + B\dfrac{\partial A}{\partial a_{ij}}$ and using $S = S^T$ it follows

$$\begin{aligned}
&= -\frac{1}{2} K_B(x,z) \cdot (x-z)^T (SB + BS)(x-z) \\
&= -\frac{1}{2} K_B(x,z) \cdot \left[(x-z)^T S(B(x-z)) + (B(x-z))^T S(x-z)\right] \\
&= -\frac{1}{2} K_B(x,z) \cdot (x-z)^T (S + S^T)(B(x-z)) \\
&= -K_B(x,z) \cdot (x-z)^T SB(x-z) \ .
\end{aligned} \tag{3}$$

[2]The manifold M is a subset of the Lie group $\mathrm{GL}_n(\mathbb{R})$ and the vector space $\mathfrak{m}$ a corresponding subspace of the Lie algebra $\mathfrak{gl}_n(\mathbb{R})$, cf. Baker (2002).

[3]The map is invertible and the map as well as its inverse map are differentiable.

For example, a simple steepest-descent step with learning rate $\eta > 0$ would lead to the new matrix $\exp(-\eta\xi)B\exp(-\eta\xi)$.

Adapting the parameters changes three properties of the kernel: the shape, which is determined by the eigenvalues of B; the orientation, when non-diagonal matrices B are allowed; and the size, which we define as the smallest volume where a certain amount, say 95 %, of the kernel is concentrated. The size is controlled by the determinant of B.

As we will see, it is sometimes reasonable to restrict the adaptation to kernels with a fixed size. This is achieved by considering the one co-dimensional linear subspace

$$\mathfrak{n} := \{A \in \mathfrak{m} \mid \mathrm{tr}(A) = 0\} \subset \mathfrak{m}$$

of matrices A fulfilling $\det(\exp(A)) = 1$. The gradient descent can be restricted to this subspace[4] by orthogonally projecting the gradient matrices ξ to $\mathfrak{n}$ subtracting $\mathrm{tr}(\xi)/n$ from the diagonal entries of ξ.

3 Application to Radius-Margin Quotients of Hard Margin SVMs

As an example, we consider model selection for hard margin SVMs for binary classification (e.g., Vapnik, 1998). The most frequently used differentiable performance measure, derived from a bound on the generalization error, is the radius-margin quotient

$$\left(\frac{R}{\gamma}\right)^2 \quad , \tag{1}$$

where R denotes the radius of the smallest ball in feature space containing all training examples and γ the margin of the SVM classifier (Schölkopf et al., 1995; Vapnik, 1998; Chapelle et al., 2002; Keerthi, 2002; Chung et al., 2003; Gold and Sollich, 2003). Of course, our approach also works in combination with other performance criteria, such as those discussed by Chapelle et al. (2002). The advantage of (5) is that its gradient can be computed very efficiently (Chapelle et al., 2002; Keerthi, 2002).

As a proof of concept, we consider an artificial chessboard test problem: Each of the ℓ training patterns (x, y) is generated by drawing a vector $\tilde{x} = (x_1, ..., x_d)^T$ from a uniform distribution on $]-2, 2[^d \subset \mathbb{R}^d$ labeled using the rule

$$y = \begin{cases} +1 & \text{if } \sum_{j=1}^{d} \lfloor x_j \rfloor \text{ is even} \\ -1 & \text{if } \sum_{j=1}^{d} \lfloor x_j \rfloor \text{ is odd} \end{cases} .$$

Then x is determined from $\tilde{x}$ by multiplication with a fixed positive definite symmetric matrix $x := B \cdot \tilde{x}$. In the experiments we set $d = 2$, $\ell = 500$, and $B = D^T \cdot \mathrm{diag}(3, \frac{1}{3}) \cdot D$, where in each trial an orthogonal matrix D is drawn randomly from the uniform distribution on the (compact Lie group $\mathrm{O}_n(\mathbb{R})$ of) orthogonal $n \times n$ matrices.

Five different kernel parameterizations are optimized:

[4] $\mathfrak{n} = \mathfrak{m} \cap \mathfrak{sl}_n(\mathbb{R})$ is the subspace of the Lie-Algebra $\mathfrak{sl}_n(\mathbb{R})$ corresponding to $\mathfrak{m}$.

constraint	# variables	impact on kernel
(A) $B = \lambda I$	1	size
(B) B diagonal, $\det(B)$ const.	$n-1$	shape
(C) B diagonal	n	size & shape
(D) $\det(B)$ const.	$n(n+1)/2-1$	shape & orientation
(E) none	$n(n+1)/2$	size, shape, & orientation

Gradient descent (using adaptive individual step-sizes) is performed using the parameterization (2.1), (2.2) with derivative (2.3) combined with the results from Chapelle et al. (2002). The optimization starts from an isotropic kernel whose variance is initialized to the median of the distances from each positive training point to the nearest negative training point, a heuristics suggested by Jaakkola et al. (1999). In (B) and (D) the optimization is restricted to $\mathfrak{n}$.

In all five scenarios the radius-margin generalization performance measure decreases, see Fig. 1. However, in all cases where the size of the kernels is not kept constant, (A), (C), and (E), the number of support vectors drastically increases due to an inherent disadvantage of the optimization criterion: Having a training dataset consisting of ℓ elements, the radius is bounded by $R \leq \sqrt{1-1/\ell} \approx 1-1/(2\ell)$. In many applications this bound is almost reached, such that the derivative of R is comparatively small and the gradient of (3.1) w.r.t. the kernel parameters is governed by the gradient of γ^{-2}. Then, the margin can easily be enlarged by increasing $\det(\exp(A))$, that is, by concentrating the kernel mass to smaller areas in input space. This leads to solutions with smaller (3.1) but increasing number of support vectors. These complex solutions using nearly all points as support vectors are highly adapted to the training dataset and are not desirable, because they tend to overfit (leading to worse test error in (A) and (C), see Fig. 1). This effect can be avoided by early stopping, by changing the optimization criterion (e.g., to the smoothed error of an external validation data set, e.g., see Chapelle et al., 2002), or by controlling the kernel size (e.g., by fixing the trace).

Comparing (C) with (E) and in particular (B) with (D) demonstrates in accordance to Friedrichs and Igel (2005) that better results can be achieved when the kernel adaptation is not restricted to diagonal matrices. The final test error in case (D) is significantly (20 trials, Wilcoxon rank-sum test, $p < 0.01$) lower than in all other cases.

References

Baker, A. (2002). *Matrix groups: An introduction to Lie group theory*. Springer-Verlag.

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

Chung, K.-M., Kao, W.-C., Sun, C.-L., and Lin, C.-J. (2003). Radius margin bounds for support vector machines with RBF kernel. *Neural Computation*, 15(11):2643–2681.

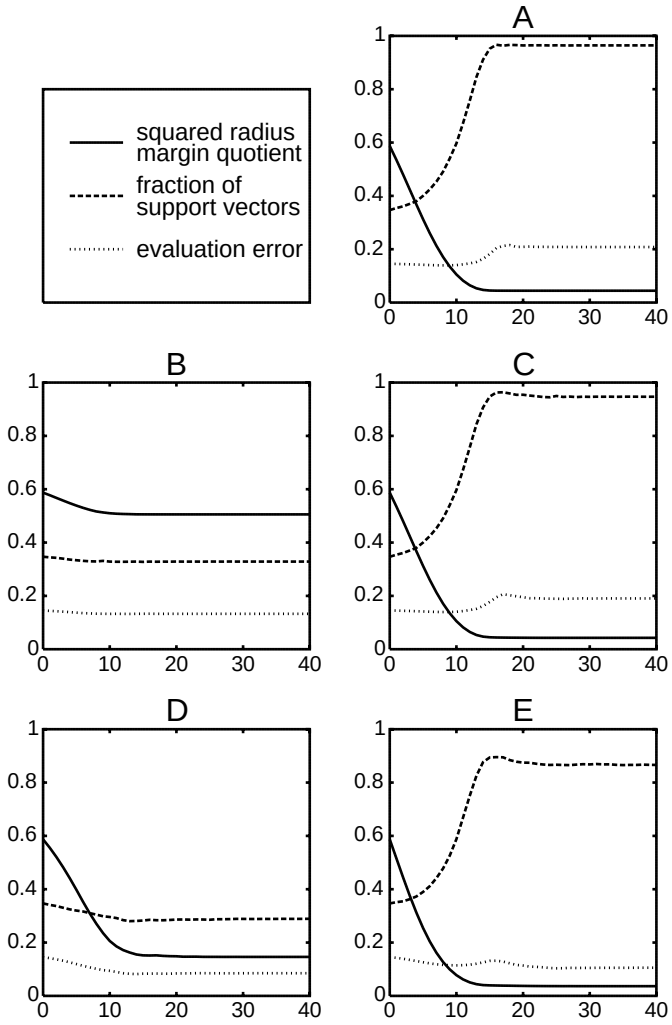

Figure 1: The diagrams show the radius-margin generalization performance measure $(R/\gamma)^2/(\ell \cdot 20)$, the fraction of support vectors, and the test error on a large separate dataset over the number of gradient descent steps. All Quantities are averaged over 20 trials. From left to right: (A) multiples of the unit matrix, (B) diagonal with fixed trace, (C) diagonal, (D) fixed trace, (E) no constraints.

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

Galleske, I. and Castellanos, J. (2002). Optimization of the kernel functions in a probabilistic neural network analyzing the local pattern distribution. *Neural Computation*, 14(5):1183–1194.

Gold, C. and Sollich, P. (2003). Model selection for support vector machine classification. *Neurocomputing*, 55(1-2):221–249.

Jaakkola, T., Diekhaus, M., and Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158. AAAI Press.

Keerthi, S. S. (2002). Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 13(5):1225–1229.

Schölkopf, B., Burges, C. J. C., and Vapnik, V. (1995). Extracting support data for a given task. *Proceedings of the First International Conference on Knowledge Discovery & Data Mining*, pages 252–257. AAAI Press.

Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley, New-York.