A novel SVM Geometric Algorithm based on Reduced Convex Hulls

Michael E. Mavroforakis Margaritis Sdralis Sergios Theodoridis
Informatics and Telecommunications Dept. of the University of Athens, TYPA buildings, Univ. Campus, 15771, Athens, Greece
mmavrof@di.uoa.gr msdralis@di.uoa.gr stheodor@di.uoa.gr

Abstract

Geometric methods are very intuitive and provide a theoretically solid viewpoint to many optimization problems. SVM is a typical optimization task that has attracted a lot of attention over the recent years in many Pattern Recognition and Machine Learning tasks. In this work, we exploit recent results in Reduced Convex Hulls (RCH) and apply them to a Nearest Point Algorithm (NPA) leading to an elegant and efficient solution to the general (linear and non-linear, separable and non-separable) SVM classification task.

1. Introduction

Geometry provides an intuitive and theoretically pleasing framework for the solution of many problems in the fields of Pattern Recognition and Machine Learning. The *Support Vector Machine* (SVM) paradigm in pattern recognition is known to posses certain advantages over a number of alternatives (e.g., [1], [2]).

Although the geometric interpretation of SVM is already known and exposed in, e.g., [3], the main stream of solving methods comes from the algebraic field (mainly decomposition). One of the most popular algebraic algorithms, combining speed and ease of implementation with very good scalability properties, is the Sequential Minimal Optimization (SMO) [4]. The geometric interpretation of SVM in the feature space is a consequence of the dual representation (i.e., the convexity of each class and finding the respective support hyperplanes that exhibit the maximal margin) for the separable case [5], [6] and of the notion of the *Reduced Convex Hull* (RCH) [7] for the non-separable case. Actually, with the exception of [8], the geometric algorithms presented until now ([9], [10]) are suitable only for solving directly the separable case and indirectly the non-separable case through the technique proposed in [11]. However, this technique incorporates not linear but quadratic penalty factors and it has been reported to lead to poor results in practical cases [9]. On the other hand, the application of geometric algorithms to the RCH framework is readily seen to amount to a combinatorial complexity.

The contribution of this work consists of the development and proof of a number of Propositions that allow the use of the popular Gilbert's geometric algorithm – initially proposed for solving the *Nearest Point Problem* (NPP) ([12]) between convex hulls –, to the paradigm of SVM, for both the separable and non-separable classification tasks. The derived geometric algorithm, involving RCH, reduces the complexity from combinatorial to quadratic.

2. SVM and Reduced Convex Hulls

A SVM finds the best separating (*maximal margin*) hyperplane between two classes of training samples in the feature space, which is in line with optimizing bounds concerning the generalization error [1]. The playground for SVM is the *feature space* $\mathcal{H}$, which is a *Reproducing Kernel Hilbert Space* (RKHS), where the mapped patterns live ($\Phi : \mathcal{X} \to \mathcal{H}$). It is not necessary to know the mapping Φ itself analytically, but only its kernel, i.e., the value of the inner products of the mappings of all the samples ($k(x_1, x_2) = (\Phi(x_1) \mid \Phi(x_2))$ for all $x_1, x_2 \in \mathcal{X}$) [13]. Through the "kernel trick", it is possible to transform a nonlinear classification problem to a linear one, but in a higher (maybe infinite) dimensional space $\mathcal{H}$.[1]

This classification task, expressed in its dual form, is equivalent with finding the closest points between

[1] In the rest of this work, for keeping the notation clearer and simpler, the quantities x will be used instead of $\Phi(x)$, since in the final results, the patterns enter only through inner products and not individually, therefore making the use of kernels readily applicable.

the convex hulls generated by the (mapped) patterns of each class in the feature space [5], i.e., it is a Nearest Point Problem (NPP). Finally, in case that the classification task for the given model corresponds to non-separable datasets, i.e., the convex hulls of the (mapped) patterns in the feature space are overlapping, the problem is still solvable, provided that the corresponding hulls are reduced, so that to become non-overlapping [7], [14]. This is illustrated in Fig. 1. Therefore, the need to introduce the framework of the reduced convex hulls has been apparent. Besides, in our case, this framework has to be extended by a set of RCH-related mathematical properties that will allow the complexity of the derived algorithm to be reduced to quadratic levels.

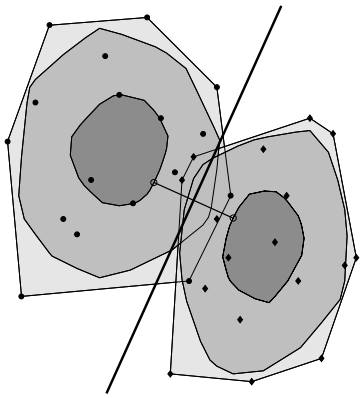

Fig. 1. The initial convex hulls (light gray), generated by the two training datasets (of disks and diamonds respectively) are overlapping; still overlapping are the RCHs with $\mu = 0.4$ (darker gray); however, the RCHs with $\mu = 0.1$ (darkest gray) are disjoint and, therefore, separable. The nearest points of the RCHs, found through the application of the NPA presented here, are shown as circles and the corresponding separating hyperplane as bold line.

Definition 1: The set of all convex combinations of points in some set X, with the additional constraint that each coefficient a_i is upper-bounded by a non-negative number $\mu < 1$ is called the *reduced convex hull* of X and is denoted as $\mathrm{R}(X,\mu)$: $\mathrm{R}(X,\mu) \equiv$

$$\left\{ w : w = \sum_{i=1}^{k} a_i x_i,\ x_i \in X,\ \sum_{i=1}^{k} a_i = 1,\ 0 \le a_i \le \mu \right\}$$

In this way, the initially overlapping convex hulls, with a suitable selection of the bound μ, can be reduced so that to become separable. Once separable, the theory and tools developed for the separable case can be readily applied. The algebraic proof is found in [7] and [5] and the geometric one in [3].

The bound μ, for a given set of original points, plays the role of a reduction factor, since it controls the size of the generated RCH; the effect of the value of bound μ to the size of the RCH is shown in Fig. 1.

Although, at first glance, this is a nice result that lends itself to a geometric solution, i.e., finding the nearest points between the RCHs, it is not a straightforward task: The nearest points between two convex hulls depend directly on their extreme points [15], which, for the separable case are some of the original points. However, in the non-separable case, each extreme point of the RCHs is a reduced convex *combination* of the original points. Hence, a direct employment of a geometric NPA (which obviously depends on the original points) is impractical, since an intermediate step of combinatorial complexity has been introduced. Recently ([8]), two theorems were proved that overcome this computational difficulty, leading to quadratic complexity (i.e., as the standard SVM formulation). Specifically, it was shown that: a) the (candidates to be) extreme points of an RCH are linear combinations of a *specific number* m of the original points ($m = \lceil 1/\mu \rceil$, where $\lceil x \rceil$ denotes the ceiling of x), with weights *specific* coefficients that are analytically computed and b) it is not the extreme RCH points that are needed, but rather their projections onto a specific direction. Then, by exploiting the previous results (stated in a)), an efficient procedure was developed to compute the extreme RCH point with the *minimum projection* by *sorting* the projections of all the original points in *ascending* order and combining appropriately the m smaller of them.

In the sequel, a number of novel Propositions are stated (without proofs due to lack of space), that are necessary in order to extend the results of [8] to Gilbert's algorithm. This algorithm transforms the NPP to a Minimum Norm Problem (MNP) and works on Minkowski difference sets. Thus, it is necessary to prove that the Minkowski difference of two RCHs is a RCH itself, having a direct (and explicit) relation with the original sets. Besides, since Gilbert's algorithm [12] finds at each iteration step the point of a line segment with minimum norm and uses it in the next iteration step, it is necessary to prove that such a point belongs to the RCH.

Proposition 1: The set $-\mathrm{R}(X,\mu)$ is still a RCH; actually it is $\mathrm{R}(-X,\mu)$.

Proposition 2: Scaling is a RCH-preserving property, i.e., $c\mathrm{R}(X,\mu) = \mathrm{R}(cX,\mu)$, $c \in \mathbb{R} - \{0\}$.

Proposition 3: The Minkowski sum of two RCH is also a RCH; actually it is the RCH produced by the Minkowski sum of the original sets, with coefficients'

bound the product of the original coefficients' bounds, i.e., $\mathrm{R}(X_1,\mu_1)+\mathrm{R}(X_2,\mu_2)=\mathrm{R}(X_1+X_2,\mu_1\mu_2)$.

Corollary 1: The Minkowski difference of two RCH $Z=\{z: z=x-y, x\in \mathrm{R}(X_1,\mu_1), y\in \mathrm{R}(X_2,\mu_2)\}$ is the RCH $\mathrm{R}(X_1-X_2,\mu_1\mu_2)$.

Corollary 2: $a_1\mathrm{R}(X_1,\mu_1)+a_2\mathrm{R}(X_2,\mu_2) = \mathrm{R}(a_1X_1+a_2X_2,\mu_1\mu_2)$, $a_1,a_2\in\mathbb{R}-\{0\}$.

Corollary 3: Any point of the line segment joining two arbitrary points of the set $Z=\{z: z=x-y, x\in R(X_1,\mu_1), y\in R(X_2,\mu_2)\}$, belongs to the set, since this is a primitive property of RCH and Z is such.

3. Geometric Algorithm for SVM separable and non-separable tasks

Using the above RCH framework, the general (non-separable) SVM classification task can be formulated as follows [6], [3]: Given a set of patterns $\{x_i\}$ and their corresponding labels $\{y_i\}$, $y_i\in\{-1,1\}$, where $I_1=\{i: y_i=1\}$, $I_2=\{i: y_i=-1\}$, $X_1=\{x_i, i\in I_1\}$, $X_2=\{x_i, i\in I_2\}$, find the couple[2] of points (x_1^*, x_2^*), such that

$$(x_1^*, x_2^*)=\arg\min\left(\|x_2-x_1\|\right), \quad (1)$$

where $x_1\in \mathrm{R}(X_1,\mu_1)$, $x_2\in \mathrm{R}(X_2,\mu_2)$ and assuming that the parameters (μ_1,μ_2) have been selected to guarantee that $\mathrm{R}(X_1,\mu_1)\cap \mathrm{R}(X_1,\mu_1)=\varnothing$. This is clearly a NPP and is equivalent to the following MNP ([9]): find z^*, such that

$$z^*=\arg\min_{z\in Z}\left(\|z\|\right), \quad (2)$$

where

$$Z=\{z: z=x_1-x_2, x_1\in \mathrm{R}(X_1,\mu_1), x_2\in \mathrm{R}(X_2,\mu_2)\}.$$

A brief description of the original Gilbert's algorithm, (provided that Z is a convex set, of which we need to find the minimum norm member z^*), is given below:

Step 1: Choose $w\in Z$.

Step 2: Find the point $z\in Z$ with the minimum projection onto the direction of w. If $\|w\|\cong\|z\|$ then $z^*=w$; stop.

Step 3: Find the point w^{new} of the line segment $[w,z]$, with minimum norm (closest to the origin). Set $w\leftarrow w^{new}$ and go to Step 2.

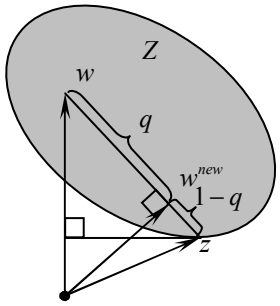

Fig. 2: Elements involved in Gilbert's algorithm.

The idea behind the algorithm is very simple and intuitive and the elements involved in the above steps of the algorithm, are illustrated in Fig. 2.

It turns out that when applying Gilbert's algorithm to RCH, only its basic philosophy remains intact. The fact that the extreme points are now reduced convex combinations of the original points dictates the derivation of a distinctly different algorithmic scheme. More specifically:

1. The set Z, which in this case is the Minkowski difference of the two RCH, generated by the original pattern classes, must be a convex set. Actually, Z is a RCH, as it is ensured by Corollary 1 and therefore a convex set.
2. The initial w of Step 1 has been chosen to be the centroid of the RCH and hence, $w\in Z$.
3. The minimum norm point w^{new}, found at each step, must be a feasible solution, i.e., be the vector joining two points, each in the RCH of the corresponding class. This is ensured by Corollary 3.
4. All vectors involved in the calculations should not be used directly (as in the original Gilbert's algorithm), but only through inner products, since Z is a RKHS. Furthermore, at each step, the best solution found so far (w^{new}) has to be expressed in terms of the original points and their corresponding coefficients (Lagrange multipliers) a_i. Therefore, $w=w_1-w_2=\sum_{i\in I_1}a_ix_i-\sum_{i\in I_2}a_ix_i$.

[2] There may be more than one such couple of points, in case of degeneracy, but the distance between the points of each couple will be the same for all couples.

5. The point $z_r \equiv \arg\min_{z \in Z} \left(\langle z, w \rangle / \|w\| \right)$ (with minimum projection of Z RCH on the direction of w) is computed as follows: Recall that z_r is the difference of two vectors z_{1r} and z_{2r}, each of which is a reduced convex combination of the original points of the corresponding class. Then, by exploiting the two theorems derived in [8], we get $z_{kr} = \mu_k \sum_{i \in \tilde{I}_k(1:end-1)} x_i + \lambda_k x_{\tilde{I}_k(end)}$, where $k \in \{1,2\}$, end is the last element of the specific ordered set and $\lambda_k \equiv 1 - \lfloor 1/\mu_k \rfloor \mu_k$. $\tilde{I}_k$ is the set of the first $\lceil 1/\mu_k \rceil$ indices of the (original) points with sorted projections (in increasing order) onto $w_1 - w_2$ (if $k=1$) or $w_2 - w_1$ (if $k=2$) respectively.
6. The update of the coefficients a_i is computed according to the following Lemma: **Lemma**: The coefficients a_i^{new} of w^{new} (with $i \in I_k$) are $a_i^{new} = (1-q) a_i + q\mu_k \sum_{j \in \tilde{I}_k(1:end-1)} \delta_{ij} + q\lambda_k \sum_{j \in \tilde{I}_k(end)} \delta_{ij}$ where $q = \min\left(1, \langle w, w - z_r \rangle / \|z_r - w\|^2\right)$.

The proofs and the calculations for the steps 5 and 6 above are rather lengthy and the technicalities and details of the implementation of the algorithm are not presented here due to lack of space.

3. Experiments

In order to extensively investigate the performance (both in speed and accuracy) of the new algorithm presented here, a number of publicly available test datasets, with known success rates under specific SVM training models (referred in the literature) have been used. Three different SVM classification algorithms were implemented (in Matlab), tested and compared: the SMO algorithm, as it was presented by Platt in [4] (denoted as SMO), a modified SMO algorithm presented by Keerthi et. al. in [16] (denoted as SMO-K1) and the algorithm presented here (denoted as RCH-G).

Each algorithm was trained and tested for each dataset, under the same model (kernel with the corresponding parameters) in order to achieve the same accuracy referred in the literature ([17], [18]). The accuracies of all classifiers, achieved for each specific dataset, were calculated under the same validation scheme, i.e., the same validation method and the same data realizations (100 publicly available realizations). The results of the runs are summarized in Table 1.

Table 1: Results achieved for each algorithm (number of kernel evaluations and run times in seconds).

Dataset	Method	Success (%)	Kernel Eval.	Time (s)
Diabetis	SMO	76.7±1.8	2.01×10^7	63.8
	SMO-K1	76.7±1.8	1.45×10^6	8.4
	RCH-G	76.3±1.8	6.99×10^5	6.5
German	SMO	76.0±2.2	1.37×10^7	2840
	SMO-K1	76.1±2.2	9.00×10^6	161
	RCH-G	76.3±0.4	2.71×10^6	15
Waveform	SMO	88.8±0.6	9.43×10^7	2005
	SMO-K1	89.2±0.5	2.20×10^6	65
	RCH-G	88.5±0.7	1.46×10^6	23.2
Thyroid	SMO	94.6±2.4	1.27×10^6	31.50
	SMO-K1	94.7±2.5	8.32×10^4	0.98
	RCH-G	95.4±2.1	5.78×10^4	0.52
Heart	SMO	83.9±3.3	2.05×10^6	51.22
	SMO-K1	83.2±4.2	2.62×10^5	1.54
	RCH-G	84.8±3.3	4.70×10^4	0.81

The results of the new geometric algorithm presented here, compared to the most popular and fast algebraic ones, are very encouraging: The difference in the number of kernel evaluations is noticeable. Moreover the difference in the execution times is even more noteworthy. The enhanced performance of the new algorithm is due to the fact that although the algebraic algorithms (especially SMO-K1) make a clever utilization of the cache, where kernel values are stored, they cannot avoid repetitive searches in order to find the best two points to use for each optimization step. Furthermore, the geometric algorithm presented here is a straightforward optimization scheme, with a clear optimization target at each iteration step, always aiming to the global minimum and it is independent of obscure and sometimes inefficient heuristics.

4. Conclusion

A novel geometric algorithm for the solution of the general SVM task has been presented. The algorithm was tested against a number of widely used datasets and resulted in enhanced performance, compared to SMO and one of its popular variants, with respect to kernel evaluations and time requirements.

5. References

[1] N. Cristianini, J. Shawe-Taylor *An introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[2] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, 3rd edition, Academic Press, 2006.

[3] D. Zhou, B. Xiao, H. Zhou, R. Dai "Global Geometry of SVM Classifiers", Technical Report in AI Lab, Institute of Automation, Chinese Academy of Sciences. Submitted to NIPS 2002.

[4] J. Platt "Fast training of support vector machines using sequential minimal optimization" in B. Schölkopf, C. Burges and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pp. 185–208. MIT Press, 1999.

[5] K. P. Bennett, E. J. Bredensteiner "Duality and Geometry in SVM classifiers" in Pat Langley, editor, *Proc. 17th International Conference on Machine Learning*, Morgan Kaufmann, 2000, pp. 57–64.

[6] K. P. Bennett, E. J. Bredensteiner "Geometry in Learning" in C. Gorini, E. Hart, W. Meyer and T. Phillips, editors, *Geometry at Work*, Mathematical Association of America, 1998.

[7] D. J. Crisp, C. J. C. Burges "A geometric interpretation of ν-SVM classifiers" NIPS 12, 2000, pp. 244–250.

[8] M. Mavroforakis, S. Theodoridis "A geometric approach to Support Vector Machine (SVM) classification", to appear in *IEEE Transactions on Neural Networks*, also available from http://cgi.di.uoa.gr/~idsp.

[9] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy "A fast iterative nearest point algorithm for support vector machine classifier design", Technical Report No. TR-ISL-99-03, Department of CSA, IISc, Bangalore, India, 1999.

[10] V. Franc, V. Hlaváč "An iterative algorithm learning the maximal margin classifier", *Pattern Recognition 36*, 2003, pp. 1985–1996.

[11] T. T. Friess, R. Harisson "Support vector neural networks: the kernel adatron with bias and soft margin" Technical Report ACSE-TR-752, University of Sheffield, Department of ACSE, 1998.

[12] E. Gilbert "Minimizing the quadratic form on a convex set", *SIAM J. Control*, vol. 4, 1966, pp 61-79.

[13] B. Schölkopf, and A. Smola, *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*, The MIT Press, 2002.

[14] M. Mavroforakis, and S. Theodoridis, "Support Vector Machine (SVM) Classification through Geometry", *Proceedings of EUSIPCO 2005*, Antalya, Turkey, also available from http://cgi.di.uoa.gr/~idsp.

[15] J.-B. Hiriart-Urruty, C. Lemaréchal, *Convex Analysis and Minimization Algorithms I*, Springer-Verlag, 1991.

[16] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design", Technical report, Dept of CSA, IISc, Bangalore, India, 1999.

[17] G. Rätsch, T. Onoda and K.-R. Müller , "Soft Margins for AdaBoost", *Machine Learning*, vol 42 - 3 , pp 287-320, 2001.

[18] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, " Fisher discriminant analysis with kernels", in Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pp. 41-48. IEEE, 1999.