Journal of Advances in Applied Mathematics, Vol. 2, No. 3, July 2017
https://dx.doi.org/10.22606/jaam.2017.23001

On Sparsity of Soft Margin Support Vector Machines

Jochen Merker[1*]

Faculty of Computer Science, Mathematics and Natural Sciences, University of Applied Sciences Leipzig, Germany
Email: jochen.merker@htwk-leipzig.de

Abstract In supervised machine learning, a support vector machine (SVM) constructs from binary classified training data a linear classifier by solving a linearly constrained convex optimization problem. Depending on the number N of training data and the dimension D of the feature space, it either is advantageous to solve the primal problem or the dual problem. In this article, the case $D >> N$ is discussed where D is so large that even a calculation of the dot product of fully occupied vectors in dimension D is too slow for the desired (e.g. real-time) application. Then a way to speed up the classification is to use an SVM which constructs a sparse linear classifier by solving an optimization problem involving the 1-norm, i.e. many components of the classifying vector are zero so that much less than D multiplications are neccessary to calculate the dot product. For a soft-margin SVM, in this article a theorem on the number of non-zero components is shown.

Keywords: Support vector machines, SVM, sparse classifier, L1 regularization, LASSO, big data, machine learning.

1 Introduction

Among different supervised machine learning methods like decision trees, rule-based classification or Bayesian statistics, *support vector machines* (SVMs) [1] have been proven to be efficient tools in many applications. For example, the review [2] of classification methods for internet traffic emphasizes the key finding of [3] that "their classifier based on a support vector machine (SVM) outperformed other ML algorithms and produced robust results once it was trained with a representative, unbiased training set", and the survey [4] of methods for encrypted traffic classification emphasizes the results of [5] which "demonstrated that SVM methods provide comparable accuracy with less false positives".

While the training of SVMs from N binary classified data points has a rather high complexity, because a convex optimization problem subject to linear inequality constraints has to be solved, an important advantage of SVMs is that once they have been trained, the classification is possible by a simple dot product of vectors, i.e. by a linear classifier. This still holds true, if the data is embedded into a much higher dimensional space via a feature map to make the data linearly separable, and then the kernel trick [6] is used to obtain on the one hand a nonlinear classifier and to avoid on the other hand calculations in the higher dimensional space. A different approach to obtain a nonlinear classifier is suggested in [7], there a combination of support vector machines and decision trees is proposed. In every case, classification is very fast as long as the feature vectors are not too high-dimensional. Yet, the dimension of these feature vectors is identical with the dimension D of the feature space, in which the data points lie. For big data (e.g. high-resolution pictures, internet traffic), this dimension D may be very large, and the purpose of this paper is to investigate analytically, how a speed up of the classification can be made possible by using an appropriate SVM to generate the classifier. Such a speed up has the potential to allow applications which currently are not possible by the available methods. A concrete example is the detection of malware and computer viruses in (possibly encrypted) internet traffic. To prevent hijacking attacks, it would be most useful to be able to detect and eliminate such unwanted internet traffic already at routers or switches, before malware can attack the computer of an end user. However, running an SVM on routers or switches with low cost processors to scan internet traffic in real-time is currently not possible due to the high dimension D of the data. Here the approach proposed in this article may lead to a significant improvement.

If D is so large that minimization in dimension D is numerically out of reach, but not too large to calculate the dot product of D-dimensional vectors sufficiently fast, and minimization in dimension N is

numerically tractable, an appropriate way to generate an SVM is to use the dual optimization problem and the dual form of the classifier. However, if D is so large that even the dot product of fully occupied D-dimensional vectors can not be calculated sufficiently fast, then it is necessary to generate an SVM with sufficiently many zero components of the classifying vector, as this allows to reduce the D multiplications needed for classification to a sufficiently small number. Sparsity of the classifying vector can be obtained by using a 1-norm SVM, where the usual 2-norm of the classifying vector in the objective function of an SVM is replaced by a 1-norm, i.e. a L^1-regularization instead of an L^2-regularization of the penalty function is used or – in other words – a least absolute shrinkage and selection operator (LASSO) is constructed. In [8] it is shown that in case of non-linearly separable data for arbitrary L^p-regularization, $p \geq 1$, there are at most $D+1$-support vectors, but this result does not help much for $D >> N$. In contrast, the discussion of sparsity obtained by L^1-regularization in [9, 18.4] mentions with reference to [10] that for a 1-norm SVM there are at most N support vectors. Moreover, [10] discusses properties of the solution path obtained by varying the regularization parameter, and [11] provides an accompanying numerical study. Yet, for a 1-norm SVM the dual problem is different from that of a 2-norm SVM, particularly it is not trivial to obtain the classifying vector from the solution of the dual problem, and the aim of this article is to properly work out, how a sparse classifier can be calculated from the solution of the dual problem of a 1-norm SVM. A different approach to obtain sparsity is suggested in [12], instead of using L^1-regularization there the constraints are modified appropriately.

1.1 Outline

In section 2 we review 2-norm SVMs. In the main section 3 we discuss 1-norm SVMs, prove an a-posteriori-sparsity Lemma 2 and show in Theorem 3 how to obtain a sparse solution of the primal problem from the solution of the dual problem. Finally, we formulate the sparse SVM method based on these results.

2 Review of 2-norm SVMs

A *support vector machine* (SVM) for the binary classification of data points x is given by an affine function $f(x) := \omega^T x + \beta$: If $f(x) > 0$, then x is estimated to be of class $+1$, else of class -1. To generate an SVM from training data $x_i \in \mathbb{R}^D$ with binary classification $y_i = \pm 1$, $i = 1, \ldots, N$, in the case where D and N are not too large, usually the minimization problem

$$\frac{1}{2}\sum_{j=1}^{D}\omega_j^2 + \gamma\sum_{i=1}^{N}|\xi_i| = \min_{\omega,\beta,\xi}! \quad \text{subject to} \quad (\omega^T x_i + \beta)y_i \geq 1 - \xi_i \text{ for } i = 1, \ldots, N$$

with a regularization parameter $\gamma > 0$ is solved e.g. by a primal-dual algorithm.

For a soft-margin SVM a finite regularization parameter $\gamma < +\infty$ is chosen. In this case, there may be training data points x_i with $f(x_i) < 1$ which belong to class $+1$ (resp. x_i with $f(x_i) > -1$ which belong to class -1), such data points x_i are merely penalized via the slacks $\xi_i > 0$. For too small $\gamma > 0$, the generated SVM may have too large slacks, i.e. a too high number of misclassifications ($f(x_i) < 0$ or equivalently $\xi_i > 1$ for many x_i of class $+1$) or non-sharp classifications ($0 \leq f(x_i) < 1$ or equivalently $0 < \xi_i \leq 1$ for many x_i of class $+1$), or some extreme misclassifications ($f(x_i) << 0$ or equivalently $\xi_i >> 1$ for few x_i of class $+1$) of training data are accepted. Note that due to our choice of the 1-norm as regularization term all these different classification errors are penalized in the same way. If instead $\frac{\gamma}{p}\sum_{i=1}^{N}|\xi_i|^p$ with $p > 1$ is chosen as regularization term, then non-sharp classifications are not so strongly penalized as misclassifications. Moreover, if the training data is far away from being linearly separable, then there may be no (ω, β) which makes the slacks small. As mentioned in the introduction, in this case the data points should be embedded into a higher dimensional space to make them (nearly) linearly separable, and the kernel trick should be used to avoid calculations in this higher dimensional space. Yet, for certain training data characterized in [8], the minimizer (ω, β, ξ) may be degenerate, i.e. $(\omega, \beta) = (\mathbf{0}, 0)$ may hold so that the obtained classifier is not useful.

For a hard-margin SVM the regularization parameter is set to $\gamma = +\infty$, i.e. no slacks are allowed and $f(x_i) \geq 1$ is required for every training data point x_i of class $+1$ (resp. $f(x_i) \leq -1$ for every x_i of class

-1). An advantage of hard-margin SVMs is that minimization has to be performed merely in the variables (ω, β) of dimension $D+1$ (and not in the variables (ω, β, ξ) of dimension $D+1+N$). Particularly, to solve the minimization problem may be numerically tractable for low dimensions D even if N is very large. However, for hard-margin SVMs linearly separable training data are needed, else the minimization problem is not admissible.

A training data point x_i is called a *support vector* of the generated SVM, if the i-th constraint is active, i.e. $(\omega^T x_i + \beta) y_i = 1 - \xi_i$ or equivalently $f(x_i) \leq 1$ for x_i of class $+1$ (resp. $f(x_i) \geq -1$ for x_i of class -1). As we shall see below, if the SVM is generated by the minimization problem above, then the classifying vector ω is a linear combination of the support vectors, i.e. only the support vectors but no other training data points influence the classifier.

In matrix-vector notation, the above minimization problem reads as

$$\frac{1}{2}\|\omega\|_2^2 + \gamma\|\xi\|_1 = \min_{\omega,\beta,\xi}! \quad \text{subject to} \quad Y^T X^T \omega + \beta Y^T \mathbf{1} \geq \mathbf{1} - \xi \tag{1}$$

with the two-norm $\|\omega\|_2$ of $\omega \in \mathbb{R}^D$, $\beta \in \mathbb{R}$, the one-norm $\|\xi\|_1$ of $\xi \in \mathbb{R}^N$, the one vector $\mathbf{1} \in \mathbb{R}^N$, the diagonal classification matrix $Y = \operatorname{diag}(y_1, \ldots, y_N) \in \mathbb{R}^{N \times N}$ and the training matrix $X = (x_1, \ldots, x_N) \in \mathbb{R}^{D \times N}$. Due to the prominent role played by the 2-norm of ω in (1), we call an SVM generated in this way a 2-norm SVM. Mathematically, without any additional difficulties an arbitrary orthogonal matrix Y can be used instead of a diagonal matrix. An application for the non-diagonal case may be an uncertain binary classification of the training data, where Y model dependencies between the classification of different data points x_i. As

$$L(\omega, \beta, \xi, \lambda) = \frac{1}{2}\|\omega\|_2^2 + \gamma\|\xi\|_1 + \left(\mathbf{1} - \xi - Y^T X^T \omega - \beta Y^T \mathbf{1}\right)^T \lambda$$

is the Lagrangian of (1), the KKT-conditions for a minimizer (ω, β, ξ) read as

$$\begin{aligned}
\omega - XY\lambda = 0 \qquad & \left(\Leftrightarrow \frac{\partial L}{\partial \omega} = 0\right) \\
\mathbf{1}^T Y \lambda = 0 \qquad & \left(\Leftrightarrow \frac{\partial L}{\partial \beta} = 0\right) \\
\gamma \operatorname{sgn}(\xi) \ni \lambda \qquad & \left(\Leftrightarrow \frac{\partial L}{\partial \xi} = 0\right) \\
\left(\mathbf{1} - \xi - Y^T X^T \omega - \beta Y^T \mathbf{1}\right)^T \lambda = 0 \qquad & \text{(complementarity)} \\
Y^T X^T \omega + \beta Y^T \mathbf{1} \geq \mathbf{1} - \xi \qquad & \text{(constraints)} \\
\lambda \geq \mathbf{0} \qquad & \text{(sign condition)}
\end{aligned}$$

These conditions allow the following conclusions:

Remark 1

1. *$\xi \geq \mathbf{0}$, as $\gamma \operatorname{sgn}(\xi) \ni \lambda$ and $\lambda \geq \mathbf{0}$ imply non-negativity of ξ (of course, we want that the slacks ξ are non-negative and already assumed this implicitly, but we did nowhere require this explicitly via a constraint, yet – as we have seen now – non-negativity automatically holds for a minimizer).*
2. *$\omega = XY\lambda$ is in the case of a diagonal Y a linear combination of the support vectors (sometimes this fact is called representer theorem), as $\lambda_i \neq 0$ only for support vectors x_i.*

If the dimension D is so large that a numerical solution of the minimization problem (1) is out of reach, while minimization in dimension N is numerically still tractable, then it is advantegeous to use the dual formulation. The Lagrangian dual problem of (1) reads as

$$\frac{1}{2}\|XY\lambda\|_2^2 - \mathbf{1}^T\lambda = \min_{\lambda}! \quad \text{subject to} \quad \mathbf{1}^T Y \lambda = 0 \text{ and } \mathbf{0} \leq \lambda \leq \gamma\mathbf{1}\,, \tag{2}$$

and the classifier is given in its dual form by $f(x) := (XY\lambda)^T x + \beta$. As $\lambda_i \neq 0$ only for support vectors x_i, the complexity to calculate $\omega = XY\lambda$ scales linearly with the number of support vectors, avoiding in

the case of a diagonal Y multiplication of those x_i by $y_i\lambda_i$ where $\lambda_i = 0$ is known. However, while λ is sparse, ω usually is *fully occupied* and *not sparse*. Further, when solving the minimization problem (2), the calculation of D-dimensional vectors $XY\lambda$ should be avoided: Due to $\|XY\lambda\|_2^2 = \lambda^T((XY)^T(XY))\lambda$, where $(XY)^T(XY)$ is an $(N \times N)$-matrix, only the products $y_iy_j \cdot x_i^T x_j$ of the training data points have to be precalculated to avoid a calcuation with D-dimensional vectors.

3 1-norm SVMs

In this section we are interested in the case, where the dimension D is so large that the dot product can not be calculated sufficiently fast for the desired application. In this case it would be advantegeous if not only λ but also $\omega = XY\lambda$ would be sparse. As we show below, in contrast to 2-norm SVMs this is the case for 1-norm SVMs. A 1-norm SVM is generated from training data $x_i \in \mathbb{R}^D$ with binary classification $y_i = \pm 1$, $i = 1, \ldots, N$, by the minimization problem

$$\sum_{j=1}^{d} |\omega_j| + \frac{\gamma}{p} \sum_{i=1}^{N} |\xi_i|^p = \min! \quad \text{subject to} \quad (\omega^T x_i + \beta)y_i \geq 1 - \xi_i \,.$$

In fact, using the 1-norm instead of the 2-norm for the normal vector ω of the separating hyperplane enforces that for a minimizer (ω, β, ξ) the vector ω is sparse, i.e. many components of ω are zero. Thus, despite the very high dimension D of ω it is still cheap to form the dot product with ω. For the training of the SVM not the primal but the dual problem should be used, as there an optimization in $N << D$ variables has to be performed.

For $p = 2$ the problem reads in matrix-vector notation as

$$\|\omega\|_1 + \frac{\gamma}{2}\|\xi\|_2^2 = \min! \quad \text{subject to} \quad Y^T X^T \omega + \beta Y^T \mathbf{1} \geq \mathbf{1} - \xi \tag{3}$$

with $\omega \in \mathbb{R}^d$, $\beta \in \mathbb{R}$, $\xi \in \mathbb{R}^N$, the one vector $\mathbf{1} \in \mathbb{R}^N$, the diagonal classification matrix $Y = \operatorname{diag}(y_1, \ldots, y_N) \in \mathbb{R}^{N \times N}$ and the training matrix $X = (x_1, \ldots, x_N) \in \mathbb{R}^{d \times N}$. The Lagrangian of (3) is

$$L(\omega, \beta, \xi, \lambda) = \|\omega\|_1 + \frac{\gamma}{2}\|\xi\|_2^2 + \left(\mathbf{1} - \xi - Y^T X^T \omega - \beta Y^T \mathbf{1}\right)^T \lambda \,,$$

thus at a minimizer (ω, β, ξ) of (3) there exist Lagrangian multipliers $\lambda \in \mathbb{R}^N$ such that the KKT conditions

$$\begin{aligned}
\operatorname{sgn}(\omega) - XY\lambda \ni 0 \qquad & \left(\Leftrightarrow \frac{\partial L}{\partial \omega} = 0\right) \\
\mathbf{1}^T Y \lambda = 0 \qquad & \left(\Leftrightarrow \frac{\partial L}{\partial \beta} = 0\right) \\
\gamma\xi - \lambda = 0 \qquad & \left(\Leftrightarrow \frac{\partial L}{\partial \xi} = 0\right) \\
\left(\mathbf{1} - \xi - Y^T X^T \omega - \beta Y^T \mathbf{1}\right)^T \lambda = 0 \qquad & \text{(complementarity)} \\
Y^T X^T \omega + \beta Y^T \mathbf{1} \geq \mathbf{1} - \xi \quad & \text{(constraints)} \\
\lambda \geq 0 \qquad & \text{(sign condition)}
\end{aligned}$$

are satisfied. Hence, due to $\gamma\xi = \lambda$ we can eliminate the Lagrangian multiplier and obtain

$$\begin{aligned}
\gamma XY\xi &\in \operatorname{sgn}(\omega) \\
\mathbf{1}^T Y \xi &= 0 \\
\left(\mathbf{1} - \xi - Y^T X^T \omega - \beta Y^T \mathbf{1}\right)^T \xi &= 0 \\
Y^T X^T \omega + \beta Y^T \mathbf{1} &\geq \mathbf{1} - \xi \\
\xi &\geq 0 \,.
\end{aligned}$$

The first inclusion already indicates sparsity of ω: If $\gamma(XY\xi)_j$ lies in the open interval $(-1, 1)$ for an index j, the inclusion implies $\omega_j = 0$. This is a kind of a-posteriori-sparsity result:

Lemma 2 *If the support vectors x_i weighted by γ and the signed slacks have a j-th component between (but not equal to) -1 and $+1$, then $\omega_j = 0$.*

Particularly, if $\gamma > 0$ is small, then $\omega_j = 0$ for many j, but due to smallness of $\gamma > 0$ the generated SVM may have too large slacks. Yet, also if $\gamma > 0$ is not small, but the slacks are small, then $\omega_j = 0$ for many j. Moreover, if the support vectors weighted by the signed slacks nearly cancel out, then $\omega_j = 0$ for many j (note that due to $\mathbf{1}^T Y\xi = 0$ the signed stacks alone automatically cancel out), and this is exactly what Lemma 2 says. To generate a 1-norm SVM for $D >> N$, the Lagrangian dual problem of (3) should be solved, which reads as

$$\frac{1}{2\gamma}\|\lambda\|_2^2 - \mathbf{1}^T\lambda = \min_{\lambda}! \text{ subject to } \mathbf{1}^T Y\lambda = 0\,,\ -\mathbf{1} \le XY\lambda \le \mathbf{1} \text{ and } \lambda \ge 0\,. \tag{4}$$

However, the classifier can not be directly calculated from λ, as $XY\lambda$ is merely the sign of ω, but the absolute value of each component is unknown. Yet $\xi = \frac{1}{\gamma}\lambda$, thus (ω, β) can be determined by solving the linear equation $(Y^T X^T\omega + Y^T\mathbf{1}\beta)_I = (\mathbf{1} - \xi)_I$, where $I := \{i \,|\, \lambda_i > 0\}$ and ω automatically is zero outside the index set $J := \{j \,|\, |(XY\lambda)_j| \ge 1\}$. These $|I| \le N$ linear equations for $|J| + 1$ variables are the only conditions on (ω_J, β), thus all solutions of these linear equations are minimizers.

Theorem 3 *Let $\lambda \in \mathbb{R}^N$ be a solution of the dual problem (4), let $I := \{i \in \{1, \ldots, N\} \,|\, \lambda_i > 0\}$ and let $J := \{j \in \{1, \ldots, D\} \,|\, |(XY\lambda)_j| \ge 1\}$. Then the (in general non-unique) minimizer (ω, β, ξ) of (3) satisfies $\xi = \frac{1}{\gamma}\lambda$ and $\omega_j = 0$ for $j \notin J$, and the remaining ω_j for $j \in J$ as well as β solve the linear equations $(Y^T X^T\omega + \beta Y^T\mathbf{1})_i = 1 - \frac{1}{\gamma}\lambda_i$, $i \in I$.*

Due to Theorem 3 a posteriori not only the number $|I|$ of support vectors can be read off from the solution λ of the dual problem, but also the least number $D - |J|$ of components where ω is zero, and the sparse classifier can be obtained from λ just by solving a system of linear equations of dimension $|J|$.

4 Conclusion

Together with Lemma 2, which guarantees that for sufficiently small $\gamma > 0$ also the index set J is sufficiently small (for a more precise discussion of the dependence of the solution path on γ see [10]), we obtain from theorem 3 the following SVM algorithm to calculate a sparse classifying vector ω in the case $D >> N$:

1. Choose $\gamma > 0$ sufficiently small (or consider the whole solution path for $\gamma \in (0, \infty)$).
2. Solve the dual problem (4) to obtain $\lambda \in \mathbb{R}^N$.
3. Let $I := \{i \in \{1, \ldots, N\} \,|\, \lambda_i > 0\}$ be the indices of support vectors.
4. Let $J := \{j \in \{1, \ldots, D\} \,|\, |(XY\lambda)_j| \ge 1\}$ be the indices where not a priori $\omega_j = 0$ for $j \in J$.
5. Put $\omega_j := 0$ for every $j \notin J$.
6. Solve $(Y^T X^T\omega + \beta Y^T\mathbf{1})_i = 1 - \frac{1}{\gamma}\lambda_i$, $i \in I$, for the remaining ω_j, $j \in J$, and β.

This algorithm combines two advantages: On the one hand, a linear SVM classifier is constructed, which because of its sparsity may even be used if the dimension D of the feature space is very large. On the other hand, despite the high dimension D of the feature space, the training of the SVM is possible in acceptable time for $N << D$, because merely an optimization problem in dimension N has to be solved.

These two properties may for example allow to scan internet traffic for malware in real-time at routers or switches, because on the one hand the classifier is sufficiently fast due to sparsity of ω, and on the other hand new information about malware can be incorporated into the classifier over night by a new training of the SVM. We plan a numerical study of this algorithm and general sparse SVMs in the project KompDataSci and forthcoming papers.

Let us again point out that the smaller $\gamma > 0$ is chosen, the more components of ω are zero, however, the price for this advantage are larger classification mistakes. At present there seems to be no way to calculate $\gamma > 0$ a priori from training data so that e.g. a certain least number of components of ω is zero and at the same time the classification error is acceptable. It would be an improvement to have an automatic method for choosing the regularization parameter so that such an objective is reached, but at present $\gamma > 0$ has to be chosen carefully by hand, e.g. by considering the whole path of solutions when $\gamma \in (0, \infty)$ is varied.

Acknowledgments. The author thanks the German ministry of education and research (BMBF) for the possibility of support of his research in the programs *Machine Learning* (KompDataSci) and *Mathematics for Innovations* (FuzzyPV).

References

1. V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.
2. P. Foremski, "On different ways to classify internet traffic: a short review of selected publications," *Theor. Appl. Inf.*, vol. 25, pp. 119–136, 2013.
3. H. Kim, K. Clay, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practices," in *Proceedings of the 2008 ACM CoNEXT conference*. ACM, 2008, pp. 11–14.
4. P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Network Mgmt.*, pp. 1–24, 2014.
5. A. Khakpour and A. Liu, "An information-theoretical approach to high-speed flow nature identification," *IEEE/ACM Transactions on Networking*, vol. 21, pp. 1076–1089, 2013.
6. B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.
7. J. Fehr, K. Arreola, and H. Burkhardt, "Fast support vector machine classification of very large datasets," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, 2006.
8. E. Carrizosa, "Support vector machines and distance minimization," in *Data mining and mathematical programming / Panos M. Pardalos, Pierre Hansen, editors. CRM proceedings & lecture notes, ISSN 1065-8580; Volume 45*. AMS, 2008, pp. 1–13.
9. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009, 2013.
10. S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *The Annals of Statistics*, vol. 35, no. 3, pp. 1012–1030, 2007.
11. J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, 2004, pp. 49–63.
12. V. Gómez-Verdejo, M. M. Ramón, J. Arenas-García, and H. Molina-Bulla, "Support vector machines with constraints for sparsity in the primal parameters," *IEEE Transactions on neural networks*, vol. 22, no. 8, pp. 1269–1283, 2011.

