Classification with Margin Constraints: A Unification with Applications to Optimization

Pooria Joulani[1] **András György**[2] **Csaba Szepesvári**[1]
[1]Department of Computing Science, University of Alberta, Edmonton, AB, Canada
{pooria,szepesva}@ualberta.ca
[2]Department of Electrical and Electronic Engineering, Imperial College London, UK
a.gyorgy@imperial.ac.uk

Abstract

This paper introduces Classification with Margin Constraints (CMC), a simple generalization of cost-sensitive classification that unifies several learning settings. In particular, we show that a CMC classifier can be used, out of the box, to solve regression, quantile estimation, and several anomaly detection formulations. On the one hand, our reductions to CMC are at the loss level: the optimization problem to solve under the equivalent CMC setting is exactly the same as the optimization problem under the original (e.g. regression) setting. On the other hand, due to the close relationship between CMC and standard binary classification, the ideas proposed for efficient optimization in binary classification naturally extend to CMC. As such, any improvement in CMC optimization immediately transfers to the domains reduced to CMC, without the need for new derivations or programs. To our knowledge, this unified view has been overlooked by the existing practice in the literature, where an optimization technique (such as SMO or PEGASOS) is first developed for binary classification and then extended to other problem domains on a case-by-case basis. We demonstrate the flexibility of CMC by reducing two recent anomaly detection and quantile learning methods to CMC.

1 Introduction

Modern machine learning algorithms are based on optimization, typically minimizing a loss function over the given data set, where the loss function evaluates the quality of the hypothesis being learned. While the choice of the loss function naturally depends on the learning problem at hand (e.g., classification, regression, density estimation, outlier detection, etc.) and the structure of the hypothesis being learned, an important factor in choosing a loss function is the availability of an efficient algorithm to solve the resulting optimization problem. As such, development of efficient optimization methods has been a central problem in machine learning.

Due to the importance of binary classification, numerous papers have studied efficient optimization procedures for classification, and in particular for Support Vector Machines (SVMs) [1–40]. See also [41] and the references therein. As such, a large body of classifier training techniques currently exists. To obtain efficient optimization methods for other learning problems, such as regression and anomaly detection, an existing practice in the literature has been to extend optimization algorithms for the binary classification to those other settings on a case-by-case basis. For examples, several papers propose to extend binary SVM optimization algorithms such as Sequential Minimal Optimization (SMO) [10] or PEGASOS [42] to SVMs for regression and outlier detection [16,27,40,43–51].

In this paper, we argue that in contrast to the existing practice, optimization techniques could be developed more efficiently by observing the relationship between the loss functions for different learning settings. In particular, we introduce the problem of Classification with Margin Constraints

(CMC), a slight generalization of the binary classification problem that sheds light on the relationship between losses and algorithms for seemingly different learning settings. Specifically, we show that diverse learning problems (including recent formulations for semi-supervised anomaly detection [52] or hierarchical quantile estimation [53]) reduce to CMC without changing the underlying optimization problem. At the same time, due to the close relationship between CMC and binary classification, the optimization techniques designed for binary classification naturally work for CMC. Thus, this paper shows that there is a sufficiently inclusive model for unified optimization in all of these settings that removes the need for superficial extensions and redundant implementations.

The CMC problem is a generalization of example-dependent cost-sensitive classification, and in particular classification with uneven margins. Learning with example-dependent loss functions [44, 54–63], and in particular SVMs with class-dependent margins [54, 56, 57, 59, 61] and inter-class uneven margins [56] have been studied before. However, unlike previous work, the CMC problem allows *negative* margins, which facilitates reductions from other learning settings.

The rest of this paper is organized as follows: Section 2 provides the formal definition of the CMC problem. In Section 3, we show that regression using deviation-based losses reduces to CMC with margin-based losses, and provide several examples. Section 4 considers CMC with the hinge loss and reduces several SVM formulations, including the ν-SVM family of algorithms and SVMs for unsupervised and semi-supervised anomaly detection, to the CMC problem with the hinge loss. Section 5 concludes the paper and points to directions for future research.

2 Classification with margin constraints

As in binary classification, our goal in the CMC setting is to find a decision function that is positive on positively-labeled examples and negative otherwise. However, in the CMC problem we specify a lower (upper) bound on how positive (negative) the decision function has to be, i.e., we specify *margin sensitivities*. Formally, we are given a set $\mathcal{X}$, a class of functions $\mathcal{F} \subset \{f | f : \mathcal{X} \mapsto \mathbb{R}\}$, and a data set $(x_1, y_1, \gamma_1), (x_2, y_2, \gamma_2), \dots, (x_n, y_n, \gamma_n)$, where each example in the data set consists of a data point $x_i \in \mathcal{X}$, a label $y_i \in \{+1, -1\}$, and a margin sensitivity $\gamma_i = (\gamma_{i,0}, \gamma_{i,1}, \dots, \gamma_{i,d}) \in \mathbb{R}^{1+d}, d \geq 0$. Our goal is to find a function $f \in \mathcal{F}$ and a vector $\rho = (1, \rho_1, \rho_2, \dots, \rho_d) \in \mathbb{R}^{d+1}$ such that for all $i = 1, 2, \dots, n$, we have $f(x_i) \geq \rho^\top \gamma_i$ when $y_i = +1$ and $f(x_i) \leq -\rho^\top \gamma_i$ when $y_i = -1$, or equivalently $y_i f(x_i) \geq \rho^\top \gamma_i$. More generally, we evaluate the quality of f and ρ by a "loss function" $\ell : \mathcal{X} \times \{+1, -1\} \times \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$. Assuming that the data points are i.i.d. from an unknown distribution $P_{X,Y,\Gamma}$ over $\mathcal{X} \times \{+1, -1\} \times \mathbb{R}^{d+1}$, we want to find f and ρ that minimize the "classification risk", defined as

$$\mathcal{R}_\ell^{\text{cls}}\left(f, \rho, P_{X,Y,\Gamma}\right) := \int \ell(x, y, \rho^\top \gamma, f(x)) dP_{X,Y,\Gamma} = \mathbb{E}\left[\ell\left(X, Y, \rho^\top \Gamma, f\left(X\right)\right)\right], \tag{1}$$

where (X, Y, Γ) are random variables jointly distributed with $P_{X,Y,\Gamma}$ [1]. We denote the empirical distribution underlying our data set by $\hat{P}_n$, and the associated empirical risk by

$$\hat{\mathcal{R}}_{\ell,n}^{\text{cls}}(f, \rho) := \mathcal{R}_\ell^{\text{cls}}\left(f, \rho, \hat{P}_n\right) = \tfrac{1}{n} \sum_{i=1}^{n} \ell\left(x_i, y_i, \rho^\top \gamma_i, f(x_i)\right).$$

3 Regression as CMC

In the regression problem, we are given a loss function $\ell : \mathcal{X} \times \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$, and we are interested in finding a function $f : \mathcal{X} \mapsto \mathbb{R}$ that minimizes the regression risk

$$\mathcal{R}_\ell^{\text{reg}}\left(f, P_{X,Z}\right) := \int \ell(x, z, f(x)) dP_{X,Z} = \mathbb{E}\left[\ell(X, Z, f(X))\right], \tag{2}$$

where $P_{X,Z}$ is the (unknown) distribution over $\mathcal{X} \times \mathbb{R}$ generating the data, and (X, Z) $P_{X,Z}$. In this section we show that when ℓ is a *deviation-based loss* (defined below), we can reduces the regression problem above to a CMC problem with $d = 0$ (i.e., real-valued γ_i). We start by two definitions.

[1] In the following, we assume that every distribution we discuss is defined on a suitable sigma-field of the corresponding sample space, and that all the functions we are considering are measurable.

Classification Loss	$\phi(x,y,m)$	**Shift constant**	$\psi(x,m)$	**Regression Loss**
Hinge loss (ϕ_{hinge})	$\max\{1-m,0\}$	$\alpha^+=-\epsilon+1$, $\alpha^-=-\epsilon-1$	$\max\{\|m\|-\epsilon,0\}$	ϵ-insensitive loss
Weighted hinge loss	$\begin{cases}\tau\phi_{\text{hinge}} & y=1\\(1-\tau)\phi_{\text{hinge}} & \text{o.w.}\end{cases}$	$\alpha^+=1$, $\alpha^-=-1$	$\begin{cases}-\tau m & m\le 0\\(1-\tau)m & \text{o.w.}\end{cases}$	τ-quantile regression loss
Square loss	$(1-m)^2$	$\alpha^+=1$, $\alpha^-=-1$	m^2	Square loss
Truncated square loss	$(\max\{1-m,0\})^2$	$\alpha^+=1$, $\alpha^-=-1$	m^2	Square loss
Logistic loss	$\log\left(1+e^{-m}\right)$	$\alpha^+=\alpha^-=0$	$\log\left(1+\frac{e^{-m}+e^{m}}{2}\right)$	Log-exp regression loss

Table 1: Examples of common classification losses and their corresponding regression loss. We assume $\tau\in(0,1)$ for quantile regression.

Definition 1 (Deviation-based loss). *A regression loss $\ell:\mathcal{X}\times\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R}$ is a* deviation-based loss *if there exists a function $\psi:\mathcal{X}\times\mathbb{R}\mapsto\mathbb{R}$ such that $\ell(x,z,t)=\psi(x,t-z)$ for all $x\in\mathcal{X},z,t\in\mathbb{R}$.*

Definition 2 (Margin-based loss). *A loss function $\ell:\mathcal{X}\times\{+1,-1\}\times\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R}$ for CMC classification is a* margin-based loss *if there exists a function $\phi:\mathcal{X}\times\{+1,-1\}\times\mathbb{R}\mapsto\mathbb{R}$ such that $\ell(x,y,s,t)=\phi(x,y,yt-s)$ for all $x\in\mathcal{X},y\in\{+1,-1\},t,s\in\mathbb{R}$.*

In the following, we identify deviation-based or margin-based losses with their corresponding functions ψ and ϕ, e.g., we allow writing $\mathcal{R}^{\text{cls}}_{\phi}$ and $\mathcal{R}^{\text{reg}}_{\psi}$.

Next, we show that a CMC classifier for a margin-based loss ϕ can solve regression problems with the deviation-based loss $\psi(x,m)=\frac{1}{2}\phi(x,1,m+\alpha^+)+\frac{1}{2}\phi(x,-1,-m-\alpha^-)$, where α^+ and α^- are any pair of constants. To see that, consider a regression sample $(x_1,z_1),(x_2,z_2),\ldots,(x_n,z_n)$. Make a copy of the data points x_i, and assign a positive label to all examples in the first copy and a negative label to all examples in the second copy. More precisely, for $i=1,2,\ldots,n$, let $x'_i=x'_{i+n}=x_i$, $y'_i=1$ and $y'_{i+n}=-1$. Set the margin sensitivities to $\gamma'_i=z_i-\alpha^+$ and $\gamma_{i+n}=-z_i+\alpha^-, i=1,2,\ldots,n$. Then, $(x'_1,y'_1,\gamma'_1),\ldots,(x'_{2n},y'_{2n},z'_{2n})$ is a CMC data set, and for every f, the empirical risk in ψ of f over the regression data set is equal to the empirical risk in ϕ of f on the CMC data sets. This is not limited to the empirical risk, as the next Theorem shows[2].

Theorem 1. *Consider a distribution $P_{X,Z}$ over $\mathcal{X}\times\mathbb{R}$ and a deviation-based loss ψ. Suppose that $\psi(x,m)=\frac{1}{2}\psi^+(x,m+\alpha^+)+\frac{1}{2}\psi^-(x,-m-\alpha^-)$ for some functions ψ^+ and ψ^- and constants α^+ and α^-. Define the margin-based classification loss ϕ as:*

$$\phi(x,y,m)=\begin{cases}\psi^+(x,m), & \textit{if } y=1;\\ \psi^-(x,m), & \textit{otherwise.}\end{cases} \tag{3}$$

Then there exist a distribution $P_{X,Y,\Gamma}$ over $\mathcal{X}\times\{+1,-1\}\times\mathbb{R}$ such that for all functions f,

$$\mathcal{R}^{\text{cls}}_{\phi}\left(f,P_{X,Y,\Gamma}\right)=\mathcal{R}^{\text{reg}}_{\psi}\left(f,P_{X,Z}\right). \tag{4}$$

Table 1 shows several regression losses and their corresponding CMC loss. This shows, for example, that a CMC classifier using the weighted hinge loss can also solve SVM regression and quantile estimation problems *out of the box*, without the need for new implementations[3]. Similarly, a classifier using the truncated square loss can also do least-square regression. Interestingly, we also derive a form of regression using a robust loss (the "log-exp" loss above) that reduces to logistic regression.

[2] Even more generally, ψ could be the sum of any number of scaled, mirrored and shifted copies of ϕ.

[3] Note that "copying" the data does not necessarily reduce efficiency. For example, the extension of SMO to regression performs the exact same steps as when SMO for classification is applied with this reduction.

SVM Formulation	d	μ	c_i	γ_i
1-Class SVM for anomaly detection [50]	$d=1$	$\mu=(0,\nu)^\top$	$c_i=1$	$\gamma_i=(-1,1)^\top$
ν-SVM Classifier [65]	$d=2$	$\mu=(0,\nu,0)^\top$	$c_i=1$	$\gamma_i=(-1,1,-y_i)^\top$
Semi-supervised anomaly detection (SSAD) [52]	$d=2$	$\mu=(0,1,\kappa)^\top$	$c_i=\begin{cases}\eta_u & i\in\mathcal{U}\\ \eta_l & \text{o.w.}\end{cases}$	$\gamma_i=\begin{cases}(-1,1,0) & i\in\mathcal{U}\\ (-1,y_i,1) & \text{o.w.}\end{cases}$
Hierarchical Quantil Estimation (q-OCSVM) [53]	$d=q+1$	$\mu=(0,\frac{1}{q},\dots,\frac{1}{q})^\top$	$c_{i+nj}=\frac{1}{q\nu_j}$	$\gamma_{i+nj}=(-1,0,\dots,1,0,\dots)$

Table 2: Examples of SVM formulations reduced to CMC with the hinge loss (problem (5)). In case of SSAD, $\mathcal{U}$ indicates the set of unlabelled examples. For q-OCSVM, we create q copies of the data, and index the j-th copy of x_i as x'_{i+nj}, where $0\le j<q$ and $1\le i\le n$. In that case, γ_{i+nj} has a 1 at index $j+1$. In the corresponding rows, q, ν_j, ν, κ, η_u and η_l are hyper-parameters.

4 CMC with the hinge loss and Support Vector Machines

In this section, we focus on CMC with the hinge loss, reducing several existing as well as new SVM formulations to CMC [4]. Consider the example-dependent cost-sensitive hinge loss for CMC,

$$\ell(x,y,\rho^\top\gamma,f(x))=\phi(x,y,yf(x)-\rho^\top\gamma)=c(x)\max\{1+\rho^\top\gamma-yf(x),0\},$$

where $c(x)\ge 0$ gives the scaling of the loss for data point x. Let $c_i:=c(x_i)$ denote the scale of the loss for the i-th example in the data set. Different choices of γ_i and c_i result in different (existing and new) SVM formulations. For example, suppose that $\mathcal{F}$ is the set of linear functions $f(x)=w^\top x$, where the ℓ_2-norm of w is regularized and the vector ρ is supposed to maximizes the margin sensitivity [5]. The corresponding CMC SVM optimization problem is

$$\min_{w,\rho}\frac{1}{2}w^\top w-\mu^\top\rho+\frac{1}{n}\sum_{i=1}^{n}c_i\max\{1+\rho^\top\gamma-yf(x),0\},\tag{5}$$

where $\mu\in\mathbb{R}^d$ can be thought of as a prior guess of the average margin sensitivity γ.

Note that (5) is not harder to solve than the standard ν-SVM classifier of [65], but several ν-SVM formulations reduce to (5). Table 2 summarizes the choices of μ, γ_i and c_i that result in some of these reductions[6]. The bottom two rows show the reductions for two recently proposed methods: a semi-supervised anomaly detection algorithm and a hierarchical quantile estimation method. Using the reduction to CMC, efficient training will be immediately available for these (and possibly other) new methods, simply by using an existing CMC classifier.

5 Conclusion and future work

We introduced the CMC problem, and showed that although CMC is only slightly different from binary classification, several other learning settings reduce to CMC. The reductions result in exactly the same optimization problems as those solved by the original method, but the CMC view enables us to uniformly apply efficient optimizaiton ideas, especially those developed for binary classification.

The CMC problem is also interesting from a theoretical point of view. Given that the reductions are at the risk level, an interesting question is whether these reductions could facilitate a unified analysis of learning under these settings. Another interesting question is whether similar reductions exist for other, more diverse learning settings such as clustering or for multi-class problems.

[4] In principle, we could do similar manipulations for other margin-based losses. We have chosen the hinge loss since the rich body of SVM formulations allows us to better demonstrate the power of the CMC framework.

[5] Note that the reduction depends only on the loss, not the regularization of f or penalization of ρ; the reduction works as long as the CMC classifier applies the same restrictions on f and ρ as the reduced problem. For example, all results would remain true if we had instead used the "Extended ν-SVM" formulation [64].

[6] Other variants of ν-SVM, e.g., ν-SVM for regression and generalizations to parameteric-sensitivity models [65], as well as C-SVM formulations, are also special cases of (5), but are excluded for lack of space.

Acknowledgements

This work was supported by the Alberta Innovates Technology Futures through the Alberta Ingenuity Centre for Machine Learning and NSERC.

References

[1] S Sathiya Keerthi, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and Krishna RK Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *Neural Networks, IEEE Transactions on*, 11(1):124–136, 2000.

[2] Michael Vogt and Vojislav Kecman. Active-set methods for support vector machines. In *Support vector machines: Theory and applications*, pages 133–158. Springer, 2005.

[3] Christopher Sentelle, Georgios C Anagnostopoulos, and Michael Georgiopoulos. An efficient active set method for svm training without singular inner problems. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 2875–2882. IEEE, 2009.

[4] Katya Scheinberg. An efficient implementation of an active set method for svms. *The Journal of Machine Learning Research*, 7:2237–2257, 2006.

[5] Edgar Osuna, Robert Freund, and Federico Girosi. An improved training algorithm for support vector machines. In *Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*, pages 276–285. IEEE, 1997.

[6] Ivor W Tsang, James T Kwok, and Pak-Ming Cheung. Core vector machines: Fast svm training on very large data sets. In *Journal of Machine Learning Research*, pages 363–392, 2005.

[7] Christopher Sentelle, Georgios C Anagnostopoulos, and Michael Georgiopoulos. Efficient revised simplex method for svm training. *Neural Networks, IEEE Transactions on*, 22(10):1650–1661, 2011.

[8] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *The Journal of Machine Learning Research*, 6:1579–1619, 2005.

[9] Manu Nandan, Pramod P. Khargonekar, and Sachin S. Talathi. Fast svm training using approximate extreme points. *J. Mach. Learn. Res.*, 15(1):59–98, January 2014.

[10] John C Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods*, pages 185–208. MIT Press, 1999.

[11] Nikolas List and Hans Ulrich Simon. General polynomial time decomposition algorithms. *J. Mach. Learn. Res.*, 8:303–321, May 2007.

[12] S. Sathiya Keerthi, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and Karuturi Radha Krishna Murthy. Improvements to platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3):637–649, 2001.

[13] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, pages 409–415, 2001.

[14] Claudio Cifarelli, Mario R Guarracino, Onur Seref, Salvatore Cuciniello, and Panos M Pardalos. Incremental classification with generalized eigenvalues. *Journal of classification*, 24(2):205–219, 2007.

[15] Shai Fine and Katya Scheinberg. Incremental learning and selective sampling via parametric optimization framework for svm. *Advances in neural information processing systems*, 1:705–712, 2002.

[16] Pavel Laskov, Christian Gehl, Stefan Krüger, and Klaus-Robert Müller. Incremental support vector learning: Analysis, implementation and applications. *The Journal of Machine Learning Research*, 7:1909–1936, 2006.

[17] Liva Ralaivola and Florence d'Alché Buc. Incremental support vector machine learning: A local approach. In *Artificial Neural Networks—ICANN 2001*, pages 322–330. Springer, 2001.

[18] Zhizheng Liang and YouFu Li. Incremental support vector machine learning in the primal and applications. *Neurocomputing*, 72(10):2249–2258, 2009.

[19] Alistair Shilton, Marimuthu Palaniswami, Daniel Ralph, and Ah Chung Tsoi. Incremental training of support vector machines. *Neural Networks, IEEE Transactions on*, 16(1):114–131, 2005.

[20] Gaëlle Loosli, Stéphane Canu, SVN Vishwanathan, and Alex Smola. Invariances in classification: an efficient svm implementation. In *ASMDA*, 2005.

[21] Hsiang-Fu Yu, Cho-Jui Hsieh, Kai-Wei Chang, and Chih-Jen Lin. Large linear classification when data cannot fit in memory. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):23, 2012.

[22] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR : A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[23] Thorsten Joachims. Making large-scale svm learning practical. Technical report, Technische Universität Dortmund, Sonderforschungsbereich 475: Komplexitätsreduktion in multivariaten Datenstrukturen, 1998.

[24] Tobias Glasmachers and Christian Igel. Maximum-gain working set selection for svms. *The Journal of Machine Learning Research*, 7:1437–1466, 2006.

[25] Chien-Ming Huang, Yuh-Jye Lee, Dennis K.J. Lin, and Su-Yun Huang. Model selection for support vector machines via uniform design. *Computational Statistics & Data Analysis*, 52(1):335 – 346, 2007.

[26] Tobias Glasmachers. On related violating pairs for working set selection in smo algorithms. In *ESANN*, pages 475–480. Citeseer, 2008.

[27] Gaëlle Loosli, Gilles Gasso, and Stéphane Canu. Regularization paths for ν-svm and ν-svr. In *Advances in Neural Networks–ISNN 2007*, pages 486–496. Springer, 2007.

[28] Tobias Glasmachers and Christian Igel. Second-order smo improves svm online and active learning. *Neural Computation*, 20(2):374–382, 2008.

[29] Kai-Wei Chang and Dan Roth. Selective block minimization for faster convergence of limited memory large-scale linear models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 699–707. ACM, 2011.

[30] S. V. N. Vishwanathan, Alex J. Smola, and M. Narasimha Murty. Simplesvm. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 760–767. AAAI Press, 2003.

[31] Linda Kaufman. Solving the quadratic programming problem arising in support vector classification. In *Advances in kernel methods*, pages 147–167. MIT Press, 1999.

[32] Olvi L Mangasarian and David R Musicant. Successive overrelaxation for support vector machines. *Neural Networks, IEEE Transactions on*, 10(5):1032–1037, 1999.

[33] Christopher P Diehl and Gert Cauwenberghs. SVM incremental learning, adaptation and optimization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2685–2690. IEEE, 2003.

[34] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[35] Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.

[36] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

[37] Fan Rong-En, Chen Pai-Hsuen, Lin Chih-Jen, and Thorsten Joachims. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6(12):1889 – 1918, 2005.

[38] Yuh-Jye Lee and O.L. Mangasarian. SSVM : A smooth support vector machine for classification. *Computational Optimization and Applications*, 20(1):5–22, 2001.

[39] Gaëlle Loosli and Stéphane Canu. Comments on the core vector machines: Fast svm training on very large data sets. *The Journal of Machine Learning Research*, 8:291–301, 2007.

[40] David MJ Tax and Pavel Laskov. Online svm learning: from classification to data description and back. In *Neural Networks for Signal Processing, 2003. NNSP'03. 2003 IEEE 13th Workshop on*, pages 499–508. IEEE, 2003.

[41] John Shawe-Taylor and Shiliang Sun. A review of optimization methodologies in support vector machines. *Neurocomputing*, 74(17):3609–3618, 2011.

[42] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos : primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

[43] Gyemin Lee and C.D. Scott. The one class support vector machine solution path. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–521–II–524, April 2007.

[44] Francis R Bach, David Heckerman, and Eric Horvitz. Considering cost asymmetry in learning classifiers. *The Journal of Machine Learning Research*, 7:1713–1741, 2006.

[45] F. de Morsier, D. Tuia, M. Borgeaud, V. Gass, and J.-P. Thiran. Semi-supervised novelty detection using svm entire solution path. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(4):1939–1950, April 2013.

[46] Gyemin Lee and Clayton Scott. Nested support vector machines. *Signal Processing, IEEE Transactions on*, 58(3):1648–1660, 2010.

[47] Xiaoyun Wu and Rohini K Srihari. New ν-Support Vector Machines and their sequential minimal optimization. In *ICML*, pages 824–831, 2003.

[48] Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

[49] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[50] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[51] LEE Changki. Pegasos algorithm for one-class support vector machine. *IEICE TRANSACTIONS on Information and Systems*, 96(5):1223–1226, 2013.

[52] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46(1):235–262, 2013.

[53] Assaf Glazer, Michael Lindenbaum, and Shaul Markovitch. q-OCSVM : A q-quantile estimator for high-dimensional distributions. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 503–511. Curran Associates, Inc., 2013.

[54] Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012.

[55] Yu-Feng Li, James T Kwok, and Zhi-Hua Zhou. Cost-sensitive semi-supervised support vector machine. In *AAAI*, volume 10, pages 500–505, 2010.

[56] Hamed Masnadi-Shirazi, Nuno Vasconcelos, and Arya Iranmehr. Cost-sensitive support vector machines. *arXiv preprint arXiv:1212.0975*, 2012.

[57] Chan-Yun Yang, Jr-Syu Yang, and Jian-Jun Wang. Margin calibration in svm class-imbalanced learning. *Neurocomputing*, 73(1):397–411, 2009.

[58] Peter Geibel and Fritz Wysotzki. Perceptron based learning with example dependent and noisy costs. In *ICML*, pages 218–225, 2003.

[59] Ulf Brefeld, Peter Geibel, and Fritz Wysotzki. Support vector machines with example dependent costs. In *Machine Learning: ECML 2003*, pages 23–34. Springer, 2003.

[60] Clayton Scott. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 153–160, New York, NY, USA, June 2011. ACM.

[61] Yaoyong Li and John Shawe-Taylor. The svm with uneven margins and chinese document categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, pages 216–227. MIT Press, 2003.

[62] Mark A. Davenport, Richard G. Baraniuk, and Clayton D. Scott. Tuning support vector machines for minimax and neyman-pearson classification. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32(10):1888 – 1898, 2010.

[63] Xin Jin, Yujian Li, Yihua Zhou, and Zhi Cai. Applying average density to example dependent costs svm based on data distribution. *Journal of computers*, 8(1):91–96, 2013.

[64] Fernando Pérez-Cruz, Jason Weston, DJL Herrmann, and B Scholkopf. Extension of the nu-svm range for classification. *NATO SCIENCE SERIES SUB SERIES III COMPUTER AND SYSTEMS SCIENCES*, 190:179–196, 2003.

[65] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207 – 1245, 2000.