ML2R Theory Nuggets
The Dual Problem of L_2 Support Vector Machine Training

Christian Bauckhage
Machine Learning Rhine-Ruhr
University of Bonn
Bonn, Germany

Rafet Sifa
Machine Learning Rhine-Ruhr
Fraunhofer IAIS
St. Augustin, Germany

ABSTRACT

We derive the dual problem of L_2 support vector machine training. This involves setting up the Lagrangian of the primal problem and working with the Karush-Kuhn-Tucker conditions. As a payoff, we find that the dual poses a rather simple optimization problem that can be solved by the Frank-Wolfe algorithm.

1 INTRODUCTION

Consider a set of labeled training data $\big\{(\boldsymbol{x}_j, y_j)\big\}_{j=1}^n$ where the data $\boldsymbol{x}_j \in \mathbb{R}^m$ have been sampled from two classes Ω_1 and Ω_2 and the labels $y_j \in \{-1, +1\}$ indicate class membership in the sense that

$$y_j = \begin{cases} +1 & \text{if } \ \boldsymbol{x}_j \in \Omega_1 \\ -1 & \text{if } \ \boldsymbol{x}_j \in \Omega_2 \end{cases}$$

Give data like these, we might want to train a **binary classifier** that can predict class labels of previously unseen data. A simple yet fairly general mathematical model of such a machine is a **binary *linear* classifier**, i.e. a function $y : \mathbb{R}^m \to \{-1, +1\}$ where

$$\begin{aligned} y(\boldsymbol{x}) &= \begin{cases} +1 & \text{if } \boldsymbol{w}^\top \boldsymbol{x} - \theta \geq 0 \\ -1 & \text{otherwise} \end{cases} \\ &= \operatorname{sign}\big(\boldsymbol{w}^\top \boldsymbol{x} - \theta\big) \end{aligned}$$

The two parameters $\boldsymbol{w} \in \mathbb{R}^m$ and $\theta \in \mathbb{R}$ of such a classifier are called *weight vector* and *threshold value* and training means to estimate them from the training data. This is commonly formalized as an optimization problem and there are many ideas of what to actually optimize. Depending on which optimization criterion is chosen, the resulting estimates may differ and the trained classifier may work more or less well and will go by different names (naïve Bayes classifier, least squares classifier, linear discriminant classifier, support vector machine, …).

In this note, we are concerned with **support vector machines (SVMs)**. These, too, come in different kinds and flavors. The most well known are the standard SVMs due to Cortes and Vapnik [5]. Other popular flavors include least squares SVMs due to Suykens and Vanderwalle [15] or ν-SVMs by Schölkopf et al. [11]. A surprisingly less well known variant are L_2 SVMs whose origins can be traced back to work by Frieß and Harrison [7] and Mangasarian and Musicant [9]. These are the ones, we will study here.

L_2 SVMs are known to be robust, reliable, and fast and easy to train [1]. The latter is because the optimization problem that needs to be solved when training them is comparatively simple and to work this out is what this note is all about.

In the following, we first present the primal problem of training an L_2 SVM and then derive the corresponding dual problem. This is the problem we consider to be simple and we conclude our discussion by explaining why this is.

Note: This theory nugget does not provide an introduction to support vector machines but assumes that readers are familiar with the underlying theory and the respective jargon (margins, slack variables, …).

2 THE PRIMAL L_2 SVM TRAINING PROBLEM

Recall that, during training, a (linear) support vector machine for binary classification tries to learn the maximum-margin hyperplane between the training data from the two classes. Any hyperplane can be characterized by $\boldsymbol{w}^\top \boldsymbol{x} - \theta = 0$ and the idea is to estimate $\boldsymbol{w}$ and θ such that the distance or *margin* $\rho \in \mathbb{R}$ between the plane and the nearest training data $\boldsymbol{x}_j$ from either class is maximized. Should the two classes overlap, there is no separating hyperplane. This issue can be circumvented by trying to identify a plane such that the individual margins $\rho - \xi_j$ are maximize where the $\xi_j \geq 0 \in \mathbb{R}$ are *slack variables* which we may gather in a vector $\boldsymbol{\xi} \in \mathbb{R}^n$.

While this training objective appears well specified, there once again are several ways of how to formalize it. For instance, the classical (or L_1) SVMs due to Cortes and Vapnik [5] solve the following primal problem

$$\begin{aligned} \underset{\boldsymbol{w}, \theta, \boldsymbol{\xi}}{\operatorname{argmin}} \ & \tfrac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{j=1}^n \xi_j \\ \text{s.t.} \ & \begin{aligned} y_j \cdot \big(\boldsymbol{w}^\top \boldsymbol{x}_j - \theta\big) - 1 + \xi_j \geq 0, \quad & 1 \leq j \leq n \\ \xi_j \geq 0, \quad & 1 \leq j \leq n \end{aligned} \end{aligned}$$

where $C \geq 0 \in \mathbb{R}$ is a parameter that allows for tuning the slack.

However, in this note, we are interested in L_2 SVMs and the **primal problem of training an L_2 support vector machine** is to solve

$$\begin{aligned} \underset{\boldsymbol{w}, \theta, \rho, \boldsymbol{\xi}}{\operatorname{argmin}} \ & \tfrac{1}{2} \|\boldsymbol{w}\|^2 + \tfrac{1}{2} \theta^2 - \rho + \tfrac{C}{2} \sum_{j=1}^n \xi_j^2 \\ \text{s.t.} \ & y_j \cdot \big(\boldsymbol{w}^\top \boldsymbol{x}_j - \theta\big) - \rho + \xi_j \geq 0, \quad 1 \leq j \leq n \end{aligned} \tag{1}$$

Looking at this constrained optimization problem, several remarks appear to be in order:

In contrast to the primal L_1 SVM training problem, all occurrences of slack variables in the objective function are now in form of squares hence the name L_2 SVM. Moreover, since the slack variables enter the objective function as squares, the sum over these squares can never be negative. This is why the non-negativity constraints $\xi_j \geq 0$ have been dropped.

Again in contrast to the primal L_1 SVM training problem, the objective function of the primal L_2 SVM training problem also involves the parameters θ and ρ. While the L_2 training problem may thus look more daunting than the L_1 training problem, we will see below that the way these parameters enter the problem has been designed very cleverly. In fact, the primal L_2 training problem has been formalized so cleverly that its practically more important dual is downright simple. All this is to say that we actually do not have to worry about the ostensible overhead in (1).

However, before we begin to derive the dual problem of L_2 SVM training, we will first (re)write the primal problem in a more concise form. To this end, we introduce a new symbol and write

$$z_j = y_j \cdot \boldsymbol{x}_j \tag{2}$$

to denote training data point $\boldsymbol{x}_j$ weighted by its label value y_j. Next, we gather all the weighted training data in a data matrix

$$Z = \left[z_1, z_2, \ldots, z_n\right] \tag{3}$$

of n columns. And, in a simmilar vein, we also gather the given training labels y_j in an n-dimensional vector

$$\boldsymbol{y} = \left[y_1, y_2, \ldots, y_n\right]^\intercal \tag{4}$$

Moreover, with respect to the two more complicated terms in the objective function of (1), we note that we can write

$$\|\boldsymbol{w}\|^2 = \boldsymbol{w}^\intercal \boldsymbol{w} \tag{5}$$

as well as

$$\sum_{j=1}^{n} \xi_j^2 = \boldsymbol{\xi}^\intercal \boldsymbol{\xi} \tag{6}$$

Finally, we let $\mathbf{0}, \mathbf{1} \in \mathbb{R}^n$ denote the vectors of all zeros and all ones, respectively. All of this allows us to express the primal L_2 SVM training problem as

$$\begin{aligned} \underset{\boldsymbol{w},\theta,\rho,\boldsymbol{\xi}}{\operatorname{argmin}} \quad & \tfrac{1}{2}\boldsymbol{w}^\intercal\boldsymbol{w} + \tfrac{1}{2}\theta^2 - \rho + \tfrac{C}{2}\boldsymbol{\xi}^\intercal\boldsymbol{\xi} \\ \text{s.t.} \quad & \left[Z^\intercal \boldsymbol{w} - \theta\cdot\boldsymbol{y}\right] - \rho\cdot\mathbf{1} + \boldsymbol{\xi} \geq \mathbf{0} \end{aligned} \tag{7}$$

3 THE DUAL L_2 SVM TRAINING PROBLEM

In order to derive the dual of the above primal problem, we first of all note that (7) constitutes a quadratic minimization problem with a total of n greater-than-or-equal-to constraints subsumed in a single matrix-vector expression.

Hence, if we introduce n **Lagrange multipliers** μ_j which we may gather in a vector $\boldsymbol{\mu} \in \mathbb{R}^n$, we obtain the following **Lagrangian**

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, \theta, \boldsymbol{\xi}, \rho, \boldsymbol{\mu}) = {} & \tfrac{1}{2}\boldsymbol{w}^\intercal\boldsymbol{w} + \tfrac{1}{2}\theta^2 - \rho + \tfrac{C}{2}\boldsymbol{\xi}^\intercal\boldsymbol{\xi} \\ & - \boldsymbol{\mu}^\intercal\left[Z^\intercal\boldsymbol{w} - \theta\cdot\boldsymbol{y} - \rho\cdot\mathbf{1} + \boldsymbol{\xi}\right] \end{aligned} \tag{8}$$

Now, recall that the **Karush-Kuhn-Tucker conditions** describe a set of criteria any valid solution to our inequality constrained problem must fulfill [3, 4] . The KKT 1 condition (stationarity), for instance, demands that, at a solution, we must have $\nabla\mathcal{L} = \mathbf{0}$.

Next, we therefore partially derive our Lagrangian with respect to its parameters and equate the resulting expressions to zero. If we recall some basic rules from multivariate calculus [10], this is easily done and we find

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = \boldsymbol{w} - Z\boldsymbol{\mu} \overset{!}{=} \mathbf{0} \quad\Rightarrow\quad \boldsymbol{w} = Z\boldsymbol{\mu} \tag{9}$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = \theta + \boldsymbol{y}^\intercal\boldsymbol{\mu} \overset{!}{=} 0 \quad\Rightarrow\quad \theta = -\boldsymbol{y}^\intercal\boldsymbol{\mu} \tag{10}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}} = C\,\boldsymbol{\xi} + \boldsymbol{\mu} \overset{!}{=} \mathbf{0} \quad\Rightarrow\quad \boldsymbol{\xi} = -\frac{1}{C}\,\boldsymbol{\mu} \tag{11}$$

$$\frac{\partial\mathcal{L}}{\partial\rho} = -1 + \mathbf{1}^\intercal\boldsymbol{\mu} \overset{!}{=} 0 \quad\Rightarrow\quad 1 = \mathbf{1}^\intercal\boldsymbol{\mu} \tag{12}$$

Plugging the three results in (9), (10), and (11) back into (8) eliminates the parameters $\boldsymbol{w}$, θ, and $\boldsymbol{\xi}$ and the Lagrangian becomes

$$\begin{aligned} \mathcal{L}(\rho, \boldsymbol{\mu}) = {} & \tfrac{1}{2}\boldsymbol{\mu}^\intercal Z^\intercal Z\boldsymbol{\mu} + \tfrac{1}{2}\boldsymbol{\mu}^\intercal\boldsymbol{y}\boldsymbol{y}^\intercal\boldsymbol{\mu} - \rho + \tfrac{1}{2C}\boldsymbol{\mu}^\intercal\boldsymbol{\mu} \\ & - \boldsymbol{\mu}^\intercal Z^\intercal Z\boldsymbol{\mu} - \boldsymbol{\mu}^\intercal\boldsymbol{y}\boldsymbol{y}^\intercal\boldsymbol{\mu} + \rho\cdot\boldsymbol{\mu}^\intercal\mathbf{1} - \tfrac{1}{C}\boldsymbol{\mu}^\intercal\boldsymbol{\mu} \end{aligned} \tag{13}$$

Further simplification and application of the result in (12) leads to an even cleaner expression

$$\begin{aligned} \mathcal{L}(\rho, \boldsymbol{\mu}) &= -\tfrac{1}{2}\boldsymbol{\mu}^\intercal Z^\intercal Z\boldsymbol{\mu} - \tfrac{1}{2}\boldsymbol{\mu}^\intercal\boldsymbol{y}\boldsymbol{y}^\intercal\boldsymbol{\mu} - \tfrac{1}{2C}\boldsymbol{\mu}^\intercal\boldsymbol{\mu} \\ &= -\tfrac{1}{2}\boldsymbol{\mu}^\intercal\left[Z^\intercal Z + \boldsymbol{y}\boldsymbol{y}^\intercal + \tfrac{1}{C}\boldsymbol{I}\right]\boldsymbol{\mu} \\ &\equiv \mathcal{D}(\boldsymbol{\mu}) \end{aligned} \tag{14}$$

Function $\mathcal{D}(\boldsymbol{\mu})$ is called the **Lagrangian dual** and we note that it only depends on the Lagrange multipliers μ_j in vector $\boldsymbol{\mu}$. We also note that $\mathcal{D}(\boldsymbol{\mu})$ is a **(definite) quadratic form** in $\boldsymbol{\mu}$ or, more specifically, a **concave function** (due to the scaling factor of $-1/2$) so that it has a unique maximum.

However, if we set out to maximize $\mathcal{D}(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$ we must incorporate two constraints. The one is a consequence of (12) which demands that $\mathbf{1}^\intercal\boldsymbol{\mu} = 1$; the other is due to the KKT 3 condition (dual feasibility) which demands that $\boldsymbol{\mu} \geq \mathbf{0}$.

Because of **Lagrange duality** and the two constraints we just pointed out, we therefore find that **the dual problem of training an L_2 support vector machine** consists in solving

$$\begin{aligned} \underset{\boldsymbol{\mu}}{\operatorname{argmax}} \quad & -\tfrac{1}{2}\boldsymbol{\mu}^\intercal\left[Z^\intercal Z + \boldsymbol{y}\boldsymbol{y}^\intercal + \tfrac{1}{C}\,\boldsymbol{I}\right]\boldsymbol{\mu} \\ \text{s.t.} \quad & \mathbf{1}^\intercal\boldsymbol{\mu} = 1 \\ & \boldsymbol{\mu} \geq \mathbf{0} \end{aligned} \tag{15}$$

4 CONCLUSION

L_2 SVMs as discussed in this note are a lesser known member of the support vector machine family. We presented the primal problem of training such an SVM (7) and then derived the corresponding dual problem (15). To conclude our discussion, we note the following:

First of all, if we could solve the problem in (15) for the optimal vector of Lagrange multipliers $\boldsymbol{\mu}_*$, we could use it to determine the actually sought after parameters $\boldsymbol{w}$ and θ of our support vector machine. This is thanks to (9) and (10) which provide us with

$$\boldsymbol{w} = Z\boldsymbol{\mu}_* \tag{16}$$

$$\theta = -\boldsymbol{y}^\intercal\boldsymbol{\mu}_* \tag{17}$$

Second of all, the major difference between L_2 SVMs and the better known classical L_1 SVMs is that the slack variables enter the primal objective in squared form. From the point of view of the performance of a correspondingly trained classifier, this makes no difference whatsoever [1, 8]. However, from the point of view of ease of training it does.

We just saw that one of the favorable properties of the cleverly designed primal L_2 SVM training problem is that the corresponding dual is rather simple. Although we are dealing with a constrained optimization problem, the feasible set in which the optimal solution must reside is just the standard simplex

$$\Delta^{n-1} = \left\{ \boldsymbol{\mu} \in \mathbb{R}^n \;\middle|\; \boldsymbol{\mu} \geq \mathbf{0} \wedge \mathbf{1}^\intercal \boldsymbol{\mu} = 1 \right\} \tag{18}$$

Moreover, since our objective function is quadratic and concave, maximizing $\mathcal{D}(\boldsymbol{\mu})$ is the same as minimizing $-\mathcal{D}(\boldsymbol{\mu})$.

These two observation, however, mean that we could also write the dual L_2 SVM training problem as

$$\underset{\boldsymbol{\mu} \in \Delta^{n-1}}{\operatorname{argmin}} \; \tfrac{1}{2} \, \boldsymbol{\mu}^\intercal \left[\boldsymbol{Z}^\intercal \boldsymbol{Z} + \boldsymbol{y}\boldsymbol{y}^\intercal + \tfrac{1}{C} \boldsymbol{I} \right] \boldsymbol{\mu} \tag{19}$$

Third of all, the expression in (19) now immediately reveals that the dual L_2 SVM training problem is a convex minimization problem over a compact convex set!

These are the kind of problems where the Frank-Wolfe algorithm excels [6]. It is easy to implement and converges fairly quickly to the optimal solution [1, 13, 14] and can even be realized by neural networks [2, 12].

All of this is to say that it is easy to train an L_2 Support vector machine and we will discuss implementation details in our ML2R coding nuggets series.

ACKNOWLEDGMENTS

This material was produced within the Competence Center for Machine Learning Rhine-Ruhr (**ML2R**) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01IS18038C). The authors gratefully acknowledge this support.

REFERENCES

[1] C.M. Alaiz and J.A.K. Suykens. 2018. Modified Frank-Wolfe Algorithm for Enhanced Sparsity in Support Vector Machine Classifiers. *Neurocomputing* 320, Dec (2018).

[2] C. Bauckhage. 2017. A Neural Network Implementation of Frank-Wolfe Optimization. In *Proc. ICANN*.

[3] C. Bauckhage and T. Dong. 2019. Lecture Notes on Machine Learning: The Karush-Kuhn-Tucker Conditions (Part 2). B-IT, University of Bonn.

[4] C. Bauckhage and D. Speicher. 2019. Lecture Notes on Machine Learning: The Karush-Kuhn-Tucker Conditions (Part 1). B-IT, University of Bonn.

[5] C. Cortes and V. Vapnik. 1995. Support Vector Networks. *Machine Learning* 20, 3 (1995).

[6] M. Frank and P. Wolfe. 1956. An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly* 3, 1–2 (1956).

[7] T.T. Frieß and R.F. Harrison. 1998. *The Kernel Adatron With Bias Unit: Analysis of the Algorithm (Part 1)*. Technical Report ACSE Research Report 729. Dept. of Automatic Control and Systems Engineering, University of Sheffield.

[8] Y. Koshiba and S. Abe. 2003. Comparison of L1 and L2 Support Vector Machines. In *Proc. IJCNN*.

[9] O.L. Mangasarian and D.R. Musicant. 2001. Lagrangian Support Vector Machines. *J. of Machine Learning Research* 1 (2001), 161–177.

[10] K. B. Petersen and M. S. Pedersen. 2012. *The Matrix Cookbook*. Technical University of Denmark.

[11] B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. 2000. New Support Vector Algorithms. *Neural Computation* 12, 5 (2000).

[12] P. Schramowski, C. Bauckhage, and K. Kersting. 2018. Neural Conditional Gradients. *arXiv:1803.04300[cs.LG]* (2018).

[13] R. Sifa. 2018. An Overview of Frank-Wolfe Optimization for Stochasticity Constrained Interpretable Matrix and Tensor Factorization. In *Proc. ICANN*.

[14] R. Sifa, D. Paurat, D. Trabold, and C. Bauckhage. 2018. Simple Recurrent Neural Networks for Support Vector Machine Training. In *Proc. ICANN*.

[15] J.A.K. Suykens and J.P.L. Venderwalle. 1999. Lest Squares Support Vector Machine Classifiers. *Neural Processing Letters* 9, 3 (1999).