GEOMETRY OF SUPPORT VECTOR MACHINES AND FEATURE SELECTION

MOHAMMAD REZA KARIMI

Abstract. In the class, the SVMs where introduced as a variation to the Perceptron algorithm: SVM uses the Hinge loss instead of the Perceptron loss. In the first part of this handout, we give a geometrical interpretation to what SVM does, and return to the Hinge loss from geometry. For interested readers, we also bring the concept of support vectors, as well as the dual of the SVM problem, which can be considered as a prelude to kernels. In the second part, we bring some examples for the problem of feature selection in machine learning. Sections marked with a * are considered as optional.

Contents

Notations. We do not use bold font to distinguish vectors from scalars, as it is easily understood from the context. If x is a vector, its ith component is denoted by x^i. We index different vectors by lower indices, e.g., x_1 and x_j. We denote by $\langle u, v\rangle$ the inner product of u and v, defined by $\langle u, v\rangle = \sum u^i v^i$. The Euclidean norm of a vector x is denoted by $\|x\|$.

Learning and Adaptive Systems, ETH Zürich
E-mail address: mkarimi@ethz.ch.
Date: March 2019.

Part 1. Geometry of SVM

0. Preliminaries

Before starting off, we bring some facts about (affine) hyperplanes. A *hyperplane* in $\mathbb{R}^d$ is defined by the set of solutions to a single linear equation. That is, it is the set of all $x \in \mathbb{R}^d$ satisfying

$$\langle w, x\rangle = w^1x^1 + \cdots + w^dx^d = 0,$$

where $w = (w^1, \ldots, w^d) \in \mathbb{R}^d$ is a nonzero vector, and is sometimes called *the normal vector*, as it is orthogonal to all points on the hyperplane. Note that the dimension of a hyperplane is $d-1$, as it is the kernel of a linear transformation with one dimensional image. See the recap slides to learn more about kernels and images of linear transformations.

For $b \in \mathbb{R}$, the set $\{x \in \mathbb{R}^d \mid \langle w, x\rangle = b\}$ is a translated hyperplane, which is sometimes called an *affine* hyperplane, or simply a hyperplane. This hyperplane is parallel to $\{x \mid \langle w, x\rangle = 0\}$. Increasing b will move the hyperplane in the direction of w, and decreasing b will move the hyperplane in the direction of $-w$. Sometimes b is called the *intercept* of the hyperplane.

For a point $z \in \mathbb{R}^d$, the orthogonal projection of z on the hyperplane $\langle w, x\rangle = b$ can be computed (geometrically) as follows: one moves z along the line defined by w, until it hits the hyperplane. Concretely, one is looking for $t \in \mathbb{R}$ such that

$$\langle w, z + tw\rangle = b.$$

Expanding the inner product gives $\langle w, z\rangle + t\|w\|^2 = b$, or $t = \frac{b-\langle w,z\rangle}{\|w\|^2}$. One is also able to compute the Euclidean distance, which is $\|tw\| = |t| \cdot \|w\| = \frac{|b-\langle w,z\rangle|}{\|w\|}$. A special case is demonstrated in the following lemma.

Lemma 1. *Let $w \in \mathbb{R}^d$ and $H = \{x \in \mathbb{R}^d \mid \langle w, x\rangle = 0\}$ be a hyperplane. The orthogonal distance of a point $z \in \mathbb{R}^d$ to H can be computed as*

$$\frac{|\langle w, z\rangle|}{\|w\|}.$$

Specifically, if w is a unit vector, the inner product $\langle w, z\rangle$ directly gives the distance of z to H.

The last reminder is about separation. A hyperplane $H = \{x \in \mathbb{R}^d \mid \langle w, x\rangle = b\}$ divides the space into three regions: the points on H, the *half-space* where w is pointing to, $\{x \mid \langle w, x\rangle - b > 0\}$, and the half-space on the other side of H, which is $\{x \mid \langle w, x\rangle - b < 0\}$.

1. Linear Classification

Let $d \in \mathbb{N}$ and $\mathcal{D}$ be an (unknown, but fixed) distribution over $\mathbb{R}^d \times \{-1, 1\}$. Assume that we are given n i.i.d. samples $(x_1, y_1), \ldots, (x_n, y_n) \sim \mathcal{D}$, where $x_i \in \mathbb{R}^d$ are the features, and $y_i \in \{-1, 1\}$ are the labels. For now, we make the following assumption (we will relax this assumption in later sections):

Assumption 2 (Linear Separability)**.** *There exists a hyperplane that separates the data points with $y = 1$ from those with $y = -1$. That is, there exists $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that*

$$y_i(\langle w, x_i\rangle - b) \geq 0, \quad \forall i \in \{1, \ldots, n\}.$$

Any such hyperplane is called a separating hyperplane. *Note that w is pointing towards the datapoints with $y = 1$.*

Our goal is to find (among all possible separating hyperplanes) the hyperplane H with maximum *margin*, where the margin is the distance of the closest point among $x_1, \ldots, x_n$ to H, see Figure 1. The reason behind this choice of a hyperplane is two-fold: firstly, the intuitive picture

suggests that this hyperplane is the *correct* separating hyperplane. Secondly, it is related to generalisation power and stability, which we do not cover in this handout.

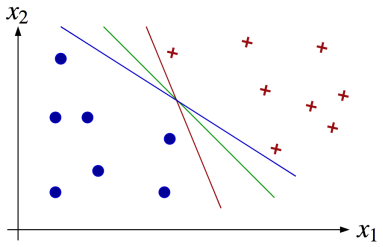

FIGURE 1. Different separating hyperplanes for a dataset. The green line denotes the largest margin separator.

2. Maximum Margin Separator

In the previous section, we worked with affine hyperplanes. To make the computations easier, we augment each x_i with a 1, and augment w with $-b$, i.e.,

$$x_i \leftarrow (x_i^1, \dots, x_i^d, 1),$$
$$w \leftarrow (w^1, \dots, w^d, -b).$$

Using this new x_i and w, we only need to consider hyperplanes passing through the origin in $\mathbb{R}^{d+1}$ and forget about the intercept. For ease of notation, let us also define $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ to be the complete dataset.

Definition 3 (Margin)**.** *Let H be any separating hyperplane of D, defined via w as above. The margin of H with respect to D is defined as*

$$\gamma_D(w) := \min_{(x,y)\in D} \frac{|\langle w, x\rangle|}{\|w\|}.$$

By the results so far, one way to write down the problem of finding the maximum margin separator is the following optimisation problem:

$$\begin{aligned} &\underset{w}{\text{maximise}}\, \gamma_D(w) \\ &\text{s.t.} \quad y_i \cdot \langle w, x_i\rangle \geq 0 \text{ for all } i \in \{1, \dots, n\}. \end{aligned} \tag{2.1}$$

The constraint in the problem above is simply stating that w is a separating hyperplane of D.

We need another assumption to proceed (this assumption will be removed in the following sections).

Assumption 4. *There exists a separating hyperplane with a nonzero margin.*

The reader is encouraged to create an example that this assumption does not hold. Making this assumption is necessary; otherwise, the optimisation problem (2.1) has the trivial solution of 0.

Let w be such a separating hyperplane. Define $a := \min |\langle w, x_i\rangle| > 0$. This means that

$$|\langle w, x_i\rangle| \geq a,$$

which implies, by the fact that w is a separating hyperplane, that

$$y_i \cdot \langle w, x_i\rangle \geq a.$$

Dividing w by a, gives rise to a separator $\tilde{w}$ with the same margin, but with the new constraints

$$y_i \cdot \langle \tilde{w}, x_i \rangle \geq 1.$$

Note that in this case, $\gamma_D(\tilde{w}) = 1/\|\tilde{w}\|$. Thus, maximising the margin is equivalent to minimising $\|\tilde{w}\|$, or similarly, minimising $\|\tilde{w}\|^2$. The final optimisation problem can be stated as follows:

Definition 5 (Hard-SVM)**.** *Let D be a linearly separable dataset, where Assumption 4 holds. The following optimization problem (called the Hard-SVM) yields the maximum margin separator for D:*

$$\begin{aligned} &\underset{w}{\text{minimise}}\ \tfrac{1}{2}\|w\|^2 \\ &\text{s.t.}\quad y_i \cdot \langle w, x_i \rangle \geq 1 \text{ for all } i \in \{1, \ldots, n\}. \end{aligned} \tag{2.2}$$

3. Soft-SVM

In the last section, we had two assumptions (Assumptions 2 and 4) about the dataset D, which may not hold in a general dataset. Note that when these assumptions are violated, the optimisation problem (2.2) has no solutions. In this section, we drop these assumptions, and derive a *soft* version of SVM.

The key idea of this derivation is letting some constraints in 2.2 to be violated. One way to achieve this is to introduce some *slack variables* for each constraint. Let $\xi_1, \ldots, \xi_n \geq 0$, and define a new optimization problem

$$\begin{aligned} &\underset{w,\xi_1,\ldots,\xi_n}{\text{minimise}}\ \tfrac{1}{2}\|w\|^2 \\ &\text{s.t.}\quad y_i \cdot \langle w, x_i \rangle \geq 1 - \xi_i, \text{ for all } i \in \{1, \ldots, n\}, \\ &\qquad \xi_i \geq 0, \text{ for all } i \in \{1, \ldots, n\}. \end{aligned} \tag{3.1}$$

The variables ξ_i denote the amount that the ith constraint is violated: $\xi_i = 0$ denotes no violation, and larger ξ_i shows larger violation, see Figure 2.

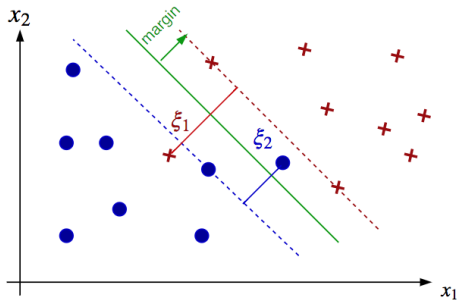

Figure 2. Slack variables in Soft-SVM.

Although this optimisation problem has solutions for every dataset D (even the ones that are not linearly separable), it is easy to see that its solution is always $w = 0$. This is not surprising, as we have not defined any cost for violation of constraints. The simplest way to define such a cost, is as follows:

$$\begin{aligned} &\underset{w,\xi_1,\ldots,\xi_n}{\text{minimise}}\ \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.}\quad y_i \cdot \langle w, x_i \rangle \geq 1 - \xi_i, \text{ for all } i \in \{1, \ldots, n\}, \\ &\qquad \xi_i \geq 0, \text{ for all } i \in \{1, \ldots, n\}, \end{aligned} \tag{3.2}$$

where $C > 0$ denotes the softness of the problem: larger C goes for larger violation penalties. The optimisation problem (3.2) is called *Soft-SVM*. One can observe that when $C \to \infty$, the Soft-SVM problem gets similar to Hard-SVM problem.

4. From Soft-SVM to Hinge Loss

Let us start from optimisation problem (3.2). For a fixed w, the optimisation problem is equivalent to

$$\begin{aligned} &\underset{\xi_1,\dots,\xi_n}{\text{minimise}} \sum_{i=1}^{n} \xi_i \\ &\text{s.t.} \quad \xi_i \geq 1 - y_i \cdot \langle w, x_i \rangle \,, \text{ for all } i \in \{1,\dots,n\}, \\ &\qquad \xi_i \geq 0, \text{ for all } i \in \{1,\dots,n\}. \end{aligned} \tag{4.1}$$

In a more compact way, the set of constraints is the same as

$$\xi_i \geq \max\{0, 1 - y_i \cdot \langle w, x_i \rangle\} = \ell_{\text{hinge}}(y_i \cdot \langle w, x_i \rangle),$$

where ℓ_{hinge} is the Hinge loss defined as

$$\ell_{\text{hinge}}(t) = \max\{0, 1 - t\}.$$

As the objective of the optimisation is $\sum \xi_i$, and the constraints are independent of eachother, we find the optimal values of ξ_i by setting them to their lower bounds:

$$\xi_i^\star(w) = \ell_{\text{hinge}}(y_i \cdot \langle w, x_i \rangle).$$

Note that this optimal value is evaluated for a fixed w, thus we used the notation $\xi_i^\star(w)$. This derivation suggests the following optimisation problem:

$$\underset{w}{\text{minimise}} \; \tfrac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \ell_{\text{hinge}}(y_i \cdot \langle w, x_i \rangle). \tag{4.2}$$

The following lemma states that this new optimisation problem has the same solution as in (3.2).

Lemma 6. *The optimisation problems (4.2) and (3.2) have the same solution.*

Proof. Let $w^\star$ and $\xi_1^\star, \dots, \xi_n^\star$ be the optimal solutions to (3.2). Feasibility implies that for all $i \in \{1,\dots,n\}$,

$$\xi_i^* \geq \ell_{\text{hinge}}(y_i \cdot \langle w^\star, x_i \rangle).$$

Optimality implies that the inequality above is indeed equality. Thus, the optimisation problem gets reduced to (4.2). Reduction of (4.2) to (3.2) is similar. □

This completes our discussion about the equivalence of regularized Hinge loss minimisation and the geometric approach to SVM. We conclude with the following remark.

Remark 7. *There is a hidden issue when we do the transformation described in the begining of Section 2. As seen in optimisation problems (2.2) and (4.2), the last coordinate of vector w is the intercept of the separating hyperplane. As we are penalising the norm of w, we are also penalising the intercept as well. This means (roughly) that we are looking for hyperplanes that are closer to the origin.*

The "correct" way to write the optimisation problem for SVM is to include b (the intercept) and write the problem in terms of b and w, having in mind that b shall not be penalised. Although this is possible, the clarity of our current derivation and little difference in experiments, made us choosing the former approach. The reader is encouraged to derive the latter derivation and see the difference.

5. Support Vectors*

Note. For this section, we need material discussed in the recap slides related to KKT Optimality Conditions. This section is *not* related to the exam and is only here for the interested reader.

We start with the Hard-SVM problem (2.2). Notice that the objective function is convex, and all the conditions are linear inequalities. Thus, the KKT conditions apply. First, we have the stationarity condition:

$$w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0, \tag{5.1}$$

where $\lambda_i \geq 0$. Let us denote by T the set of *tight constraints* of (2.2), i.e.,

$$T = \{i \in \{1, \ldots, n\} \mid y_i \cdot \langle w, x_i \rangle = 1\}.$$

In SVM literature, we refer to these points as *support vectors*. The complementary slackness of KKT theorem states that for $i \notin T$, we have $\lambda_i = 0$. Thus, one can rewrite (5.1) as

$$w - \sum_{i \in T} \lambda_i y_i x_i = 0,$$

or equivalently

$$w = \sum_{i \in T} \lambda_i y_i x_i.$$

We have thus proved the following lemma:

Lemma 8. *The solution to Hard-SVM can be expressed (or represented) as a linear combination of the support vectors, i.e., those points of the dataset which are exactly on the margin.*

The good thing about support vectors is that they can be much smaller than the whole dataset, and predictions (in the dual SVM) will get less expensive.

6. Dual of the SVM Problem*

Let us move a bit further and study the *dual* of the Hard-SVM problem (2.2). The first thing that we do is to put the constraints in the objective function using the following trick. Let us define the function

$$g(w) = \max_{\substack{\alpha \in \mathbb{R}^n \\ \alpha \geq 0}} \sum_{i=1}^{n} \alpha^i (1 - y_i \cdot \langle w, x_i \rangle).$$

This function has a closed form as follows

$$g(w) = \begin{cases} 0 & \text{if } y_i \cdot \langle w, x_i \rangle \geq 1 \text{ for all } i \in \{1, \ldots, n\}, \\ \infty & \text{otherwise.} \end{cases}$$

That is, $g(w) = 0$ if w satisfies all the constraints of (2.2), and is infinity otherwise. Thus, the optimization problem (2.2) is equivalent to the following (unconstrained) optimisation problem

$$\min_w \tfrac{1}{2}\|w\|^2 + g(w),$$

or the following *minimax* problem

$$\min_w \max_{\substack{\alpha \in \mathbb{R}^n \\ \alpha \geq 0}} \tfrac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha^i (1 - y_i \cdot \langle w, x_i \rangle).$$

We note that in this situation we can exchange min and max.[1] Thus, we arrive to the following equivalent problem:

$$\max_{\substack{\alpha\in\mathbb{R}^n\\ \alpha\geq 0}} \min_{w} \tfrac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha^i(1 - y_i \cdot \langle w, x_i\rangle). \tag{6.1}$$

The inner minimisation problem can be solved analytically, and one gets

$$w = \sum_{i=1}^{n} \alpha^i y_i x_i,$$

which is again the result of Lemma 8. Note that for constraints which are not tight, $\alpha^i = 0$. Replacing the value of w in (6.1) gives

$$\max_{\substack{\alpha\in\mathbb{R}^n\\ \alpha\geq 0}} \sum_{i=1}^{n} \alpha^i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha^i\alpha^j y_i y_j \langle x_i, x_j\rangle . \tag{6.2}$$

Note that solving (6.2) only needs the inner products of x_i's. This is helpful in situations where the dimensions of the problem is high (or even infinite!). You will learn in the course that the inner product $\langle x_i, x_j\rangle$ can be replaced with a so-called kernel $k(x_i, x_j)$, which enables one to solve nonlinear separation problems.

Part 2. Feature Selection

We bring three examples of feature selection in ML. The first two examples emphasise on the fact that feature selection is not an obvious procedure, and one needs to be careful in certain situations. The next example focuses on the Lasso regression and gives a picture for one dimensional case. All examples are borrowed from [1].

7. Linear Models

Assume a regression problem, where features (x) are in $\mathbb{R}^2$ and responses (y) are real values. Define the following distribution over the features and responses:

$$\begin{aligned} x^1 &\sim \text{Uniform}(-1, 1)\\ y &= (x^1)^2\\ x^2 &= y + z, \quad \text{where } z \sim \text{Uniform}(-0.01, 0.01). \end{aligned}$$

The question is, "which feature to choose? x^1 or x^2?"

Since the response is *a function of* x^1, one may say that x^1 is the best selected feature. However, in a linear model, x^2 performs much better than x^1 (why?).

8. Importance of a Global Look

An intuitive procedure for feature selection is as follows: for each feature, individually, we assign a score (of how good the feature is), and we pick the features with highest score. While this method seems promising, it can fail dramatically.

For linear regression, as an example, one can define the score of each feature to be the goodness of fit for that single feature, i.e., the empirical squared loss of the prediction using only that feature. It turns out (why?) that this score is proportional to the correlation of the feature and the response: the higher the correlation, the higher the score. Thus, one shall pick the features with highest correlations (see *Pearson's Correlation Coefficient*).

[1] This is called Strong Duality.

The following example shows how this method fails. Consider a linear regression problem on $\mathbb{R}^2$. Define the distribution over features (x) and response (y) as follows:

$$\begin{aligned} y &= x^1 + 2x^2, \\ x^1 &\sim \text{Uniform}(\{\pm 1\}) \\ x^2 &= -\frac{1}{2}x^1 + \frac{1}{2}z, \quad \text{where } z \sim \text{Uniform}(\{\pm 1\}). \end{aligned}$$

Notice that $\mathbb{E}[yx^1] = 0$, meaning that x^1 is not a good predictor alone! But one can see that any good predictor should take x^1 in to account to be able to predict well.

9. LASSO Regularisation

Consider a one dimensional linear regression problem with Lasso regularisation. The data points are $x_1, \dots, x_n \in \mathbb{R}$ and the responses are $y_1, \dots, y_n \in \mathbb{R}$. The problem is equivalent to solving the following optimisation problem

$$\underset{w \in \mathbb{R}}{\text{minimise}} \frac{1}{2n} \sum_{i=1}^{n} (x_i w - y_i)^2 + \lambda |w|, \tag{9.1}$$

where λ is the regularisation parameter. Rewriting the objective function, we get the following equivalent problem

$$\underset{w \in \mathbb{R}}{\text{minimise}} \frac{1}{2} \left(w^2 \cdot \frac{1}{n} \sum_{i=1}^{n} x_i^2 - w \cdot \frac{1}{n} \sum_{i=1}^{n} x_i y_i \right) + \lambda |w|.$$

By normalising the data, we can assume that $\frac{1}{n} \sum x_i^2 = 1$, and get

$$\underset{w \in \mathbb{R}}{\text{minimise}} \frac{1}{2} \left(w^2 - w \cdot \frac{1}{n} \langle x, y \rangle \right) + \lambda |w|.$$

The following lemma then gives us the optimal solution

$$w^\star = \text{sign}(\langle x, y \rangle) \left[\frac{|\langle x, y \rangle|}{n} - \lambda \right]_+,$$

which basically means that if the correlation between x and y is below the threshold λ, the solution is $w^\star = 0$. The reader is encouraged to derive the same solution for Ridge regression and see that there is no thresholding effect in that case.

Lemma 9. *The solution to the optimisation problem*

$$\underset{w \in \mathbb{R}}{\text{minimise}} \frac{1}{2} w^2 - xw + \lambda |w|$$

for $\lambda \geq 0$ is

$$w^\star = \text{sign}(x)[|x| - \lambda]_+,$$

where $[a]_+ = \max\{a, 0\}$.

Proof. A simple proof proceeds with a case by case analysis and is left to the reader. □

References

[1] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.