CIS5200: Machine Learning	Spring 2025
Recitation 5	
Release Date: February 19, 2025	*Due Date: End of Recitation*

1 Understanding the Margin

Given a hyperplane defined by $w^\top x = 0$, what is the shortest distance from a point x_i to this hyperplane? We'll answer this question step by step.

1. For any point x, there is a point x_p that is the perpendicular projection from the hyperplane to the point. Why?

   SOLUTION: The hyperplane is defined by the normal vector w, meaning that any point x can be decomposed into a component along w and a perpendicular component. The perpendicular projection x_p is the closest point on the hyperplane to x, meaning the vector from x_p to x is normal to the hyperplane.

2. Because the vector pointing from x_p to x, $d = x - x_p$, is perpendicular to the hyperplane, it is parallel to w. Why?

   SOLUTION: d is perpendicular to the hyperplane and w is perpendicular to the hyperplane, so they must be parallel.

3. What does this tell us about the relationship between w and d?

   SOLUTION: Since d is parallel to w, we can write $d = \alpha w$ for some scalar α. This tells us that the shortest path from x to the hyperplane lies along the direction of w.

4. Since x_p is on the hyperplane, what can we say about w and x_p?

   SOLUTION: Since x_p lies on the hyperplane, by definition, it satisfies:

   $$w^\top x_p = 0.$$

Now let's prove that this distance is:

$$\|d\| = \frac{|w^\top x_i|}{\|w\|}$$

SOLUTION: From the definition $d = x - x_p$, and since d is parallel to w, we can express:

$$w^\top x_p = 0$$

$$w^\top (x - d) = 0 \Rightarrow w^\top x - w^\top d = 0$$

$$w^\top x = \alpha w^\top w.$$

Solving for α,

$$\alpha = \frac{w^\top x}{\|w\|^2}.$$

Thus, the distance is:

$$\|d\| = |\alpha|\|w\| = \frac{|w^\top x|}{\|w\|}.$$

This is the shortest distance from x to the hyperplane.

2 Deriving the Hard-Margin SVM

2.1 Maximizing the Margin

Suppose we have a linearly separable dataset $\mathcal{D}$ with labels $y_i \in \{-1, 1\}$. Why would we want a margin classifier?

SOLUTION: Maximizing the margin ensures robustness to small perturbations in the data and improves generalization. A large-margin separator minimizes the worst-case classification uncertainty.

Define the margin of the hyperplane w with respect to the entire dataset $\mathcal{D}$.

SOLUTION: The margin of the hyperplane w with respect to the dataset is defined as:

$$\gamma(w, \mathcal{D}) = \min_i \frac{|w^\top x_i|}{\|w\|}$$

What optimization problem arises when we seek to maximize the margin?

SOLUTION: We formulate the problem as:

$$\max_{w,b} \quad \gamma(w, \mathcal{D})$$

subject to the constraint that all points are correctly classified:

$$y_i(w^\top x_i) \geq 1, \quad \forall i.$$

What are some problems with this formulation?

SOLUTION: Doesn't actually fit the data, and (even if we adjust it), it involves maximizing a fraction, which is difficult.

2.2 Rescaling the Problem: Eliminating the Fraction

Observation: The hyperplane equation is scale-invariant. That is, if (w, b) is a valid solution, then for any constant $c \neq 0$, (cw, cb) defines the same hyperplane.

Since we are interested in the relative distances, we can arbitrarily scale w such that:

$$\min_i |w^\top x_i| = 1.$$

With this scaling, our geometric margin now simplifies to:

$$\gamma = \frac{1}{\|w\|}$$

Thus, maximizing the margin γ is equivalent to minimizing $\|w\|$.

2.3 Rewriting the Optimization Problem

Substituting $\gamma = \frac{1}{\|w\|}$, our original problem:

$$\max_{w,b} \gamma$$

is equivalent to:

$$\min_{w,b} \quad \|w\|.$$

Since optimization problems are easier to work with in **quadratic** form, we minimize $\frac{1}{2}\|w\|^2$ instead:

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2$$

subject to the same constraint:

$$y_i(w^\top x_i + b) \geq 1, \quad \forall i.$$

This is the **Hard-Margin SVM Optimization Problem**.

Key Takeaways: - The constraints enforce correct classification. - The objective function encourages a **maximum-margin hyperplane**. - This is a **quadratic program** (QP) that can be solved efficiently.

2.4 Why Support Vectors Matter

What is a support vector, and why does it play a crucial role in SVMs?

Solution: A support vector is a data point that lies exactly on the margin boundary:

$$y_i(w^\top x_i + b) = 1.$$

Why are they important?

- They determine the optimal hyperplane.

- If we remove all other points (except the support vectors), the solution remains unchanged.

- The number of support vectors is often much smaller than the total dataset, making SVMs efficient.

3 Deriving the Soft-Margin SVM

3.1 Introducing Slack Variables

Why do we need the Soft Margin SVM?

Solution: The Hard Margin SVM assumes the data is perfectly linearly separable. If not, there may be no feasible solution. We introduce slack variable ξ_i to allow some classification errors.

How do we modify the constraints to allow violations?

Solution: We introduce a slack variable for every prediction, which allows our prediction to be wrong by some amount:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i.$$

3.2 Reformulated Objective

How does the optimization problem change?

Solution: We now minimize a trade-off between maximizing margin and penalizing violations (minimizing sum of slack vars):

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i$$

subject to:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

Here, C is a hyperparameter controlling the trade-off:

- Large $C \rightarrow$ Prioritizes correct classification (smaller margin).

- Small $C \rightarrow$ Allows some misclassification to maximize margin.

4 ν-SVM (a variant of the Soft Margin SVM)

Recall that in the standard soft-margin SVM, we introduce slack variables ξ_i and a penalty parameter C. In the ν-SVM, instead of C, we use a parameter $\nu \in (0, 1]$ and introduce an additional variable $\rho \geq 0$. Doing so, we obtain the following primal optimization problem:

$$\min_{\mathbf{w}, b, \boldsymbol{\xi}, \rho} \quad \mathbf{w}^\top \mathbf{w} - \nu\rho + \frac{1}{m}\sum_{i=1}^{m} \xi_i$$

subject to

$$\begin{aligned} & y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) \geq \rho - \xi_i, && i = 1, 2, \ldots, m, \\ & \xi_i \geq 0, && i = 1, 2, \ldots, m, \\ & \rho \geq 0 \end{aligned}$$

1. Briefly describe the roles of ρ and ν in this formulation. Describe how this set-up is different from the standard soft margin SVM setting where the penalty parameter is C.

   SOLUTION:
   ρ can be viewed as a margin offset. In the usual SVM, we force the margin to be "1" when deriving the constraints, whereas here ρ is the variable that ends up controlling where the margin boundary lies.

   ν influences the balance between maximizing ρ (thereby increasing margin) and minimizing the sum of slacks $\sum \xi_i$. Crucially, one can show ν serves as an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors.

   So, ν is a "fraction parameter" that makes the fraction of margin violations and the fraction of support vectors more directly controlled than the traditional soft margin SVM approach with C.

2. From the above optimization problem, each training instance must satisfy a modified margin constraint:

   $$y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) \geq \rho - \xi_i$$

   By rearranging these constraints (and using $\rho \geq 0$), you can eliminate the ξ_i variables in favor of a "ν-hinge" loss. Show that at optimality, the slack variable ξ_i can be expressed via

   $$\xi_i = \max\Big\{0, \rho - y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big)\Big\}$$

   and substitute this expression into the objective function to derive a purely unconstrained form in terms of $\mathbf{w}, b$, and ρ.

   SOLUTION:
   From

   $$y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big) \geq \rho - \xi_i \implies \xi_i \geq \rho - y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big)$$

   If $\rho - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0$, the minimum ξ_i satisfying the constraint is 0. If $\rho - y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0$, the minimum ξ_i is $\rho - y_i(\mathbf{w}^\top \mathbf{x}_i + b)$.

So, at optimality:

$$\xi_i = \max\Big\{0, \rho - y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big)\Big\}$$

In the objective, replace each ξ_i with $\max\{0, \rho - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$. The objective becomes:

$$\mathbf{w}^\top \mathbf{w} - \nu\rho + \frac{1}{m}\sum_{i=1}^{m} \max\Big\{0, \rho - y_i\big(\mathbf{w}^\top \mathbf{x}_i + b\big)\Big\}$$

We can typically retain the constraint $\rho \geq 0$, but otherwise the slack constraints $\xi_i \geq 0$ are effectively gone, having been absorbed by the max function.

Note that the term

$$\ell_{\nu\text{-hinge}}\big(\mathbf{x}_i, y_i; \mathbf{w}, b, \rho\big) = \max\Big\{0, \rho - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\Big\}$$

can be viewed as a modified hinge loss. The usual hinge loss is $\max\{0, 1 - y_i \cdot \ldots\}$. Here, the added modification is in ρ.

From the perspective of ERM, the risk to minimize can be broken down as

$$\underbrace{||\mathbf{w}||^2}_{\text{regularizer}} - \nu\rho + \underbrace{\frac{1}{m}\sum_{i=1}^{m} \max\{0, \rho - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}}_{\text{loss}}$$

5 Multiclass SVMs: One-vs-One vs. One-vs-All

How do we extend SVMs to handle multiple classes?

Solution: Since SVMs are inherently binary classifiers, we need to use:

1. One-vs-All (OvA): Train K classifiers, each distinguishing one class vs. all others.

2. One-vs-One (OvO): Train $\frac{K(K-1)}{2}$ classifiers, each distinguishing between two classes.

How does the decision process differ between OvA and OvO?

Solution: - In OvA, the class with the highest confidence score wins. - In OvO, each classifier votes, and the majority class is chosen.

Which approach is better: OvO or OvA?

Solution: - OvO is better for small datasets, since each classifier sees fewer samples. - OvA scales better for large datasets** because it requires training only K classifiers.

6 Recap and Discussion Questions

1. Why does maximizing the margin improve generalization?
2. How does the hinge loss function in SVMs compare to the 0-1 loss?
3. If an SVM has a high number of support vectors, what does that tell us about the dataset?