Support Vector Machine

Tram Nghi Pham

December 1, 2018

Support Vector Machine Overview

1. So far, we have explored **Linear Regression** and **Logistic Regression** for classification
2. In the end, we only care about assigning each point based on "good" decision boundary
3. **Support Vector Machines (SVMs)** are an attempt to model decision boundaries directly

Support Vector Machine Overview

1. So far, we have explored **Linear Regression** and **Logistic Regression** for classification
2. In the end, we only care about assigning each point based on "good" decision boundary
3. **Support Vector Machines (SVMs)** are an attempt to model decision boundaries directly
4. Here is the setup of the problem:
   - Given: training dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in R^d$ and $y_i \in \{-1, +1\}$
   - Goal: find a $d-1$ dimensional **hyperplane** (i.e decision boundary) H which separates the $+1's$ from the $-1's$

Perceptron

1. SVMs are extensions of **perceptron** classifier
2. Given that the training data is **linearly separable**, the perceptron algorithms find a $d - 1$ dimensional hyperplane that perfectly separates the $+1's$ from the $-1's$

Perceptron

1. SVMs are extensions of **perceptron** classifier
2. Given that the training data is **linearly separable**, the perceptron algorithms find a $d-1$ dimensional hyperplane that perfectly separates the $+1's$ from the $-1's$
3. Mathematically, the goal is to learn a weight $w \in R^d$ and a bias term $b \in R$, that satisfy the linear separability constraints:

$$\forall i, = \begin{cases} w^T x_i - b \geq 0 & \text{if } y_i = 1 \\ w^T x_i - b \leq 0 & \text{if } y_i = -1 \end{cases} \tag{1}$$

   Equivalently,

$$\forall i, y_i(w^T x_i - b) \geq 0$$

4. The resulting decision boundary is a hyperplane $H = \{x : w^T x - b = 0\}$

Motivation for SVMs

- Perceptrons have two major shortcomings
  - If data is not linearly separable, the perceptrons fails to find a stable solution
- If the data is linearly separable, the perceptrons could find infinitely many decision boundaries $\implies$ generalization issues

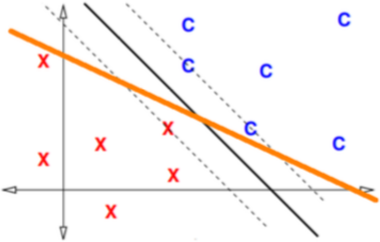

Figure: Two possible decision boundaries under the perceptron. The X's and C's represent the +1's and -1's respectively. Source: UC Berkeley CS189

Hard-Margin SVMs

- **Hard-Margin SVMs** solves the generalization problem of perceptrons by maximizing the margin
- Margin is the minimum distance from the decision boundary to any of the training points
- Variables that needs to be optimized over are the margin m, and the parameters of the hyperplanes, w and b.

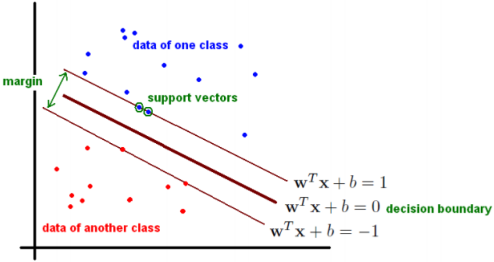

Figure: Support Vector Machine. Source: Zoya Gavrilov's note

Hard-Margin SVMs

- The objective is to maximize the margin m, subject to the following constraints:
  1. All points classified as $+1$ are to the positive side of the hyperplane and their distance to H is greater than the margin
  2. All points classified as -1 are to the negative side of the hyperplane and their distance to H is greater than the margin
  3. The margin is non-negative

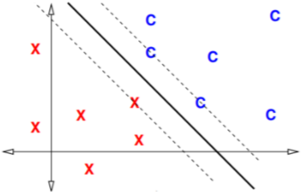

Figure: The optimal decision boundary maximizes the margin. Source: UC Berkeley CS189

Hard-Margin SVMs: Mathematical Formulation

- The linear decision boundary is the hyperplane $H = \{x | w^T x = b\}$ where w is normal to H
- Consider two points x_A, x_B lie on the decision surface. Then

$$y(x_A) = y(x_B) = 0 \implies w^T(x_A - x_B) = 0$$

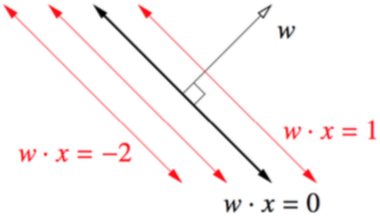

Figure: w is normal to hyperplane. Source: UC Berkeley CS189

Hard-Margin SVMs: Mathematical Formulation

- w is perpendicular to H, the shortest distance from any arbitrary point z to the H is determined by a scaled multiple of w.
- Let x_0 be a point on H, then the distance from z to H is the length of the projection from $z - x_0$ to the vector w, which is

$$D = \frac{|w^T(z - x_0)|}{||w||_2} = \frac{|w^T z - w^T x_0|}{||w||_2} = \frac{|w^T z - b|}{||w||_2}$$

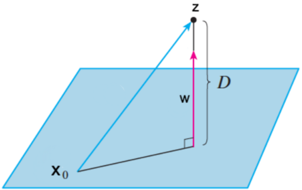

Figure: Shortest distance from z to H is determined by projection of $z - x_0$ onto w. Source: Numerical Analysis by Timothy Sauer

Hard-Margin SVMs: Mathematical Formulation

- The distance from any of the training points x_i to H is
$$\frac{|w^T x_i - b|}{||w||_2}$$
- In order to ensure that positive points are on the positive side of the hyperplane outside a margin of size m, and that negative points are on the negative side of the hyperplane outside a margin of size m, we can express the constraint
$$y_i \frac{|w^T x_i - b|}{||w||_2} \geq m$$

Hard-Margin SVMs: Mathematical Formulation

- The distance from any of the training points x_i to H is

$$\frac{|w^T x_i - b|}{||w||_2}$$

- In order to ensure that positive points are on the positive side of the hyperplane outside a margin of size m, and that negative points are on the negative side of the hyperplane outside a margin of size m, we can express the constraint

$$y_i \frac{|w^T x_i - b|}{||w||_2} \geq m$$

- Putting everything together,

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

$$s.t\ y_i \frac{|w^T x_i - b|}{||w||_2} \geq m, \forall i$$

$$m \geq 0$$

Hard-Margin SVMs: Mathematical Formulation

- The distance between hyperplanes $H_1 : w^T x = a$ and $H_2 : w^T x = c$ is $\frac{|a-c|}{||w||_2}$
- We wish to optimize the margin $\frac{2m}{||w||_2}$.
- Let $m = 1$. Maximizing the margin then corresponds to minimizing $||w||_2$, or more conveniently, $\frac{1}{2}||w||_2^2$

$$\min_{w,b} \frac{1}{2}||w||_2^2$$

$$s.t\ y_i(w^T x_i - b) \geq 1 \forall i$$

- The hyperplane is completely determined by those x_i which lie nearest to it. These points are called *support vectors*

Soft-Margin SVMs

- The hard-margin SVM optimization problem encounters two problems:
  1. It has a unique solution only if the data is linearly seperable, but it has no solution otherwise
  2. It is very sensitive to outliers
- We now consider **Soft-Margin SVMs** which is not sensitive to outliers and can work even in the presence of data that is not linearly separable

Soft-Margin SVMs

- A *soft-margin SVM* modifies the constraints from the hard-margin SVM by allowing some points to violate the margin
- Soft-margin SVM introduces **slack variable** ξ_i, one for each training point so that each x_i need only be a distance of $1 - \xi_i$ from the hyperplane
- The soft-margin SVM optimization problem is

$$\min_{w,b,\xi_i} \frac{1}{2}||w||_2 + C\sum_{i=1}^{n} \xi_i$$

$$s.t\ y_i(w^T x_i - b) \geq 1 - \xi_i.\forall i$$

$$\xi_i \geq 0 \forall i$$

- C is a hyperparameter. A large C keep ξ_i small and vice versa.

- We need to bound value of ξ_i because by setting them too large, then any point may violate the margin by an arbitrarily large distance, which makes choice of w meaningless
- Table below compares the effects of having a large C versus a small C:

	small C	large C
Desire	maximize margin	keep ξ_i's small or zero
Danger	underfitting	overfitting
Outliers	less sensitive	more sensitive

Relation to Logistic Regression

- The solf-margin SVM is an example of *empirical risk minimization (ERM)* algorithm. Regularlized ERM algorithms are a family of learning methods of the form

$$L(y_i, w^T x_i - b) = \min_{w,b} \frac{1}{n} \sum_{i=1}^{n} (y_i, w^T x_i - b)) + \lambda ||w||^2$$

- In the context of classification, we consider 0-1 **step-loss**:

$$L_{step}(y, w^T x - b) = \begin{cases} 1 & w^T x - b < 0 \\ 0 & w^T x - b \geq 0 \end{cases} \tag{2}$$

- 0-1 loss is difficult to optimize since it is neither convex for differentiable

Relation to Logistic Regression

- We begin to modify the 0-1 loss to be convex. The point with $y(w^T x - b) \geq 0$ should remain the same at 0 loss, but we consider allowing a linear penalty "ramp" for misclassified points. This leads to **hinge loss**:

$$L_{hinge}(y, w^T x + b) = max(1 - y(w^T x - b), 0)$$

Figure: Step (0-1) loss, hinge loss, and squared loss. Source: UC Berkeley CS189

Relation to Logistic Regression

- Using the hinge loss, regularized regression becomes

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} max(1 - y_i(w^T x_i - b), 0) + \lambda ||w||^2$$

Relation to Logistic Regression

- Recall the original soft-margin SVM optimization problem is

$$\min_{w,b,\xi_i} \frac{1}{2}||w||_2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t\ y_i(w^T x_i - b) \geq 1 - \xi_i.\forall i$$

$$\xi_i \geq 0\forall i$$

- Manipulating the first constraint, we have

$$\xi_i \geq 1 - y_i(w^T x_i - b)$$

- Combining with the constrains that $\xi_i \geq 0$, we have

$$\xi_i \geq max(1 - y_i(w^T x_i - b), 0)$$

- We can re-write the soft-margin SVM as,

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

$$\text{s.t } \xi_i = max(1 - y_i(w^T x_i - b), 0)$$

Relation to Logistic Regression

- We can re-write the soft-margin SVM as,

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

$$\text{s.t } \xi_i = max(1 - y_i(w^T x_i - b), 0)$$

- Simplifying further, we can remove the constrains

$$\min_{w,b,\xi_i} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} max(1 - y_i(w^T x_i - b), 0)$$

Relation to Logistic Regression

- If we divide by Cn (which does not change the optimal solution of the optimization problem), we can see that this formulation is equivalent to the regularized regression problem, with $\lambda = \frac{1}{2Cn}$

$$\min_{w,b,\xi_i} \lambda||w||^2 + \frac{1}{n}\sum_{i=1}^{n} max(1 - y_i(w^T x_i - b), 0)$$

- From this perspective, SVM is closely related to other fundamental classification algorithms such as regularized least squares and logistic regression. The difference lies in the choice of loss function: square- loss for LS and log-loss for logistic.

References

[1] S. Ross, A First Course in Probability, 6th Ed, Prentice Hall, 2002
[2] UC Berkeley, CS189 Fall 2017
[3] Bishop, Pattern Recognition and Machine Learning