Support Vector Machines

Pramook Khungurn

December 11, 2019

- Three properties make SVM attractive:
  - SVM constructs a **maximum margin separator**. This helps them generalize well.
  - SVM constructs a linear separating plane, but they have the ability to embed the data into a higher-dimensional space, using the **kernel trick**. The high dimensional linear separator is actually non-linear in the input.
  - SVMs are non-parametric model. They retain some inputs, but in practice only a small fraction is stored. So, SVMs combine advantages of parametric and non-parametric models: flexible to represent complex functions, but resistant to overfitting.
- SVM learning follows the following conventions:
  - The class labels are $+1$ and -1 rather than 0 and 1.
  - The separating hyperplane is specified by two components: the weights $\mathbf{w}$ and the bias b. The equation of the plane is
    $$\mathbf{w} \cdot \mathbf{x} + b = 0.$$
- To find the maximum margin separator, we solve the following quadratic program:
  $$\text{maximize } \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j \cdot \mathbf{x}_k)$$
  subjected to contrainst:
  - $\sum_j \alpha_j y_j = 0$,
  - $\alpha_j \geq 0$.
- Once we have the solution to the above quadratic program, we recover
  $$\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j.$$
  However, it's not necessary to do so because we note that
  $$\mathbf{w} \cdot \mathbf{x} = \sum_j \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}).$$
  The hypothesis thus becomes
  $$h(\mathbf{x}) = \text{sign}\left(\sum_j \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}) - b \right),$$
  so we can represent the hypothesis by keeping $\mathbf{x_j}$ such that $\alpha_j \neq 0$ and the weights α_j and b.

- The quadratic program above has two desirable properties:
  - The target function is convex. So there's a global maximum that can be found.
  - The expression only involves dot products of pairs of points.
- The second desirable property has a far-reaching consequence. It allows us to find a linear separator, not only in $\mathbf{x}$, but in another high-dimensional feature space $F(\mathbf{x})$. All we need is a **kernel function**:

  $$K(\mathbf{x}_j, \mathbf{x}_k) = F(\mathbf{x}_j) \cdot F(\mathbf{x}_k).$$

  We only need to know how to compute $F(\mathbf{x}_j) \cdot F(\mathbf{x}_k)$. We don't even need to know what F is!

  This is the **kernel trick**. It allows for an optimal linear separator to be found efficiently in feature spaces with billions of dimensions.