SVM

April 26, 2020

1 Support Vector Machine

1.0.1 Distance beween a point and hyperplane

In N dimensional space, the minimal distance from a point $\{y_i\}$ to the N-1 hyperplane $w_i x_i + b = 0$ can be achieved by the minimal of the following Lagrange function:

$$L = [(x_i - y_i)(x_i - y_i) + \mu(w_i x_i + b)], \tag{1}$$

which is given by

$$\frac{\partial L}{\partial x_j} = [2(x_j - y_j) + \mu w_j] = 0. \tag{2}$$

Substituting $x_i = y_i - \frac{1}{2}\mu w_i$ back to the hyperplane equation, we have

$$0 = w_i(y_i - \frac{1}{2}\mu w_i) + b, \tag{3}$$

$$\mu = \frac{2(b + w_i y_i)}{|w|^2}. \tag{4}$$

Then the minimal distance between y and hyperplane reads:

$$d = \frac{1}{2}|\mu||w| = \frac{|b + w_i y_i|}{|w|}. \tag{5}$$

1.0.2 Objective function

Suppose we have a sample with predictors $\{X^i\}$ and outcome $\{Y^i\}$. Suppose the outcome takes two values ± 1, given feature x, the outcome is predicted according to

$$y = sign(b + w^T x)1. \tag{6}$$

If the outcome is correctly predicted, the distance can also be written as:

$$d = \frac{y(b + w^T x)}{|w|}, \tag{7}$$

with the normal direction of the hyperplane pointing to the Y=+1 class.

The goal of SVM is to maximize the smallest distance among all data points d_{min} by varying $\{w, b\}$. d_{min} is given by:

$$d_{min} = \arg\min_{i}\{d^i\}, \tag{8}$$

with $d^i = \frac{y^i(b+w^T x^i)}{|w|}$. We denote the observation corresponding to d_{min} as x_{min}, then an alternative way to formulate this maximazation process is:

$$\arg\max_{\{w,b\}} \frac{y_{min}(b + w^T x_{min})}{|w|}, \tag{9}$$

$$s.t. \frac{y^i(b + w^T x^i)}{|w|} - \frac{y_{min}(b + w^T x_{min})}{|w|} \geqslant 0. \tag{10}$$

If the data point corresponds to the minimal distance is unchanged during the variation process, we can utilize the scale invariance of $\{w, b\}$ in representing the same hyperplane to make $y_{min}(b + w^T x_{min}) = 1$. Then the problem can be reexpressed as:

$$\arg\min_{\{w,b\}} |w|^2, \tag{11}$$

$$s.t. y^i(b + w^T x^i) - 1 \geqslant 0 \tag{12}$$

$$y_{min}(b + w^T x_{min}) = 1. \tag{13}$$

The problem involves inequality constraints and can be solved through the Karush–Kuhn–Tucker conditions, which will be introduced in the following.

1.0.3 Lagrange multiplier

The Lagrange multiplier method is to solve the following problem:

$$maximize : f(x) \tag{14}$$

$$subject\ to : g(x) = 0. \tag{15}$$

In the two-dimensional example shown by the above picture, we need to find the maximum of $f(x, y)$ on the red line of condition $g(x, y) = 0$. A necessary condition is that the derivative of $f(x, y)$ along the tangent direction of the red line is zero. This condition happens in two cases: (1) $\nabla f = 0$ in regardless of g, (2) ∇f parallel to ∇g. The two cases can be denoted by a single expression:

$$\nabla f = \lambda \nabla g, \tag{16}$$

where λ is called the Lagrange multiplier and equals to zero for the first case. Of course the above equation has to be combined with the feasibility condtion

$$g(x) = 0. \tag{17}$$

Then the two equations can be further combined as the stationary points condition of the Lagrangian $\mathcal{L}(x,\lambda) = f(x) + \lambda g(x)$:

$$\nabla_{x,\lambda}\mathcal{L} = 0, \tag{18}$$

where $\mathcal{L}(x,\lambda)$ is a function depending on extra dimensions denoted by λ.

1.0.4 Karush–Kuhn–Tucker conditions

KKT conditions are generalization of the Lagrange condition to include inequality constraints:

$$maximize : f(x), \tag{19}$$
$$subject\ to : g(x) \leqslant 0 \tag{20}$$
$$h(x) = 0. \tag{21}$$

The idea is based on a simple observation that if the maximum happens on the boundary of $g(x) = 0$, the problem is reduced to the Lagrange problem with additional constraints, else (happens in the domain $g(x) < 0$) the inequality condition can actually be discarded and the problem is reduced to the Lagrange case only with constraint h. Following the Lagrange case, we define the Lagrangian as

$$\mathcal{L} = f(x) + \mu g(x) + \lambda h(x), \tag{22}$$

the two cases can be denoted by the complementary slackness condition:

$$\mu g(x) = 0. \tag{23}$$

Of course the primal feasibility condion:

$$g(x) \leqslant 0, \tag{24}$$

and stationary condition:

$$\nabla_{x,\lambda}\mathcal{L} = 0, \tag{25}$$

should be satisfied.

[1]:
```
import numpy as np
from sklearn import datasets
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import LinearSVC
```

[2]:
```
iris=datasets.load_iris()
x=iris['data'][:,(2,3)] # petal length, petal width
y=(iris['target']==2).astype(np.float64) # Iris virginica
```

[3]:
```
svm_clf=Pipeline([('scalar',StandardScaler()),('linear_svc',LinearSVC(C=1,loss='hinge'))])
```

[4]:
```
svm_clf.fit(x,y)
svm_clf.predict([[5.5,1.7]])
```

[4]:
```
array([1.])
```

[ ]:
```

```