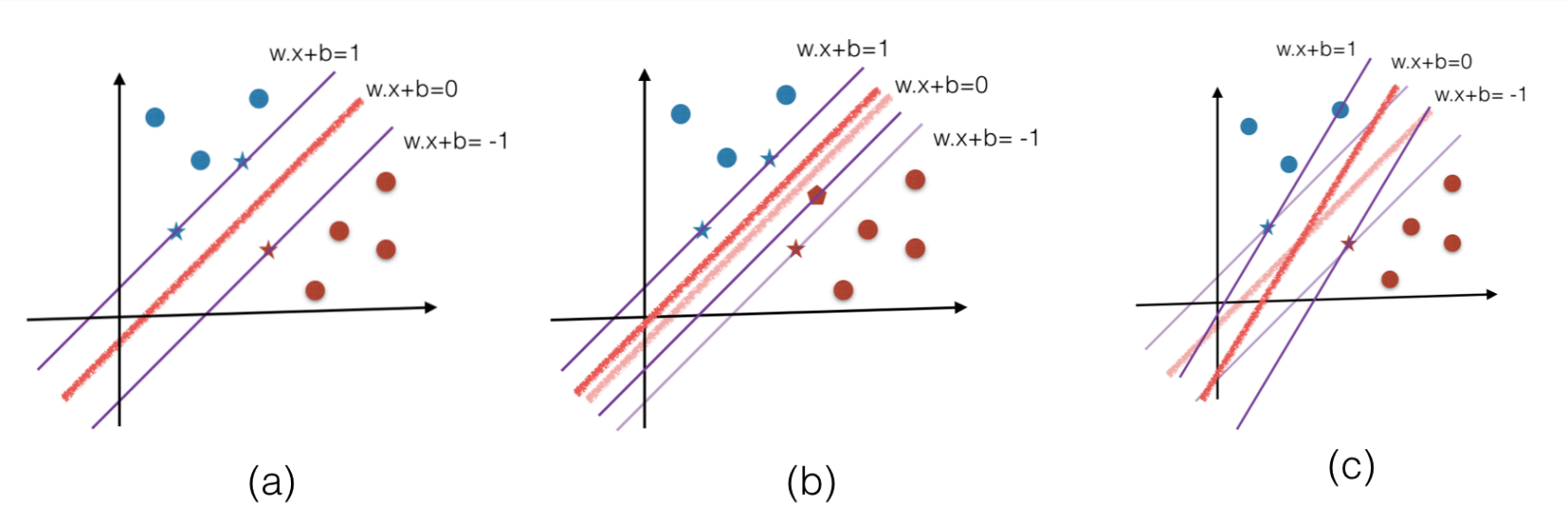
Support Vector Machine

Machine Learning Algorithm in C++

Xinqiao Wei, Minhe Ren

Basic SVM

An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points.



Binary Classification

- Label space: $Y = \{-1, +1\}$
- Feature space: $X \in \mathbb{R}^2$
- Training Set: $D = (x_1, y_1), ..., (x_n, y_n)$
- Family of classifiers: $H = \{h(x) = sign(w^T x + b), w \in \mathbb{R}^2, b \in \mathbb{R}\}$

Equation for hyperplane in $\mathbb{R}^2$ is: $w^T + b = 0$. We need to find the optimum hyperplane, which is the maximum margin separating hyperplane.
Support vectors are the vectors on the boundary:

$$y_j(w^T x_j + b) = 1$$

Dual optimization problem

$$w^{SVM} = \sum_{j=1}^{n} \alpha_j^{SVM} y_j x_j$$

$$b^{SVM} = y_i - (w_{SVM})^T x_j$$

for any $\alpha_j^{SVM} > 0$ and there exist at least one such $\alpha_j^{SVM} > 0$.

$$\alpha^{SVM} = argmin[\sum_{j=1}^{n} \alpha_j - \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_j \alpha_k y_j y_k (x_j^T x_k)]$$

The condition is $\sum_{j=1}^{n} \alpha_j y_j = 0$ and $\alpha_j \geq 0$.
This can be solved by Sequential minimal Optimization.

Dual optimization problem

α is Lagrange multipliers.

$$w^{SVM} = \sum_{j=1}^{n} \alpha_j^{SVM} y_j x_j$$

$$b^{SVM} = y_i - (w_{SVM})^T x_j$$

for any $\alpha_j^{SVM} > 0$ and there exist at least one such $\alpha_j^{SVM} > 0$.

$$\alpha^{SVM} = argmin[\sum_{j=1}^{n} \alpha_j - \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_j \alpha_k y_j y_k (x_j^T x_k)]$$

The condition is $\sum_{j=1}^{n} \alpha_j y_j = 0$ and $0 \leq \alpha_j \leq C$ (C is hyper parameter).
This can be solved by Sequential minimal Optimization.

Sequential minimal optimization

SMO breaks this problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers α_j the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2,the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C$$

$$y_1\alpha_1 + y_2\alpha_2 = k$$

this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. k is the negative of the sum over the rest of terms in the equality constraint, which is fixed in each iteration.

Sequential minimal optimization Algorithm

Here are the algorithm:

The algorithm proceeds as follows:

1. Find a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
2. Pick a second multiplier α_2 and optimize the pair (α_1, α_2).
3. Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence. This is critical for large data sets since there are $n(n-1)/2$ possible choices for α_i and α_j.

Sequential minimal optimization Algorithm

Here are the algorithm:

The algorithm proceeds as follows:

1. Find a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
2. Pick a second multiplier α_2 and optimize the pair (α_1, α_2).
3. Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence. This is critical for large data sets since there are $n(n-1)/2$ possible choices for α_i and α_j.