CISC 371 Class 30

Support Vector Machine – Dual Formulation

Texts: [1] pp. 417–422; [2] pp. 150–153

Main Concepts:

- *Lagrange multipliers add variables to a problem*
- *Lagrange function includes equality constraints*
- *Lagrange equation imposes gradient constraints*
- *Linear equality constraints produce a linear equation*

Sample Problem, Machine Inference: How can we effectively compute an optimal separating hyperplane?

Notation: We have previously used λ as the symbol for a Lagrange multiplier of an inequality constraint. The SVM literature and code almost universally use the symbol α, so we will adopt the current usage to make it easier to read articles and documentation.

Rewriting the primal formulation: To write the dual formulation succinctly, it is useful to gather the various scalars and vectors into matrix-vector form. We will retain the notation for the weight vector $\vec{w}$ and the bias scalar b because these are the optimization arguments.

The constraint on a data vector $\vec{x}_j$ that is imposed by its label is: if the label $y_j = +1$ then the data vector should be on the positive "side" of the classification hyperplane; and if the label $y_j = -1$ then the data vector should be on the negative "side" of the classification hyperplane. These can be combined into the single constraint on data vector $\vec{x}_j$ that is

$$\begin{aligned} y_j(\vec{x}_j \cdot \vec{w} + b) &\geq 1 \\ \left[y_j \vec{x}_j^T\right] \vec{w} + b y_j &\geq 1 \end{aligned} \tag{30.1}$$

Because we usually write the optimization argument as a vector, Equation 30.1 suggests that the data vectors $\vec{x}_i$ be gathered into a matrix. We will write the gathered data vectors as the design matrix

$$X_{m \times n} \stackrel{\text{def}}{=} \begin{bmatrix} \vec{x}_1^T \\ \vec{x}_2^T \\ \vdots \\ \vec{x}_m^T \end{bmatrix} \tag{30.2}$$

We will use two methods for gathering the y_i labels. We will use a vector form and a matrix, which are

$$\vec{y} \stackrel{\text{def}}{=} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \qquad Y \stackrel{\text{def}}{=} \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_m \end{bmatrix} \tag{30.3}$$

Using the design matrix of Equation 30.2, the matrix form for the labels of Equation 30.3, the vector form of the labels in Equation 30.3, and the vector $\vec{1}$ that has each entry equal to 1, the inequality constraints of Equation 30.1 can be combined into a single inequality as

$$[YX]\vec{w} + b\vec{y} \geq \vec{1} \tag{30.4}$$

We can write Equation 30.4 as a level set for the value of zero, which is

$$\vec{1} - [YX]\vec{w} - b\vec{y} \leq \vec{0} \tag{30.5}$$

The objective function for the SVM is

$$f(\vec{w}) = \frac{1}{2}\vec{w}^T\vec{w} \tag{30.6}$$

The primal Lagrange function, from Equation 30.6 and Equation 30.5, is

$$\mathcal{L}(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\vec{w}^T\vec{w} + \vec{\alpha}^T[\vec{1} - YX\vec{w} - b\vec{y}] \tag{30.7}$$

Because the SVM is a constrained convex problem, the KKT conditions require that the minimum occur at a stationary point. To find the stationary point, we need to differentiate Equation 30.7 with respect to $\vec{w}$ and with respect to b, then transpose the 1-forms into vectors. This gives us

$$\begin{aligned} \left[\frac{\partial \mathcal{L}}{\partial \vec{w}}\right]^T &= \vec{w} - X^T Y \vec{\alpha} \\ \left[\frac{d\mathcal{L}}{db}\right]^T &= -\vec{\alpha}^T\vec{y} \end{aligned} \tag{30.8}$$

Setting the derivatives in Equation 30.8 to zero, at the stationary point, gives us

$$\begin{aligned} \vec{w} &= X^T Y \vec{\alpha} \\ \text{such that} \quad \vec{y}^T\vec{\alpha} &= 0 \end{aligned} \tag{30.9}$$

We can write $\vec{w}$ in terms of the data matrix X and the Lagrange multipliers $\vec{\alpha}$, subject to a linear equality constraint. Using the expression for $\vec{w}$ in Equation 30.9, and using the constraint in that equation to eliminate a term, gives us the Lagrange formula of the dual of the SVM problem as

$$\begin{aligned} \mathcal{L}_D(\vec{\alpha}, b) &= \frac{1}{2}\vec{\alpha}^T Y X X^T Y \vec{\alpha} + \vec{\alpha}^T[\vec{1} - Y X X^T Y \vec{\alpha} - b\vec{y}] \\ &= -\frac{1}{2}\vec{\alpha}^T Y X X^T Y \vec{\alpha} + \underline{1}\vec{\alpha} \end{aligned} \tag{30.10}$$

The dual of the SVM problem is to maximize the Lagrange formula of Equation 30.10 subject to: (a) the equality constraint of $\vec{y}^T\vec{\alpha} = 0$ and (b) the KKT condition that $\vec{\alpha} \geq \vec{0}$.

Observations on the dual formulation:

- there is one Lagrange multiplier α_i for each data vector $\vec{x}_i$
- primal formulation was less constrained: $\vec{w}$ was a general weight vector
- dual formulation is more constrained: $\alpha_i \geq 0$ and the sum of all $y_i\alpha_i$ is zero
- support vector: data $\vec{x}_i$ with $\alpha_i > 0$
- non-support vector: data $\vec{x}_i$ with $\alpha_i = 0$, which can be removed from the computation

30.1 Sequential Minimal Optimization: Concept

Prior to 1998, SVM optimization was performed by using the dual Lagrange formula of Equation 30.10 as the objective. John Platt, working at Microsoft Research, developed an algorithm that avoids matrix computation and is roughly linear in the number m of data vectors that are used to training the SVM.

From an initial estimate of the Lagrange multipliers, we can find the corresponding weight vector $\vec{w}$ by using Equation 30.9. Any correctly classified support vector $\vec{x}_i$ can be used to calculate the bias scalar b, using Equation 29.12.

The error for each data vector $\vec{x}_k$, for current estimates of $\vec{w}$ and b, is

$$E_k \stackrel{\text{def}}{=} \vec{w}^T\vec{x}_k + b - y_k \tag{30.11}$$

The concept for sequential minimal optimization, or SMO, is to examine carefully the equality constraint of $\vec{y}^T\vec{\alpha} = 0$. If this is expanded into a summation, it is

$$\sum_{i=1}^{m} y_i \alpha_i = 0 \tag{30.12}$$

A simplified version of the SMO algorithm begin by searching for a Lagrange multiplier α_i that does not satisfy the KKT conditions, usually to within some numerical tolerance. Another multiplier, α_j, is selected by a heuristic; we can think of this as a randomly selected multiplier.

The SMO algorithm works by updating the second multiplier using steepest descent. The calculation is performed using the values E_i and E_j from Equation 30.11. The stepsize is calculated as

$$s_{ij} = \frac{1}{\vec{x}_i \cdot \vec{x}_i + \vec{x}_j \cdot \vec{x}_j - 2\vec{x}_i \cdot \vec{x}_j} = \frac{1}{\|\vec{x}_j - \vec{x}_i\|^2}$$

The update of α_j is similar to the Perceptron Algorithm with a variable stepsize:

$$\alpha_j \leftarrow \alpha_j + s_{ij}(E_j - E_i) \tag{30.13}$$

By holding the other Lagrange multipliers as constant, the updated α_j of Equation 30.13 and the constraint equality of Equation 30.12 provide a unique update to the multiplier α_i that previously violated the KKT conditions.

From the new estimates of α_i and α_j, a new estimate of $\vec{w}$ can be computed directly. The new estimate of b is technically complicated, but in most situations is the mean of the estimates from the data vectors $\vec{x}_i$ and $\vec{x}_j$ that correspond to the updated Lagrange multipliers.

The actual SMO algorithm includes a number of other considerations, such as:

- handling cases where s_{ij} is very large
- satisfying an upper constraint on each multiplier, as $\alpha_k \leq C$
- using an advanced way of finding the dot product of data vectors

References

[1] Hastie T, Tibshirani R, Friedman J: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, 2009

[2] Beck A: Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB. Siam Press, 2014