Machine Learning **Fall 2023**

Lecture 8: Support Vector Machine

Lecturer: Liwei Wang Scribe: Guoyi Shao, Tingyang Zhang, Lijiong Chen, Yuzheng Liu, Yunchang Huang, Haoyang Ye, An

8.1 Support Vector Machine (SVM)

8.1.1 KKT Conditions

Assume $x^*; \lambda^*, \mu^*$ are the optimal solutions of (P) and (D) (which we mentioned in lectures before) respectively, we have:

1. $\nabla_x L(x; \lambda, \mu)|_{x^*;\lambda^*,\mu^*} = 0$
2. $g_i(x^*) \leq 0, h_i(x^*) = 0, \forall i \in [n]$
3. $\lambda_i^* \geq 0, \forall i \in [n]$
4. $\lambda_i^* * g_i(x^*) = 0, \forall i \in [n]$

These four conditioins are called KKT conditions, and KKT conditions are necessary and sufficient conditions.

8.1.2 Support Vector

Recall the Max Margin Classifier:

$$(P) \qquad \min_{w,b} \quad \frac{1}{2}||w||_2^2$$
$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1, \forall i \in [n],$$

and its dual form:

$$(D) \qquad \min_{\lambda} \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i^\top x_j - \sum_i \lambda_i$$
$$s.t. \ \lambda_i \geq 0, \forall i \in [n]$$
$$\sum_i \lambda_i y_i = 0$$

We apply the KKT condition 1 and 4 on it and now we have:

1. $w^* = \sum_i \lambda_i^* y_i x_i$

2. $\lambda_i[y_i(w^\top x_i + b) - 1] = 0, \forall i \in [n]$

$\lambda_i^* > 0$ only if $y_i(w^T x_i + b) - 1 = 0$, thus:

$$w^* = \sum_{i \in I} \lambda_i^* y_i x_i$$

where I is the points nearest to the hyperplane, also known as Support Vector.

8.1.3 Situations without solid constraints

In situations where there's always points can't match the constraints, we get another form of question:

$$\begin{aligned}(P) \qquad \min_{w,b,\epsilon} \quad & \frac{1}{2}||w||_2^2 + c * \sum_{i=1}^{n} \epsilon_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \epsilon_i, \forall i \in [n], \\ & \epsilon_i \geq 0, \forall i \in [n]\end{aligned}$$

and its dual form(see proof in Appendix A):

$$\begin{aligned}(D) \qquad & \min_{\lambda} \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i^\top x_j - \sum_i \lambda_i \\ & s.t.\ c \geq \lambda_i \geq 0, \forall i \in [n] \\ & \sum_i \lambda_i y_i = 0\end{aligned}$$

(P) can be rewritten as the following form:

$$(P) \qquad \min_{w,b} \quad \frac{1}{2}||w||_2^2 + c * \sum_{i=1}^{n} [1 - y_i(w^T x_i + b)]_+$$

where $[]_+$ denotes:

$$[u]_+ = \begin{cases} u & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Here the former term can be viewed as the regularization term, and the latter term can be viewed as hinge loss, which means $[1 - yf(x)]_+$, where y denotes the label and $f(x)$ denotes the predict result.

8.1.4 Expand the dim of data

Suppose we have four 2-d data points with their labels as follows:

$$((1,1),1) \qquad ((1,-1),-1) \qquad ((-1,1),-1) \qquad ((-1,-1),1)$$

This is actually the famous XOR example, and apparently these four data points are not linear separable.

However, by expanding the dim of data, we transform each 2-d data point $x = (x^1, x^2)$ into $\phi(x) = ((x^1)^2, (x^2)^2, x^1x^2, x^1, x^2)$, and these four 5-d data points are actually linear separable, which means the original data can be separated by a quadratic curve.

8.2 Bootstrap-Aggregation (Bagging)

Bootstrap-Aggregation is an integration technique to train a classifier.

Suppose that we have a dataset consisting of n samples, one choice is to train a classifier directly on the dataset. But the classifier may be weak.

Following Bootstrap-Aggregation, we can draw n samples from the original dataset with replacement, and combine them into a new dataset. After m times of the same operation, we can get m new datasets containing n samples each, on which we train m weak classifiers respectively. Finally we integrate the m weak classifiers to get a more powerful classifier.

It is worth mentioning that Bootstrap-Aggregation is not Boosting. Instead, Boosting was proposed one year later than Bootstrap-Aggregation and surpassed it in a large margin.

Appendix A

We refer $\lambda_i \geq 0$ and $\mu_i \geq 0$ as the Lagrange multiplier associated with $1 - \epsilon_i - y_i(w^T x_i + b)$ and $-\epsilon_i$, and we define the Lagrangian L (assume w's dimension is m): $\mathbf{R}^m \times \mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$ as

$$
\begin{aligned}
&L(w, b, \epsilon, \lambda, \mu) \\
&= \frac{1}{2}\|w\|^2 + C \cdot \sum_{i=1}^{n} \epsilon_i + \sum_{i=1}^{n} \lambda_i (1 - \epsilon_i - y_i(w^T x_i + b)) + \sum_{i=1}^{n} \mu_i(-\epsilon_i) \\
&= \frac{1}{2}\|w\|^2 - w^T \sum_{i=1}^{n} \lambda_i y_i x_i + \sum_{i=1}^{n} (C - \lambda_i - \mu_i)\epsilon_i + \sum_{i=1}^{n} \lambda_i - b \sum_{i=1}^{n} \lambda_i y_i \ ,
\end{aligned}
$$

then we have:

$$
\begin{aligned}
g(\lambda, \mu) &= \inf_{w, b, \epsilon} L(w, b, \epsilon, \lambda, \mu) \\
&= \begin{cases} -\dfrac{1}{2}\|\sum_{i=1}^{n} \lambda_i y_i x_i\|^2 + \sum_{i=1}^{n} \lambda_i, & \lambda_i + \mu_i = C \wedge \sum_{i=1}^{n} \lambda_i y_i = 0 \\ -\infty, & else \end{cases} .
\end{aligned}
$$

Thus, the corresponding Lagrange dual problem is:

$$
\begin{aligned}
&\max_{\lambda} -\frac{1}{2}\|\sum_{i=1}^{n} \lambda_i y_i x_i\|^2 + \sum_{i=1}^{n} \lambda_i \\
&\text{s.t.} \quad \sum_{i=1}^{n} \lambda_i y_i = 0 \ , \\
&\qquad\ \ 0 \leq \lambda_i \leq C \ , i = 0, 1, .., n \ .
\end{aligned}
$$

References

[AGM97] N. Alon, Z. Galil and O. Margalit, On the Exponent of the All Pairs Shortest Path Problem, *Journal of Computer and System Sciences* **54** (1997), pp. 255–262.

[F76] M. L. Fredman, New Bounds on the Complexity of the Shortest Path Problem, *SIAM Journal on Computing* **5** (1976), pp. 83-89.