Machine Learning Linear classification and SVM

Lecture Note 13: Linear Classification and SVM

Professor: Zhihua Zhang *Scribes: Yanfeng Chen ,Yao Cheng , Tianfan Fu*

1 Generative Model

1.1 Recap

In the last lecture, we discussed about generative model and gave an example

$$p(x, y|\boldsymbol{\theta}) = p(y|\boldsymbol{\pi})p(x|y)$$

Where $\mathbf{y}$ satisfies Bernoulli Distribution

$$p(y|\boldsymbol{\pi}) = \boldsymbol{\pi}^y(1-\boldsymbol{\pi})^{(1-y)}$$

The p-dimensional vector $\mathbf{x}$ belongs to Gaussian component $\mathbf{y}$.

$$p(\mathbf{x}|\mathbf{y}=1) = N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$p(\mathbf{x}|\mathbf{y}=0) = N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$$

Where

$$\boldsymbol{\Sigma} = \mathbf{diag}(\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}...\boldsymbol{\sigma_p})$$

Which means the independency among p dimensions given $\mathbf{y}$. Thus,

$$p(\mathbf{x}|\mathbf{y}=1) = \prod_{i=1}^{p} p(\mathbf{x_i}|\mathbf{y}=1)$$

And this is called Naive Bayes Model.

1.2 Multinomial Distribution

To make the standard Naive Bayes Model, we illustrate a model where variables satisfy discrete distribution compared to the previous continuous distribution. x_j is a multinomial random variable which satisfies that

$$p(\mathbf{y}|\boldsymbol{\pi}) = \boldsymbol{\pi_k}$$

$$\boldsymbol{\pi_k} = p(\mathbf{y}=k|\boldsymbol{\pi})$$

$$p(\mathbf{y}|\boldsymbol{\pi}) = \prod_{i=1}^{c} \boldsymbol{\pi_i}^{\mathbf{y_i}},$$

Where $\mathbf{y} \in \{1, 2...c\}$
To write it in polynomial form, we let $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2}...\mathbf{y_c})^{\mathbf{T}}$ .Then

$$p(x_1, x_2...x_p|\mathbf{y}) = \prod_{j=1}^{p} p(x_j|\mathbf{y})$$

$$p(x_1, x_2...x_p|\mathbf{y_i} = 1) = \prod_{j=1}^{p} p(x_j|\mathbf{y_i} = \mathbf{1}) = \prod_{j=1}^{p}\prod_{k=1}^{m} \eta_{ijk}^{x_j^k}$$

where $\mathbf{y_i} = 1$ means the $\mathbf{x}$ belongs to the i^{th} class. In the multi-classification problem, $x \in \{1, 2...m\}$.
The parameters are

$$\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\eta}\}$$

where

$$\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_c\},\ \boldsymbol{\eta} = \{\boldsymbol{\eta}_{ijk}\}$$

Our target is to obtain the posterior probability to make decision. The posterior probability is

$$p(\widehat{y_j}|\mathbf{x}, \boldsymbol{\theta})\ where\ j \in \{1, 2...c\}$$

Then according to Bayesian formula

$$\mathbf{p(y|x)} = \frac{\mathbf{p(x|y)p(y)}}{\mathbf{p(x)}} = \frac{\mathbf{p(x|y)p(y)}}{\boldsymbol{\Sigma}_\mathbf{y}\mathbf{p(x,y)}}$$

$$\begin{aligned}
p(\widehat{y_l}|\mathbf{x}, \boldsymbol{\theta}) &= \frac{\pi_l \prod_{j=1}^{p}\prod_{k=1}^{m} \eta_{ljk}^{x_j^k}}{\Sigma_{l=1}^{c}\pi_l \prod_{j=1}^{p}\prod_{k=1}^{m} \eta_{ljk}^{x_j^k}} \\
&= \frac{exp(ln\ \pi_l \prod_{j=1}^{p}\prod_{k=1}^{m} \eta_{ljk}^{x_j^k})}{\Sigma_{l=1}^{c}exp(ln\ \pi_l \prod_{j=1}^{p}\prod_{k=1}^{m} \eta_{ljk}^{x_j^k})} \\
&= \frac{exp(ln\ \pi_l + \Sigma_{j=1}^{p}\Sigma_{k=1}^{m}x_j^k ln\ \eta_{ljk})}{\Sigma_{l=1}^{c}exp(ln\ \pi_l + \Sigma_{j=1}^{p}\Sigma_{k=1}^{m}x_j^k ln\ \eta_{ljk})}
\end{aligned}$$

Let $\mathbf{x}_j = \mathbf{x}_j^k$,$\boldsymbol{\theta}_{lj} = ln\boldsymbol{\eta}_{ljk}$,$\boldsymbol{\beta}_l = ln\boldsymbol{\pi}_l$.Then

$$\begin{aligned} & p(\widehat{y_l}|\boldsymbol{x}, \boldsymbol{\theta}) \\ &= \frac{\boldsymbol{exp}(\mathbf{x^T}\boldsymbol{\theta}_l + \boldsymbol{\beta}_l)}{\Sigma_{l=1}^c \boldsymbol{exp}(\mathbf{x^T}\boldsymbol{\theta}_l + \boldsymbol{\beta}_l)} \end{aligned}$$

Since $\Sigma_{l=1}^c \boldsymbol{exp}(\mathbf{x^T}\boldsymbol{\theta}_l + \boldsymbol{\beta}_l)$ is a constant, we only need to maximize $\boldsymbol{exp}(\mathbf{x^T}\boldsymbol{\theta}_l + \boldsymbol{\beta}_l)$. Then we use the MLE(Maximum Likelihood Estimation) to estimate the parameters. Given a training set $D = \{(x_1, y_1), (x_2, y_2)......(x_n, y_n)\} \subset \mathbb{R}^p \times \{0, 1\}^c$,the likelihood function is

$$\begin{aligned} L(\boldsymbol{\theta}|D) &= log\ p(\mathbf{D}|\boldsymbol{\theta}) \\ &= log\ \prod_{i=1}^n p(x_i, y_i|\boldsymbol{\theta}) \\ &= log\ \prod_{i=1}^n p(y_i|\boldsymbol{\pi})p(x_i|\mathbf{y_i}, \boldsymbol{\eta}) \\ &= \Sigma_{i=1}^n[log\ p(y_i|\boldsymbol{\pi}) + log\ p(x_i|\mathbf{y_i}, \boldsymbol{\eta})] \end{aligned}$$

First we estimate $\boldsymbol{\pi}$, it is only related to the first term of the log-likelihood function. Let it be $f(\boldsymbol{\pi})$,then

$$\begin{aligned} f(\boldsymbol{\pi}) &= \Sigma_{i=1}^n log\ p(y_i|\boldsymbol{\pi}) \\ &= \Sigma_{i=1}^n log\ \prod_{l=1}^c \boldsymbol{\pi}_l^{y_{il}} \\ &= \Sigma_{i=1}^n \Sigma_{l=1}^c y_{il} log\ \boldsymbol{\pi}_l \\ &= \Sigma_{l=1}^c \Sigma_{i=1}^n y_{il} log\ \boldsymbol{\pi}_l \end{aligned}$$

Since $\Sigma_{l=1}^c \boldsymbol{\pi}_l = 1$, add the Lagrange Multiplier and obtain that

$$\widehat{l}(\boldsymbol{\pi}) = f(\boldsymbol{\pi}) + \lambda(\Sigma_{l=1}^c \boldsymbol{\pi}_l - 1)$$

Take derivative towards $\boldsymbol{\pi}_l$, we can get

$$\widehat{\boldsymbol{\pi}}_l = \frac{\Sigma_{i=1}^n y_{il}}{\Sigma_{i=1}^n \Sigma_{l=1}^c y_{il}} = \frac{\Sigma_{i=1}^n y_{il}}{\Sigma_{l=1}^c \Sigma_{i=1}^n y_{il}}$$

Then we estimate $\boldsymbol{\eta}$.We have

$$\begin{aligned} \widehat{l}(\boldsymbol{\eta}) &= \Sigma_{i=1}^n log\ p(x_i|\mathbf{y_i}, \boldsymbol{\eta}) \\ &= \Sigma_{i=1}^n \Sigma_{j=1}^p log\ p(x_{ij}|\mathbf{y_j}, \boldsymbol{\eta}) \\ &= \Sigma_{i=1}^n log\ \prod_{l=1}^c (\prod_{j=1}^p \prod_{k=1}^m \eta_{ljk}^{x_{ij}^k})^{y_{il}} \end{aligned}$$

Where

$$x_j = (x_{j1}, x_{j2}...x_{jp})^T$$

Since $\Sigma_{k=1}^{m}\boldsymbol{\eta}_{ijk} = 1$,add the Lagrange Multiplier and obtain that

$$\widehat{l}(\boldsymbol{\eta}) = \Sigma_{i=1}^{n}\Sigma_{l=1}^{c}y_{il}\Sigma_{j=1}^{p}\Sigma_{k=1}^{m}x_{ij}^{k}log\ \boldsymbol{\eta}_{ljk} + \Sigma_{i,j}\lambda_{ij}(\Sigma_{k=1}^{m}\boldsymbol{\eta}_{ijk} - 1)$$

Take derivative towards $\boldsymbol{\eta}_{ljk}$,we get

$$\frac{\partial\widehat{l}(\boldsymbol{\eta})}{\partial\boldsymbol{\eta}_{ljk}} = \lambda_{lj} + \Sigma_{i=1}^{n}\frac{y_{il}x_{ij}^{k}}{\eta_{ljk}}$$

Let $\frac{\partial\widehat{l}(\boldsymbol{\eta})}{\partial\boldsymbol{\eta}_{ljk}} = 0$,we have

$$\eta_{ljk}\lambda_{lj} + \Sigma_{i=1}^{n}y_{il}x_{ij}^{k} = 0$$

$$\Sigma_{k=1}^{m}\eta_{ljk}\lambda_{lj} + \Sigma_{k=1}\Sigma_{i=1}^{n}y_{il}x_{ij}^{k} = 0$$

Thus,

$$\lambda_{lj} = -\Sigma_{k=1}\Sigma_{i=1}^{n}y_{il}x_{ij}^{k}$$

Substitute λ_{lj} with the above formula, we can get

$$\eta_{ljk} = -\frac{\Sigma_{i=1}^{n}y_{il}x_{ij}^{k}}{\lambda_{lj}} = \frac{\Sigma_{i=1}^{n}y_{il}x_{ij}^{k}}{\Sigma_{k=1}\Sigma_{i=1}^{n}y_{il}x_{ij}^{k}}$$

The probability $\mathbf{p}(\mathbf{y}|\mathbf{x})$ has the form $f(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x})exp(\Sigma_{i=1}^{s}\eta_i(\boldsymbol{\theta})T_i(\mathbf{x}) - A(\boldsymbol{\theta}))$.Thus it belongs to exponential family and such model is called Generalized Linear Model.

2 Support Vector Machine

2.1 Introduction

Support Vector Machines (SVMs) are supervised learning models used for classification and regression analysis. The original SVM algorithm was invented by Vladimir N. Vapnik and the current standard incarnation (soft margin) was proposed by Corinna Cortes and Vapnik in 1993.

Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines. It was invented by John Platt in 1998. SMO is widely used for training support vector machines.

Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

2.2 Linear Separable

Firstly, we consider the case that the training sets are linearly separable in the space. Given some training data D, a set of n points of the form

$$D = \{(x, y)|x \in R^p, y \in \{-1, 1\}\}$$

where x is a p-dimensional real vector and y is either 1 or -1, indicating the class to which the point x belongs.
Any hyperplane can be written as the set of points **x** satisfying

$$\mathbf{w} \cdot \mathbf{x} - \mathbf{b} = \mathbf{0},$$

If the training data are linearly separable, we can select two hyper-planes in a way that they separate the data and there are no points between them, and then try to maximize their distance.These hyper-planes can be described by the equation

$$|w^T x + b| = 1$$

The distance from x to the hyperspace can be expressed as

$$Distance = \frac{|w^T x + b|}{||w||}$$

As we also have to prevent data points from falling into the margin, that means

$$min_{(x,y) \in D)} |w^T x + b| = 1$$

That is to say, our target object can be transformed into:

$$min_{(x,y) \in D)} \frac{|w^T x + b|}{||w||} = \frac{1}{||w||}$$

The following two targets are equivalent:

$$max \ \frac{1}{||w||}$$
$$min ||w||^2$$

To maximize the distance, i.e. maximize $\frac{1}{||w||}$, we equivalently consider minimize $||w||^2$.
In linear separable case, all data samples x satisfies that

$$w^T x + b \geqslant 1, when \ y = 1$$
$$w^T x + b \leqslant -1, when \ y = -1$$

This can be rewritten as:

$$y_i(w^T x_i + b) \geqslant 1, i = 1, ..., n$$

Put this together, we get the optimization problem:

$$min_{w,b}\frac{1}{2}\|w\|^2$$
$$s.t.\ y_i(w^Tx_i+b)\geqslant 1, i=1,...,n$$

Hence we get the linear classifier:

$$\widehat{y}=sgn(w^T\widehat{x}+b)$$

Recall the Convex Optimization, two necessary condition must be satisfied:
(1)the objective function must be convex.
(2)the constrain condition must be a convex set.
To solve the optimization problem, we introduce the Lagrange multiplier:

$$\begin{aligned}&L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\Sigma_{i=1}^n\alpha_i(y_i(w^Tx_i+b)-1)\\&s.t.\ \alpha_i\geqslant 0\end{aligned}$$

Take derivation:

$$\frac{\partial L}{\partial w}=w-\Sigma_{i=1}^n\alpha_iy_ix_i=0$$
$$\frac{\partial L}{\partial b}=\Sigma_{i=1}^n\alpha_iy_i=0$$

The "stationary" Karush-Kuhn-Tucker condition implies that the solution can be expressed as a linear combination of the training vectors

$$w=\Sigma_{i=1}^n\alpha_iy_ix_i$$

Complementary conditions:

$$\forall i, \alpha_i[y_i(w^Tx_i+b)-1]=0$$
$$i.e.\ \ \alpha_i=0\bigvee y_i(w^Tx_i+b)-1=0$$

Only a few α_i will be greater than zero. The corresponding $\mathbf{x_i}$ are exactly the **support vectors**, which lie on the margin and satisfy $y_i(\mathbf{w}\cdot\mathbf{x_i}-\mathbf{b})=\mathbf{1}$.
There is always **gap** between primal problem and dual problem.

Dual problem:
Writing the classification rule in its unconstrained dual form reveals that the maximum-margin hyperplane and therefore the classification task is only a function of the support vectors, the subset of the training data that lie on the margin. Using the fact that $\|\mathbf{w}\|^2=\mathbf{w}\cdot\mathbf{w}$ and substituting $\mathbf{w}=\sum_{i=1}^n\alpha_iy_i\mathbf{x_i}$, we introduce the Lagrange multiplier:

$$\begin{aligned}L&=\frac{1}{2}\|\Sigma_{i=1}^n\alpha_iy_ix_i\|-\Sigma_{i,j=1}^n\alpha_i\alpha_jy_iy_jx_i^Tx_j-\Sigma_{i=1}^n\alpha_iy_ib+\Sigma_{i=1}^n\alpha_j\\&=-\frac{1}{2}\Sigma_{i,j=1}^n\alpha_i\alpha_jy_iy_jx_i^Tx_j+\Sigma_{j=1}^n\alpha_j\end{aligned}$$

One can show that the dual of the SVM reduces to the following optimization problem:

$$\begin{aligned} & max_\alpha \ \Sigma_{i=1}^n \alpha_i - \frac{1}{2} \Sigma_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ & s.t. \ \alpha_i \geqslant 0 \bigwedge \Sigma_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

Now we adopt quadratic linear programming to solve the problem.

$$\begin{aligned} h(x) &= sgn(w^T x + b) \\ &= sgn(\Sigma_{i=1}^n \alpha_i y_i x_i^T x_i + b) \\ &= sgn(\Sigma_{i=1}^n \alpha_i y_i K(x_i, x) + b) \end{aligned}$$

$$\forall \ i, \ y_i(w^T x_i + b) = 1$$

That is,

$$\begin{aligned} y_i &= w^T x_i + b \\ b &= y_i - w^T x_i \\ b &= \frac{1}{|S|} \Sigma_{x_i \in S} \ (y_i - w^T x_i) \end{aligned}$$

Note that we can also rewrite by $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j$.

2.3 Linear Inseparable

No classification can guarantee the 100% accuracy, so are the SVM. In some cases,

$$y_i(w^T x_i + b) \not\geqslant 1$$

We can weaken the constrain, let

$$y_i(w^T x_i + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0$$

Here we call ξ_i slack parameter. The objective function can be expressed as

$$min \ \frac{1}{2} \|w\|^2 + c \ \Sigma_{i=1}^n (1 - y_i(w^T x_i + b))^p, \ p \geqslant 1$$

where c is the tradeoff and we usually take $p = 1$ or $p = 2$.
The optimization problem becomes:

$$\begin{aligned} & min_{w,b} \ \frac{1}{2} \|w\|^2 + c \ \Sigma_{i=1}^n {\xi_i}^p \\ & s.t. \ y_i(w^T x_i + b) \geqslant 1 - \xi_i, \ \xi_i \geqslant 0 \end{aligned}$$

Let $\lambda = \frac{1}{c}$, we get equivalent loss function with penalty as the second term:

$$\Sigma_{i=1}^{n}(1 - y_i(w^T x_i + b))_+ + \frac{1}{2}\lambda\|w\|^2$$

where $z_+ = max\{0, z\}$ and it is called hinge loss. Take p=1, then the corresponding Lagrange multiplier:

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + c\Sigma_{i=1}^{n}\xi_i - \Sigma_{i=1}^{n}\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] - \Sigma_{i=1}^{n}\beta_i\xi_i$$

Take derivation:

$$\begin{aligned}
\frac{\partial L}{\partial w} &= w - \Sigma_{i=1}^{n}\alpha_i y_i x_i = 0 \\
\frac{\partial L}{\partial b} &= \Sigma_{i=1}^{n}\alpha_i y_i = 0 \\
\frac{\partial L}{\partial \xi_i} &= c - \alpha_i - \beta_i = 0
\end{aligned}$$

We get

$$\begin{aligned}
&\alpha_i + \beta_i = c \\
&w = \Sigma_{i=1}^{n}\alpha_i y_i x_i \\
&\Sigma_{i=1}^{n}\alpha_i y_i = 0
\end{aligned}$$

$$\forall\ i,\ \ \alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] = 0$$

hence

$$y_i(w^T x_i + b) = 1 - \xi_i \bigvee \alpha_i = 0$$

$$\forall\ i,\ \ \beta_i\xi_i = 0$$

hence

$$\beta_i = 0 \bigvee \xi_i = 0$$

Dual Problem:

$$\begin{aligned}
&max_\alpha\ \Sigma_{i=1}^{n}\alpha_i - \frac{1}{2}\Sigma_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \\
&s.t.\ 0 \leqslant \alpha_i \leqslant c \bigwedge \Sigma_{i=1}^{n}\alpha_i y_i = 0
\end{aligned}$$

The key advantage of a linear penalty function is that the slack variables vanish from the dual problem, with the constant C appearing only as an additional constraint on the Lagrange multipliers.