18 线性模型2

18.1 本集内容简介

前面通过，最大化。

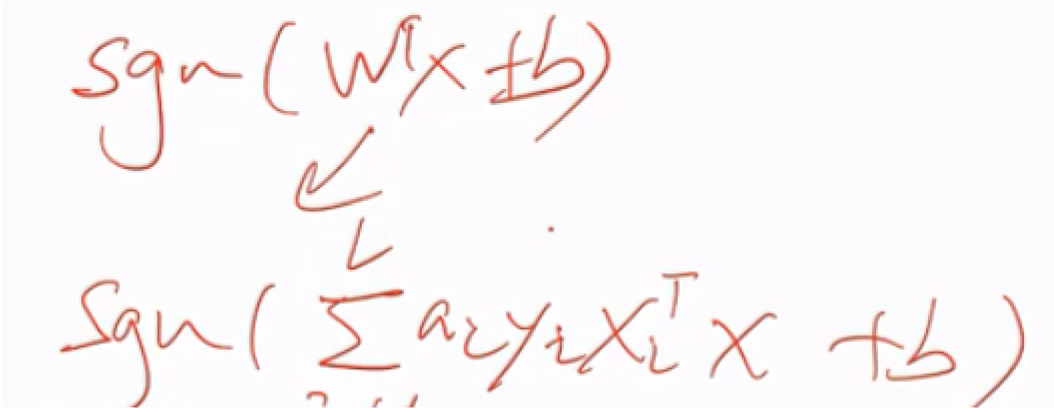
L1-loss SVC

这里就不用考虑核函数了

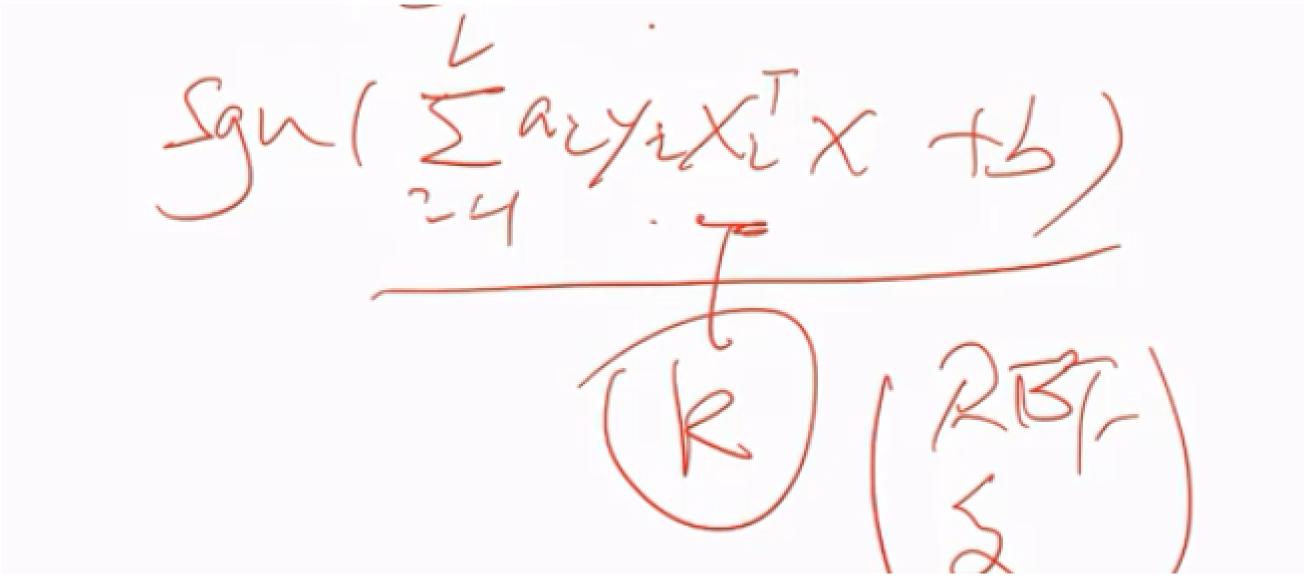
18.2 线性支持向量机简介

线性支持向量机
使用非线性核的支持向量机实际应用中存在的问题
线性支持向量机的优点，泛化性能，速度

如果不带有核函数，就是符号函数



如果是核函数，那就用k带代替，就变为非线性模型



如果数据量达到亿，那么用非线性模型是非常慢的，所以一般用线性模型

18.3 L2正则化L1-loss原问题

第一种，它的损失函数，前面是正则化，

$$1-y_i \mathbf{w}^T \mathbf{x}_i \le 0 \qquad y_i \mathbf{w}^T \mathbf{x}_i \ge 1 \qquad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

如前面讲的支持向量机。

用换元法换下去

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\left(\max\left(0, 1-y_i\mathbf{w}^T\mathbf{x}_i\right.\right.$$

hinge loss，合页损失函数

$$1-y_i \mathbf{w}^T \mathbf{x}_i \le 0 \qquad y_i \mathbf{w}^T \mathbf{x}_i \ge 1$$

$$\xi_i = \max\left(0, 1-y_i\mathbf{w}^T\mathbf{x}_i\right)$$

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

L2正则化L1-loss SVC原问题

$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \left(\max\left(0, 1 - y_i w^T x_i\right) \right)$$

hinge loss，合页损失函数

$1 - y_i w^T x_i \le 0 \qquad y_i w^T x_i \ge 1$

$$\xi_i = \max\left(0, 1 - y_i w^T x_i\right)$$

$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$\xi_i = \max\left(0, 1 - y_i w^T x_i\right) \qquad \xi \ge 0$

$\xi_i \ge 1 - y_i w^T x_i$

18.4 L2正则化L2-loss原问题

L2正则化L2-loss SVC原问题

$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \left(\max\left(0, 1 - y_i w^T x_i\right) \right)^2$$

可以用可信域牛顿法求解，也可以转化成对偶问题求解

可行域牛顿法

18.5 L2正则化对偶问题

不管L1,l2,都可以通过拉格朗日对偶，

我们并没有用梯度下降法，而是坐标下降法，这东西张开就是二次函数，还要考虑吧这样的。

L2正则化SVC对偶问题

$$\min_{\alpha} \frac{1}{2}\alpha^{T}\overline{Q}\alpha - e^{T}\alpha$$

$$0 \le \alpha_i \le U, i = 1, \dots, l$$

可以用坐标下降法求解

$$\min_{z} D_{SVM}(\alpha_1, \dots, \alpha_i + z, \dots, \alpha_l) = \frac{1}{2} Q_{ii} z^2 + \nabla_i D_{SVM}(\alpha) z + c$$

$$0 \le \alpha_i + z \le C$$

$$\nabla_i D_{SVM}(\alpha) = (Q\alpha)_i - 1 = \sum_{j=1}^{l} Q_{ij}\alpha_j - 1$$

可以直接求公式解，但要考虑不等式约束

18.6 L1正则化L2-loss原问题

介绍下一种，

正则化 原问题，jd托分类，都是对于这个问题，我可以养牛顿法来求解。

L1正则化L2-loss SVC原问题

$$\min_{\mathrm{w}} \|\mathrm{w}\|_1 + C\sum_{i=1}^{l}\left(\max\left(0, 1 - y_i \mathrm{w}^T \mathrm{x}_i\right)\right)^2$$

可以用坐标下降法求解

w_j

$$f\left(\mathrm{w} + z\mathrm{e}_j\right) - f\left(\mathrm{w}\right)$$

$$= \left|\mathrm{w}_j + z\right| - \left|\mathrm{w}_j\right| + C\sum_{i \in I\left(\mathrm{w}+z\mathrm{e}_j\right)} b_i\left(\mathrm{w} + z\mathrm{e}_j\right)^2 - C\sum_{i \in I(\mathrm{w})} b_i\left(\mathrm{w}\right)^2$$

$$= \left|\mathrm{w}_j + z\right| + L_j\left(z, \mathrm{w}\right) + c$$

$$\approx \left|\mathrm{w}_j + z\right| + L_j'\left(0, \mathrm{w}\right)z + \frac{1}{2}L_j''\left(0, \mathrm{w}\right)z^2 + c$$

子问题可以采用牛顿法求解

18.7 多类线性支持向量机、

多类线性支持向量机

k 1-1

$$\min_{\mathrm{w}_m,\xi_i} \frac{1}{2}\sum_{m=1}^{k}\mathrm{w}_m^{\mathrm{T}}\mathrm{w}_m + C\sum_{i=1}^{l}\xi_i$$

$$\mathrm{w}_{y_i}^{\mathrm{T}}\mathrm{x}_i - \mathrm{w}_m^{\mathrm{T}}\mathrm{x}_i \ge e_i^m - \xi_i, i=1,\dots,l \qquad e_i^m = \begin{cases} 0 & y_i = m \\ 1 & y_i \ne m \end{cases}$$

$$\arg\max_{m=1,\dots,k} \mathrm{w}_m^{\mathrm{T}}\mathrm{x}$$

用拉格朗日对偶转化

对偶问题

$$\min_{\alpha} \frac{1}{2}\sum_{m=1}^{k}\|\mathrm{w}_m\|^2 + \sum_{i=1}^{l}\sum_{m=1}^{k} e_i^m \alpha_i^m$$

$$\sum_{m=1}^{k}\alpha_i^m = 0, \forall i=1,\dots,l$$

$$\alpha_i^m \le C_{y_i}^m, \forall i=1,\dots,l, m=1,\dots,k$$

$$\mathrm{w}_m = \sum_{i=1}^{l}\alpha_i^m \mathrm{x}_i, \forall m, \quad \alpha = \left[\alpha_1^1,\dots,\alpha_1^k,\dots,\alpha_l^1,\dots,\alpha_l^k\right]^{\mathrm{T}}$$

$$C_{y_i}^m = \begin{cases} 0 & y_i \ne m \\ C & y_i = m \end{cases}$$

采用分治法求解

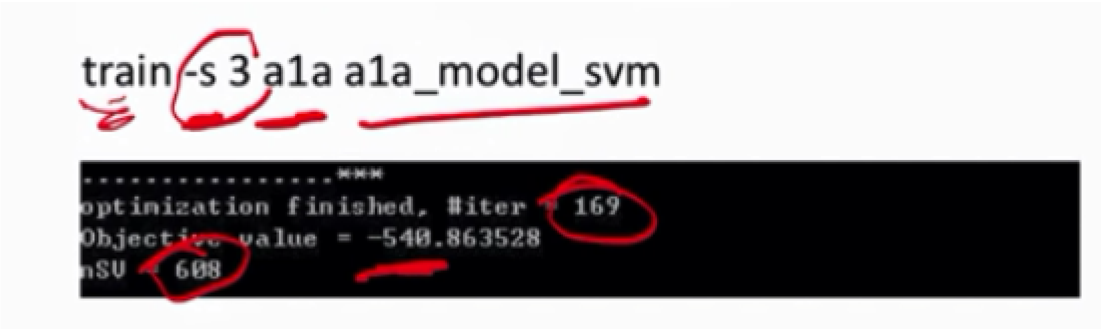
对偶问题用分治法求解

18.8 实验

使用liblinear

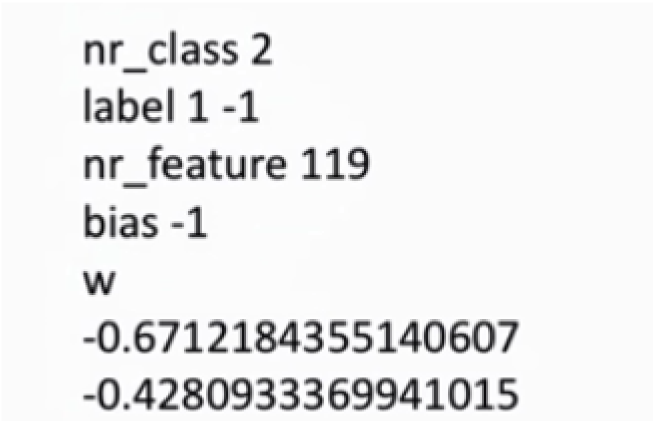
训练模型

生成的模型文件是a1a_model_svm



```
train -s 3 a1a a1a_model_svm
..................***
optimization finished, #iter = 169
Objective value = -540.863528
nSV = 608
```

模型文件的内容如下图



```
nr_class 2
label 1 -1
nr_feature 119
bias -1
w
-0.6712184355140607
-0.4280933369941015
```

- 类别数2
- label 1 -1
- 特征
- 偏置向量是负一 -1
- 权重矩阵
    - $w^T x \to > 0 \quad +1$

模型预测

前面训练完模型之后，我们用预测模型进行预测，测试样本就是a1a.t 文件。

使用前面训练的模型

: a1a_model_svm

预测结果保存到

a1a_predict_svm

精度和前面用RBF的差不多一样。

输入命令之后，显示如下

predict a1a.t a1a_model_svm a1a_predict_svm

```
E:\代码\liblinear\liblinear-2.11\liblinear-2.11\windows>predict a1a.t a1a_model_
svm a1a_predict_svm
Accuracy = 83.8125% (25945/30956)
```

18.9 libsvm和liblinear的比较

https://www.csie.ntu.edu.tw/~cjlin/liblinear/

https://www.csie.ntu.edu.tw/~cjlin/libsvm/

对于svm

libsvm支持非线性核，可以实现非线性模型；liblinear中只有线性核，是线性模型
liblinear可以直接得到w和b，而libsvm不行
求解算法不同，libsvm采用的是SMO算法；liblinear采用的是可信域牛顿法，坐标下降法

$$\mathrm{sgn}\left(\sum_{i=1}^{l} a_i y_i K(\vec{x_i}, \vec{x}) + b\right)$$

$$\mathrm{sgn}(w^T \vec{x})$$

前面在剖析libsvm模型的时候，

liblinear可以直接得到w 和b,而libsvm不行。

求解算法不同，libsvm采用的是SMO算法，而liblinear采用的是可信域牛顿法，坐标下降法。

可以自己研究这两个算法

Xiang-Rui Wang: versions 1.0-1.1.

张开伟：L1和L2损失双SVM，多级（Crammer＆Singer），L1正则化分类器的双坐标下降。

Cho-Jui Hsieh：L1和L2损失双SVM，多级（Crammer＆Singer），L1正则化分类器的双坐标下降。

范荣恩：所有问题的大帮手。

Guo-Xun Yuan and Chia-Hu Ho: L1-regularized classifiers.

Hsiang-Fu Yu和Fang-Lan Huang：L2正则化逻辑回归的双坐标下降

Hsiang-Fu Yu：python界面

Chia-Hua Ho：线性支持向量回归的求解器

Bo-Yu Chu和Chia-Hua Ho：参数选择代码

Mu-Chu Lee和Wei-Lin Chiang：求解器中两个主要操作（点和axpy）的抽象。将牛顿上的两个矩阵向量乘法组合成一个。

Chih-Yang Hsia和Ya Zhu：基于原始的Newton方法（版本2.11）中改进的信任区域更新规则的代码

Hsiang-Fu Yu和Hsin-Yuan Huang：python界面中的scipy支持（版本2.11）

在实际应用中，怎么挑这两个库呢？

如果样本量比较小，用libsvm，精度较高

如果你的特征向量维数很高，训练样本也很多，则选用liblinear

18.10 实际应用

实际应用

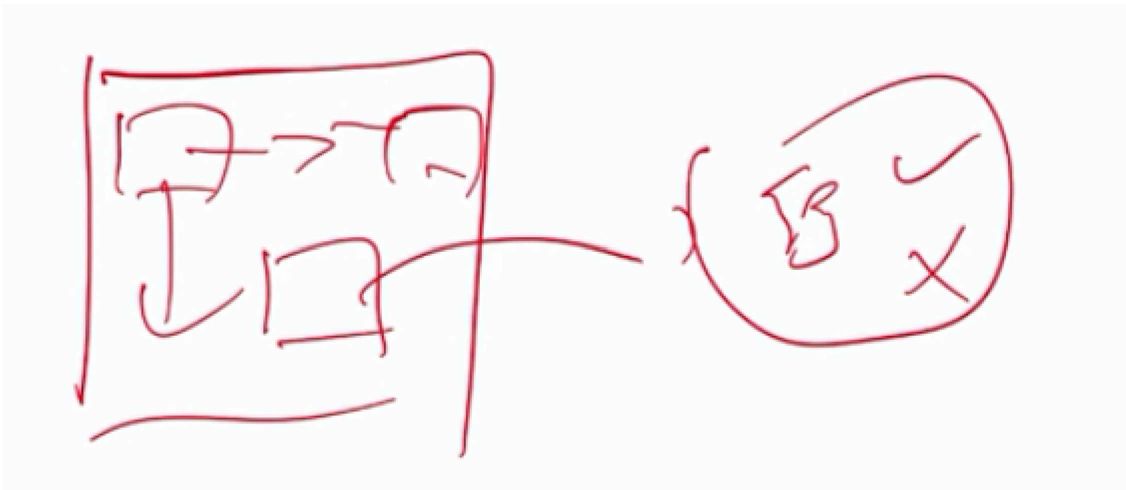
[1] Navneet Dalal, Bill Triggs. Histograms of oriented gradients for human detection. computer vision and pattern recognition. 2005.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

第一篇文献

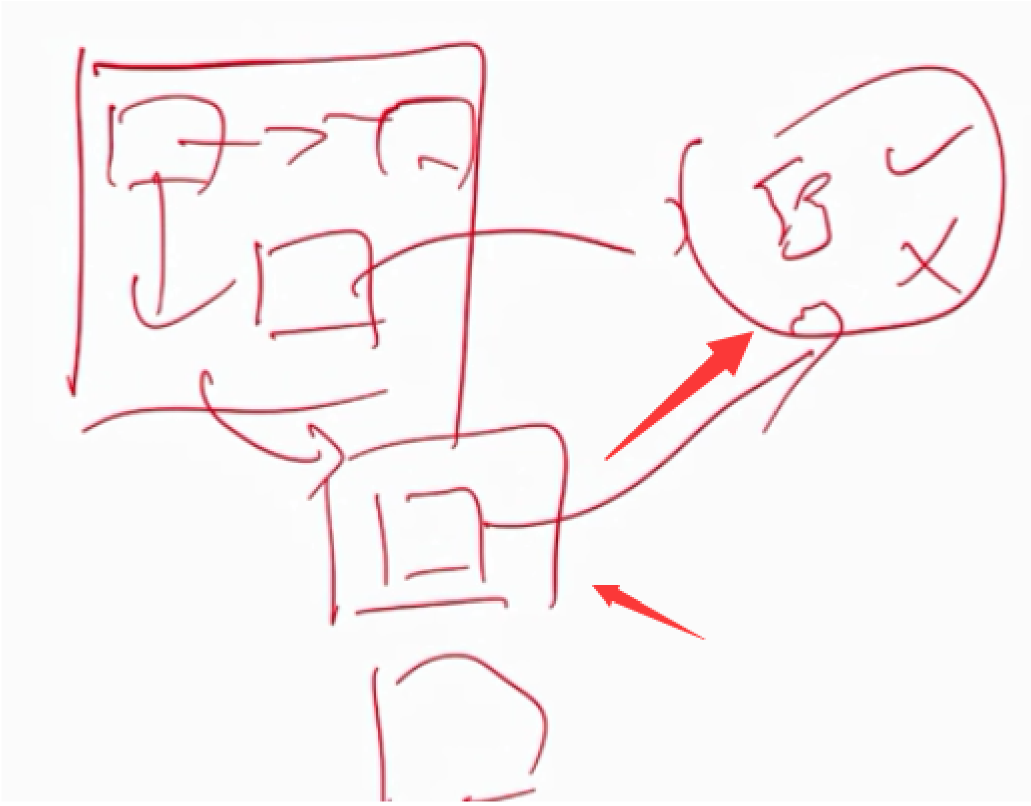
运用非常高的是第一篇文献。

HOG + L-SVM做行人检测



通过一个活动框，从左到右，从上到下，依次滑块，判断物体是不是行人，是个二分类问题。

为了检测行人的大小，经常将框的大小进行缩放，在判断是不是行人。



运算量非常大。

如果用高斯核，精度会高一点，但是速度你是受不了的。

所以只能选用线性支持向量机。

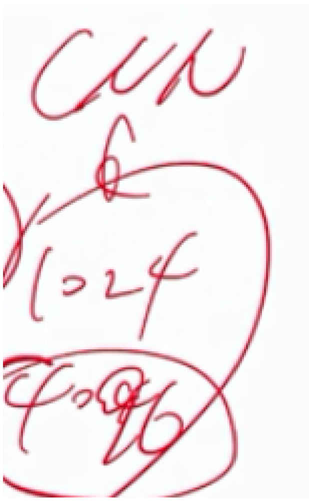
第一篇文章的核心创新点就是基于HOG特征。

传统的机器学习都是这样的。

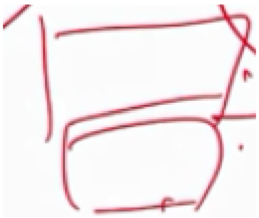
第二篇文献

是这个领域做通用目标检测的文章，

用深度学习来做图像检测，CNN.



得到这样的特征之后，紧接着对图像进行分类。是自行车，还是汽车，还是行人。都是用线性的支持向量机来做的。



如果特征的维度很高，样本数量很多，就只能用线性支持向量机来做。效果也是不错的，保持最大化分类间隔。

18.11 本集总结