



機器學習基礎與演算法

Chapter 5 支持向量機 (Support vector machines)

講師投影片Chapter5 課程投影片 資料與程式碼 播放清單

「版權聲明頁」

本投影片已經獲得作者授權台灣人工智慧學校得以使用於教學用途,如需取得重製權以及公開傳輸權需要透過台灣人工智慧學校取得著作人同意;如果需要修改本投影片著作,則需要取得改作權;另外,如果有需要以光碟或紙本等實體的方式傳播,則需要取得人工智慧學校散佈權。

課程內容

5. 支持向量機 (Support vector machines)

- -Linear SVM
- -Lagrange multiplier
- -Kernel SVM
- -Regularized linear regression
- [實作] SVC

[實作] SVR

Code 放在Hub中的course內

- 為維護課程資料, courses中的檔案皆為read-only, 如需修 改請cp至自身環境中
- 打開terminal, 輸入

cp -r courses-tpe/Machine_Learning <存放至本機的名稱>



Chapter 5 支持向量機 (Support vector machines)

- 範例程式(example)的檔名會以藍色字體顯示且旁邊附上
- 練習(exercise)的檔案以紅色字體顯示且旁邊附上

05-1: Linear SVM

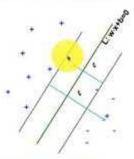


Introduce slack variable

• Not all mistakes are equally bad $\min_{\mathbf{w},b} \frac{1}{2} \mathbf{w}^{\mathsf{T}} \mathbf{w} + C \sum_{i=1}^{n} \xi_{i}$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 - \xi_i \forall i$

If a point is on the wrong side
 get penalty \(\{ \)_{\text{\te}\text{\texi}\text{\text{\text{\text{\text{\text{\text{\texi{\text{\texi}\text{\texi{\texi{\texi{\texi{\texi{\texi\texi{\texi{\texi}\texi{\texi}\\\ \ti}\\\ \



For each data point x: If $d(x, L) \ge 1$ and at the right side: don't care Else: pay linear penalty

MERALBURE





05-2: Lagrange multiplier



Generalized Lagrange multiplier

- Lagrange multipliers is generalized to include the inequality constraints under the Karush-Kuhn-Tucker (KKT) condition
- · Standard form problem



ACCENTAGE AND PARTY.

- Minimize f(x) subject to $g_i(x) \le 0$ (i = 1, ..., p) and $h_j(x) = 0$ (j = 1, ..., m)
- If the task is to maximize f(x), transform the problem into minimize -f(x)
- Lagrangian



$$\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_{i} g_{i}(\boldsymbol{x}) + \sum_{j=1}^{m} \mu_{j} h_{j}(\boldsymbol{x})$$



05-3: Kernel SVM



Generalized Lagrangian and SVM

Generalized Lagrangian

Minimize f(x) subject to $g_i(x) \le 0$ (i = 1, ..., p) and $h_i(x) = 0$ (j = 1, ..., m)

• Lagrangian $\mathcal{L}(\mathbf{x}, \lambda, \mu)$ $= f(\mathbf{x}) + \sum_{i=1}^{p} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{m} \mu_j h_j(\mathbf{x})$

· SVM:

Minimize
$$\frac{1}{2}w^Tw$$
 subject to $1-y_i(w^T\phi(x_i)+b) \le 0$

 $\begin{aligned} & \text{Lagrangian} \\ & \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) \\ & = \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + \sum \lambda_i [1 - y_i (\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b)] \end{aligned}$

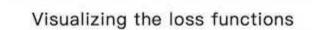


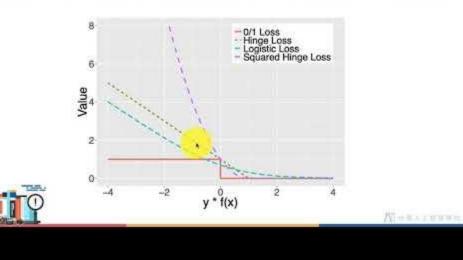




05-4: Regularized linear regression

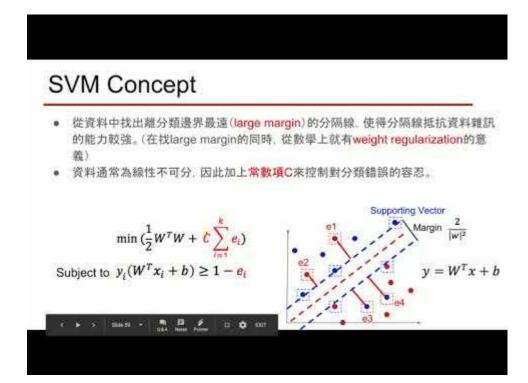








[實作課程] SVM in Scikit Learn - SVM Concept





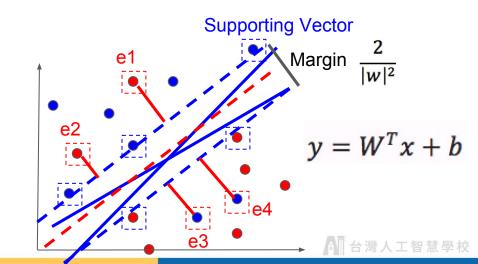
SVM in Scikit Learn - SVM Concept

- 從資料中找出離分類邊界最遠(large margin)的分隔線, 使得分隔線抵抗資料雜訊的能力較強。(在找large margin的同時, 從數學上就有weight regularization的意義)
- 資料通常為線性不可分, 因此加上常數項C來控制對分類錯誤的容忍。

$$\min\left(\frac{1}{2}W^TW + C\sum_{i=1}^k e_i\right)$$

Subject to $y_i(W^Tx_i + b) \ge 1 - e_i$

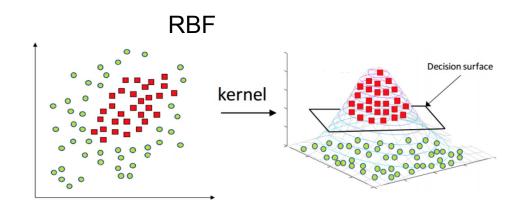




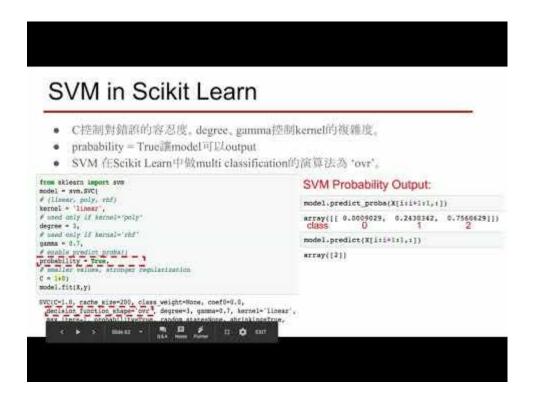
SVM Kernel Trick

- 一般狀況下,資料多為線性不可分。藉由kernel可以將資料做非線性的轉換到多維空間,甚至無限多維空間,使得SVM可以在此將資料做分類。
- Kernel即是將資料做空間轉換,代替linear regression手動生feature的方式
- 一般常見的Kernel方式:
 - Polynomial:高次方轉換
 - Radial Basis Function: 高斯轉換 ⇒ 無限多維空間

Polynomial



[實作課程] SVM in Scikit Learn - SVC





SVM in Scikit Learn

- C控制對錯誤的容忍度, degree、gamma控制kernel的複雜度。
- probability = True讓model可以output
- SVM 在Scikit Learn中做multi classification的演算法為 'ovr'。

```
from sklearn import svm
model = svm.SVC(
# {linear, poly, rbf}
kernel = 'linear',
# used only if kernel='poly'
degree = 3,
# used only if kernal='rbf'
gamma = 0.7
# enable predict_proba()
probability = True,
# smaller values, stronger regularization
C = 1.0
model.fit(X,y)
SVC(C=1.0, cache size=200, class weight=None, coef0=0.0,
 decision function shape='ovr', degree=3, gamma=0.7, kernel='linear',
  max iter=-1, probability=True, random state=None, shrinking=True,
  tol=0.001, verbose=False)
```

SVM Probability Output:

SVM Example

- 1- SVM Kernel Comparison Example---iris
- - 比較不同Kernel產生邊界的形狀差異
- 2- SVM C Parameters Example---iris



- 調整C在三個不同的Kernel上對模型複雜度的限制
- C越大邊界越複雜/簡單?
- 3- SVM Other Parameters Example---iris



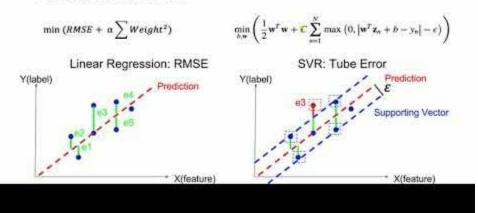
- 調整degree/ gamma
- RBF Kernel 是否可以做到極高準確率?



[實作課程] SVM in Scikit Learn - SVR

SVM for Linear Regression (SVR)

概念與SVM分類問題類似、找出一條線位於資料Supporting Vector的中心、並用常數C來控制對錯誤的容忍。





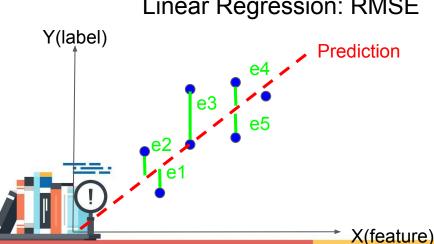
SVM for Linear Regression (SVR)

概念與SVM分類問題類似,找出一條線位於資料Supporting Vector的中心,並用 常數C來控制對錯誤的容忍。

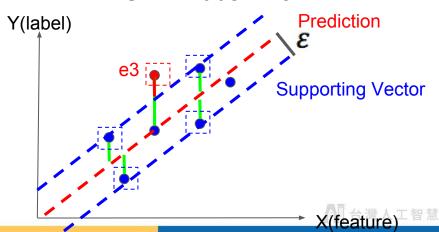
$$\min (RMSE + \alpha \sum Weight^2)$$

$$\min_{b,\mathbf{w}} \left(\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^{N} \max \left(0, \left| \mathbf{w}^T \mathbf{z}_n + b - y_n \right| - \epsilon \right) \right)$$

Linear Regression: RMSE



SVR: Tube Error



SVR in Scikit Learn

- 同SVM, 常數C用來控制模型對錯誤的容忍度。
- kernel有'linear', 'poly' 和 'rbf' 可以選擇。
 - 'poly': 用degree來控制模型複雜度(高次項擴展)
 - 'rbf':用gamma來控制模型複雜度(高斯轉換)

```
from sklearn.svm import SVR
model = SVR(kernel='poly', degree=1, C=0.01)
model.fit(X_train,y_train)
```

4- SVR Example jupyter

- 調整C, degree, gamma看看對預測結果的影響。
- 和之前提到的Linear Regression比較, SVR並不需要自行做座標轉換生成Feature, 可利用Kernel將

一安白打做座標轉換生成Feature,可利用Kerr 資料自動擴展到多維空間。



[實作課程] SVM in Scikit Learn - SVM Short Summary

Short Summary

- 用 SVM 解二元、多元分類及回歸問題
- 用不同Kernal將資料做空間轉換解非線性問題
- 調整degree或gamma來增加模型複雜度
- 利用C做控制模型對誤差的容忍度(模型複雜度)

模型	Linear Regression	Logistic Regression	SVM
用途	連續值預測	分類問題預測	連續值預測問題 分類問題預測
複雜度控制(L1,L2)	Alpha 越大控制力越強	C越大控制力越弱	C越大控制力越弱
SFF man -	3.2 £ 00 m		



Short Summary

- 用 SVM 解二元、多元分類及回歸問題
- 用不同Kernel將資料做空間轉換解非線性問題
- 調整degree或gamma來增加模型複雜度
- 利用C做控制模型對誤差的容忍度(模型複雜度)

模型	Linear Regression	Logistic Regression	SVM
用途	連續值預測	分類問題預測	連續值預測問題 分類問題預測
複雜度控制(L1,L2)	Alpha 越大控制力越強	C 越大控制力越弱	C 越大控制力越弱

SVM&SVR Exercise動手時間

SVM Exercise1 - Classification (upyter



- 選擇不同的kernel. 並調整參數。
- 利用Accuracy評估預測結果
- 利用F1 score和confusion matrix評估預測結果

SVR Exercise2 - Regression in incomplete statement with the second secon



- 觀察各Feature和房價之間的關係並選擇適當的kernel做預測
- 不生成新Feature的情況下是否可以將讓預測結果接近先前做Linear Regression的結果

