* Beyond Binary Linear Classification
  1. OVO time complexity

總共有K個類別，因此OVO的組別總共有種，而每種訓練的時間為因此總共的時間為：

相比於OVA的，可以看出當時OVA需要，但是OVO只要。

* 1. Q-dimensional polynomial transform

如果我們將原本的1維資料轉換成N維資料，並且形式如同Vandermonde matrix的每一項，也就是：

根據題目所述，是個Vandermonde matrix，其行列式為：

而且題目有說明每個均相異，因此行列式值不為零，也因此可逆。

根據Linear Regression的最佳解形式：

因為可逆，所以可以知道也可逆，又可以知道也可逆。

所以就可以改寫為：

可知經過這樣的轉換可以使等於0。。

因此確實存一個Q-dimensional polynomial transform可以使等於0。

* 1. peeking

首先我們可以知道經過轉換後的資料組成的大矩陣就是單位方陣：

所以由Linear Regression所得到的最佳權重會是：

因此可以知道：

至於的話，將***測試資料***中的經過轉換的會是個只有一個1，其他都是0的矩陣。因為***訓練資料***中每個都不一樣，如果***測試資料***跟其中一個***訓練資料***一樣，那麼就會跟其他***訓練資料***不一樣，導致只有一個值是1其他是0；如果***測試資料***跟所有***訓練資料***都不一樣，那麼就會是全都是0的0矩陣。

但其實所有的全都是0矩陣：由於訓練資料跟測試資料的都是i.i.d.的以連續型均勻分佈取樣，因此理論上發生特定值的機率為0，***測試資料***中的經過轉換的會全部都是0矩陣，因此square error就會是對應的的平方：

所以可以表示為：

* Combatting Overfitting

1. virtual examples

首先先列出：

這裡我令是加入noise後的資料。接著將上面的矩陣拆成：

加號左半邊就是原本的；所以我們接著只關注右半邊，右半邊可以改成用帶有noise的形式來表示，順便換掉的起始位置：

可以發現每一個都可以拆成：

如果取期望值會得到：

因為所以中間兩項都消掉了。至於最後的一項要分兩種情形：

這是當的時候，也就是對角線上。

這是當的時候，也就是其他地方：因為這兩個是兩個不同的取樣，所以根據獨立性值 。

所以最後加號右半邊的矩陣取期望值就可以表示為：

所以最後全部統整起來就會得到：

1. augmented error with GD

根據gradient descent algorithm：

所以我們先找出，接著再帶入：

所以可以知道：

,

1. relationship

只要解出最佳的就可以了：

對取一次微分後找極值：

所以將帶入就可以得到的關係式：

所以我們可以知道：

1. Regularizer

先稍微轉換一下：

由於矩陣是個都是正數的對角矩陣，所以可以做出下面的更動：

然後此時令，上面的式子就可以得到：

這樣就找到他們之間的關係了，。

也就是說對原本的式子找到某個使最小，就等同在稍微換位子後的式子中找到一個使最小。

1. Leave one out

永遠只會回傳數量最少的類別。題目正負兩種資料類別都給了N個，所以如果挑出一個用來Loocv預測，其他做訓練，則：

會發現就可以神奇的完美預測了。

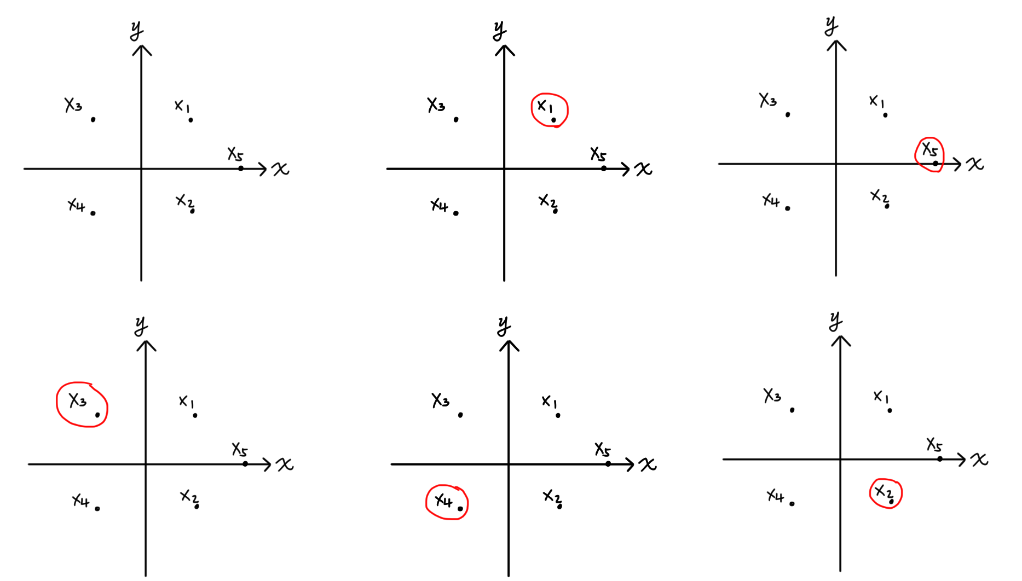
根據的公式可以知道：

達到完美的100%正確率。

* Learning Principles

1. value of the expectation

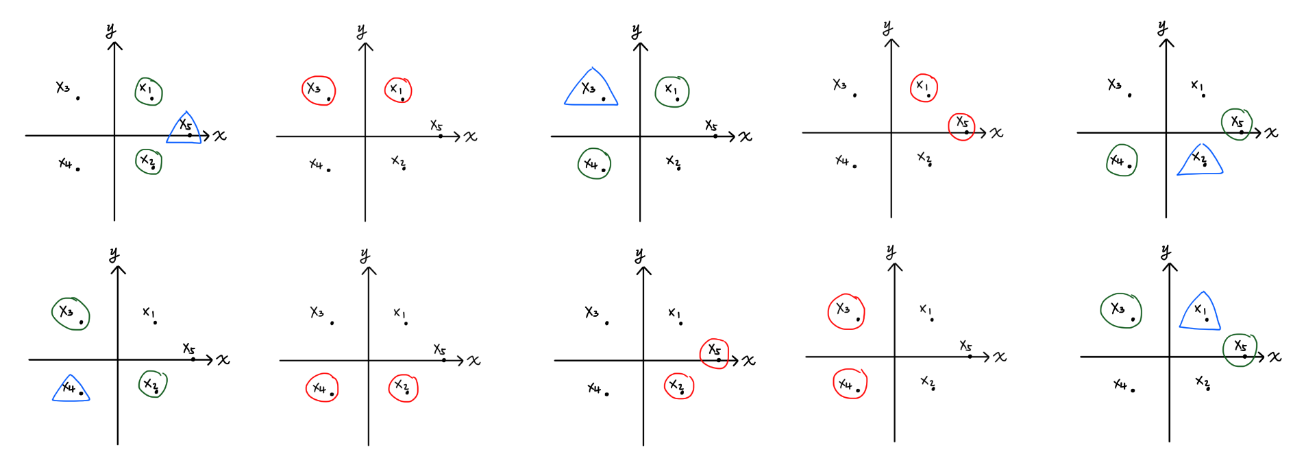
下面列出所有的可能：



上圖中，圈起來的代表label是，沒圈起來的代表是。

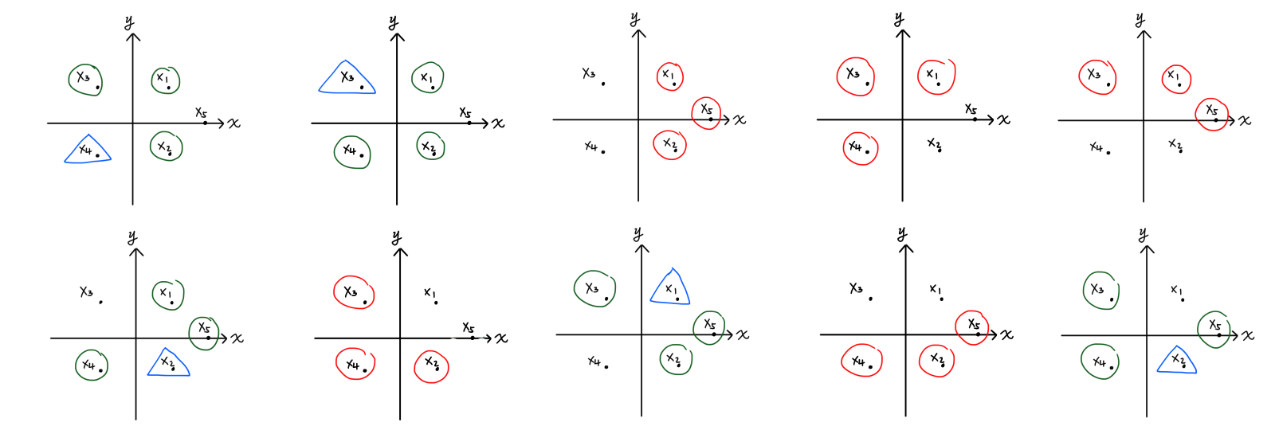
紅色的圈代表該種情形可以用一條線完美分開；左上角的都沒圈情形，也是一種可以完美分開的情形。

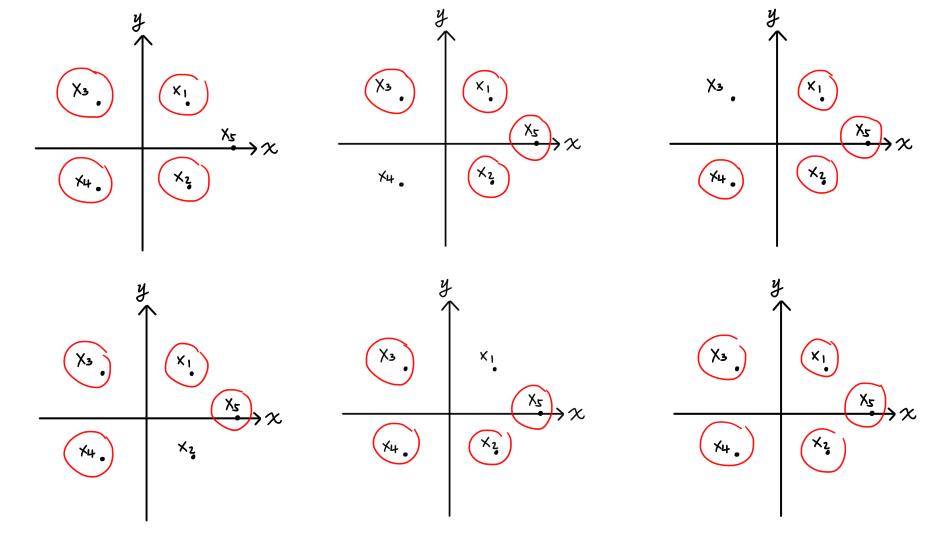
上圖是「都不圈」與「圈一個」的情形。



上圖是「圈兩個」的情形。有「圈」一樣代表該點的label是。這裡多了綠色圈，代表該種情形無法用一條線完美分開，並且我用「藍色三角形」代表了我們犧牲的那個點，他是預測錯誤的點。

三角形不是圈。



上圖是圈三個的情形，規則一樣。

最後是圈四個跟圈五個，可以發現都可以完美預測。

現在我們來算算期望值：

完美預測的錯誤率是0，所以我們只要考慮有錯的就好。

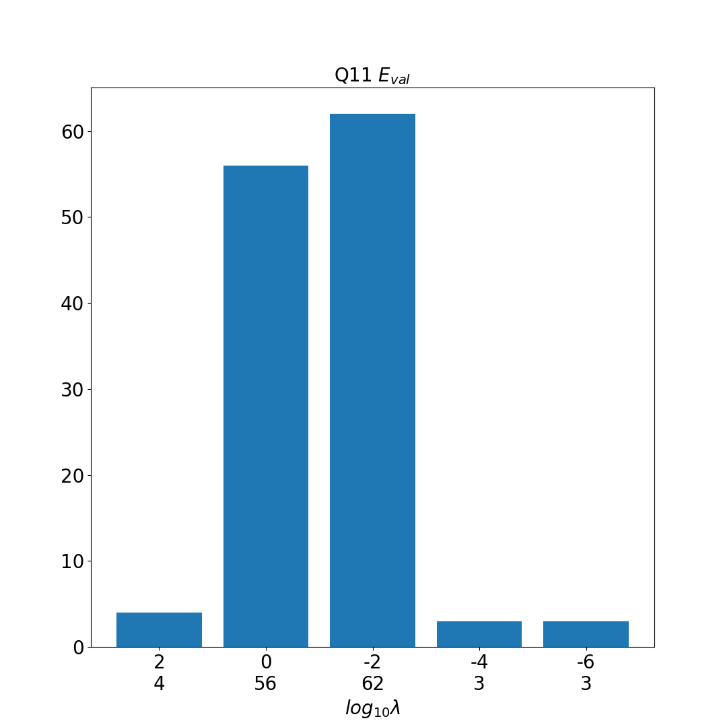
從上面的圖可以發現，有錯的話，就只有錯一個點，也就是說錯誤率是，而總共有10種情形是有預測錯誤的，再乘上發生的機率，就得到我們要的期望值了：

* Experiments with Regularized Logistic Regression

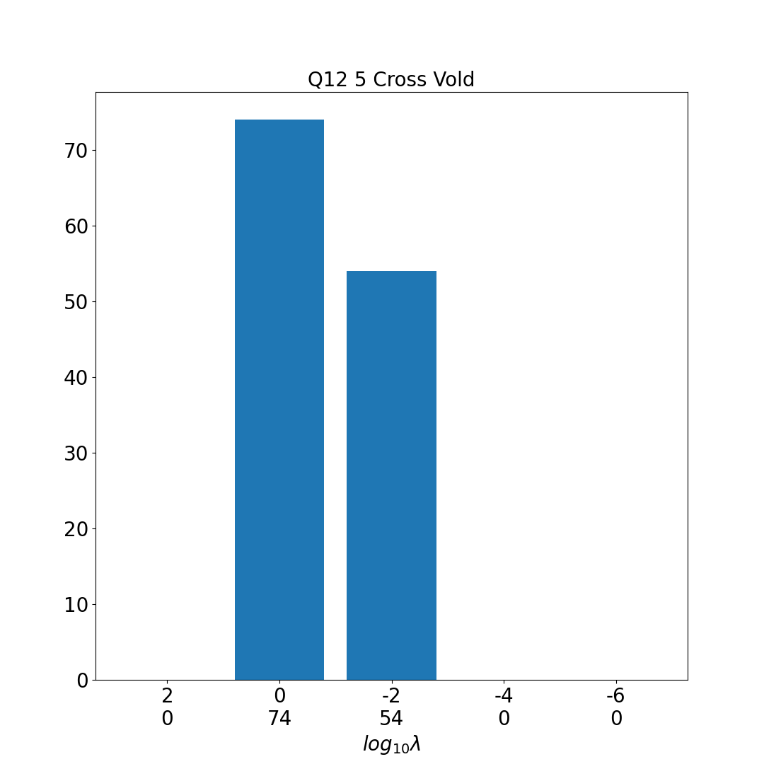
1. bestamong

根據實驗結果，最佳的是次方。

1. bestamong



1. bestamong 5 fold Cross Validation

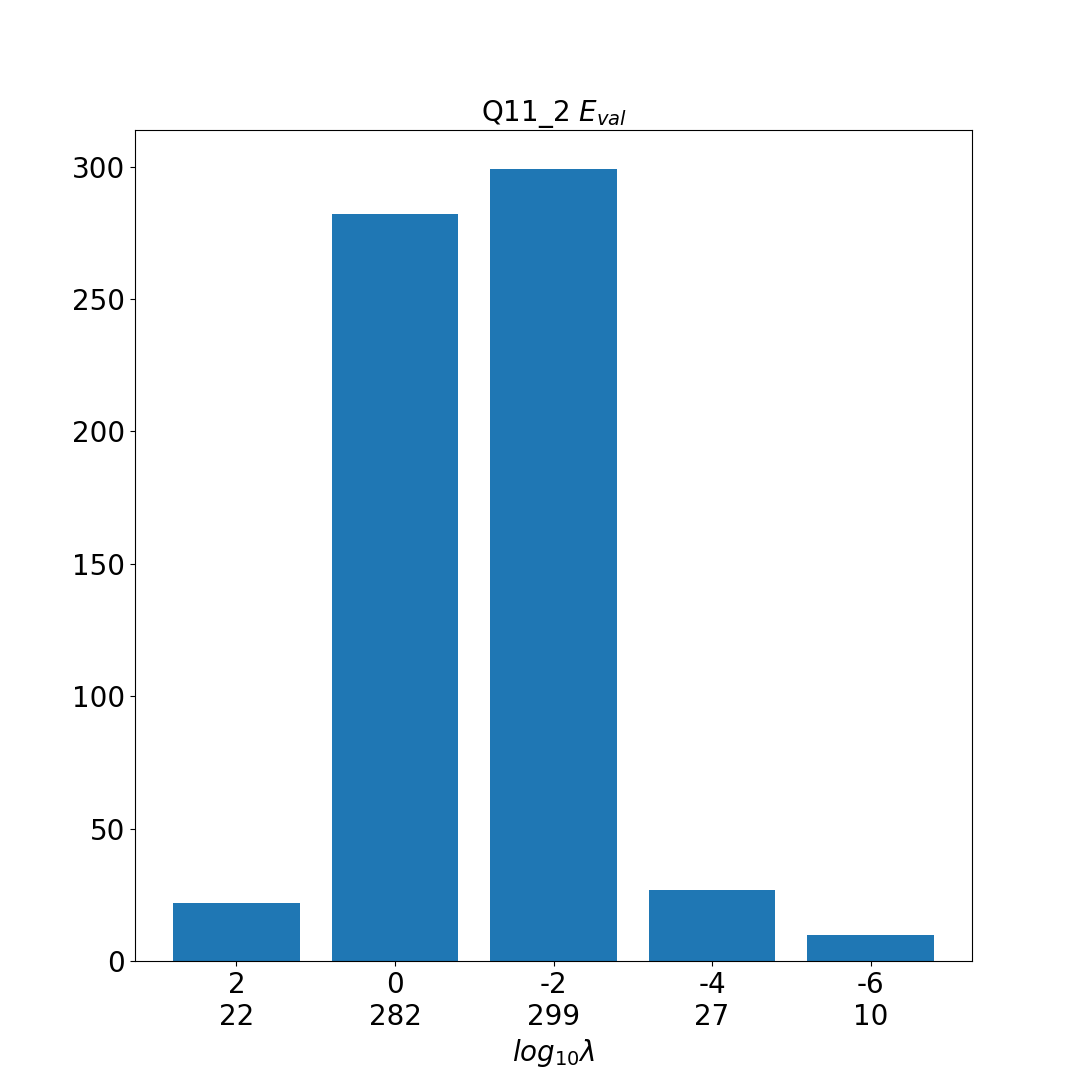


首先可以發現有經過驗證操作的兩種結果得到的主要都是或次方，而使用得到的則是次方，這裡可以看出使用做判斷會造成的overfitting現象。

接著是單純使用1次驗證以及使用Cross Validation所得到的差異，單純使用1次驗證會發現有零星的情形選擇了、跟次方，Cross Validation則沒有這樣的情形；兩者都主要選擇或次方，並且Cross Validation更偏好0次方，1次驗證則是稍微偏好次方。

對於零星的情形的差異性，我認為是因為Cross Validation可以綜合考量5個不同位置的等分去進行評估，所以如果某次random select產生的資料接近極端情形，讓1次驗證的判斷產生誤差，此時Cross Validation的特性就可以避免這種情況。

至於Cross Validation偏好0次方，1次驗證稍微偏好次方的現象，原本我以為是實驗上的誤差，但是當我將1次驗證的次數拉到128的5倍後，發現得到一樣的分佈圖：



我才意識到應該是跟剛剛零星情形的現象具有相同的原因：Cross Validation考慮了五種意見而做出決定；某次Cross Validation選擇了次方，1次驗證因為只考慮一種意見，所以有可能會受到該資料的分佈的影響而選擇了次方。

* Bonus

1. Scale or Regularize

首先要注意到老師給的條件，，因為這代表最佳的長度是大於，一旦我們加上L2 Regularizer (或者說C-constrained linear regression)後得到的最佳權重長度就會在邊界上，也就是說：

。

* If ,thensolves the C-constrained linear regression prob.

根據ridge regression的解形式：

此時如果再看原本的linear regression的解形式：

會發現跟其實就只差一個常數倍；而我們知道，因此將長度伸縮成而得到的就是：

* Ifsolves the C-constrained linear regression prob.,then

如果solves the C-constrained linear regression prob.，我們可以知道，也就是說：

注意記得確認得到的結果要滿足過程中跟可逆的假設。