HW3: Supporting Vector Machine

109511219 林錦樑

1. ν -SVM model

```
v-SVM
====================
v = 0.1  Kernel type = rbf
accuracy: 0.976
====================
v = 0.9  Kernel type = rbf
accuracy: 0.9164
====================
v = 0.5  Kernel type = rbf
accuracy: 0.9484
```

```
v = 0.5  Kernel type = linear
accuracy: 0.9284
====================
v = 0.5  Kernel type = poly
accuracy: 0.8724
====================
v = 0.5  Kernel type = sigmoid
accuracy: 0.9272
```

ν-SVM 是 C-SVM 的另一種表示方式，ν 為 0 到 1 之間的數，用來控制 supporting vectors 的數量，可以做為 margin error 的比例上限，supporting vectors 數量的比例下限。調整 ν 值比較下來，在 ν 較小時可限制住 margin error 的上限與降低 supporting vectors 數量，減少落在 margin 上的點，使得模型準確率較高，隨著 ν 值上升，準確率會下降。從下圖可以看出不同 ν 值時，supporting vectors 的數量變化。

```
Support vectors shape for nu=0.5:  (3594, 784)

Support vectors shape for nu=0.1:  (1316, 784)

Support vectors shape for nu=0.9:  (4889, 784)
```

Kernel type 的部分是 kernel function 的不同，kernel function 可以將 non-linear 的問題轉成 linear。default 的 kernel type 是 rbf，在課堂中學到的是 linear，其公式如下圖，其中 gamma 也是可調整的參數，預設為 1 / (n_features * X.var())。我在實驗時將 ν 設為預設的 0.5，比較不同 kernel type 的情況下，表現最好的依序是 rbf、linear、sigmoid、poly。

- Common kernel functions for SVM
  - linear $k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
  - polynomial $k(\mathbf{x}_1, \mathbf{x}_2) = (\gamma\, \mathbf{x}_1 \cdot \mathbf{x}_2 + c)^d$
  - Gaussian or radial basis $k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)$
  - sigmoid $k(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\gamma\, \mathbf{x}_1 \cdot \mathbf{x}_2 + c)$

2. C-SVM model

```
C-SVM
====================
C = 0.1  Kernel type = rbf
accuracy: 0.958
====================
C = 10  Kernel type = rbf
accuracy: 0.9836
====================
C = 1  Kernel type = rbf
accuracy: 0.9784
```

```
C = 1  Kernel type = linear
accuracy: 0.95
====================
C = 1  Kernel type = poly
accuracy: 0.976
====================
C = 1  Kernel type = sigmoid
accuracy: 0.9072
```

C-SVM model 是上課時學到的 SVM model。C 為 Regularization parameter，一定是正數。C 越大時，對分類錯誤的懲罰較大，margin 會變小，較擬和 training data 使得 accuracy 較高，但泛化能力較差；C 越小時，對分類錯誤的懲罰較小，margin 會變大， training data 的 accuracy 較差，但泛化能力較強。C 預設為 1，我將 C 放大 10 倍得到 C=10，C 縮小 10 倍得到 C=0.1，在 testing data 上的表現為 C 越大 accuracy 越大。比較不同 kernel type 的情況下，表現最好的依序是 rbf、poly、linear、sigmoid。

3. Supporting vectors

```
Number of support vectors for each class:  [186 124 330 285 220]
(1145, 784)
```

我選擇 C-SVM 的 model 來分析 Supporting vectors，參數都是 default setting，C=1 kernel='rbf'，從上面的結果來看，這組參數的效果是第二好的。可以透過 model.support_vectors_的 shape 得到共有 1145 個長度為 784（圖片像素點數量）的 supporting vector。使用 model. n_support_可以得到每個類別的 supporting vector 數量。其中數字 0 有 186 個，數字 1 有 124 個，數字 2 有 330 個，數字 3 有 285 個，數字 4 有 220 個。使用 model. support_則可以得到 supporting vector 在 training data 中的 index，但由於數量太多，以及 index 無法直接與圖片檔名對照，在這就不一一列出。