Basic Concepts & Classification Based on Probability

- feature vector 是 column vector。
- 每一個 feature 是 feature space 中的一個點，其中 feature space 的維度就是該 dataset 中 feature 的個數。
- **k-nearest-neighbor (kNN):** 選擇 k 個與待分類點 x 最接近的樣本，並計算這 k 個樣本中哪一個 class 佔的比例最高，則將 x 歸類到其 class 中。此方法概念簡單，但實作時需要龐大的計算量(需計算兩兩 sample 之間的距離)。
- **variation of kNN:** 從所有 class 中選擇 k 個與待分類點 x 最接近的樣本點，並分別算出每一個 class 中選擇的這些點的體積(volume)。x 將會被分到 volume 最小的那個 class 中。
- **Bayesian classifier:** 將待分類點 x 歸類到最可能的 class 中。
  - ✓ priori probability 先驗機率: P(wi),… 為 class i 佔所有樣本數的比例。
  - ✓ class-conditional probability, probability density function (pdf) 條件密度函數: P(x|wi),… 為樣本 x 在 wi 分布中的機率。可以說給定資料 wi，x 屬於 wi 的機率。
  - ✓ posteriori probability 後驗機率: P(wi|x) 作為該分類器分類的依據。當 P(w1|x)>P(w2|x)，就將 x 歸類到 class2。
  - ✓ 貝氏定理 P(wi) * P(x|wi) = P(wi|x) * P(x)，我們可以直接利用 P(wi) * P(x|wi)來做決策。當 P(w1) * P(x|w1) > P(w2) * P(x|w2)，則將 x 歸類到 class2。
  - ✓ Decision boundary 為 P(wi) * P(x|wi) = P(wj) * P(x|wj)
  - ✓ Probability of classifier error: 以 2-class 為例。
    Pe = (class2 誤歸類成 class 1 的機率) + (class1 誤歸類成 class2 的機率)
    = $P(w1) * \int_{R2} P(x|w1)dx + P(w2) * \int_{R1} P(x|w2)dx$
- **Minimum-risk classifier:** 根據風險大小做分類。
  - ✓ λ_{jk} 為將 class j 誤歸類成 class k 的 loss。
  - ✓ 以 2-class 為例。將 x 歸類到 class 1 (class 2) 的風險為 r1 (r2)。
    $r1 = \lambda_{11} * P(x|w1) * P(w1) + \lambda_{21} * P(x|w2) * P(w2)$
    $r2 = \lambda_{22} * P(x|w2) * P(w2) + \lambda_{12} * P(x|w1) * P(w1)$
    當 r1>r2，則將待分類點 x 歸類到 r2，反之亦然。
- **Gaussian (normal distribution)** 常用來表示 class 的 pdf。
- **Nonparametric pdf estimation:** 使用 local data distribution
  - ✓ $p(x) \approx \frac{k}{NV(x)}$，$V(x)$為以 x 為中心的體積、k 表示在 $V(x)$的樣本數、N 為樣本數總數。
  - ✓ 有兩種方法: 固定 $V(x) \rightarrow$ Parzen Window；固定 $k \rightarrow$ kNN。

- ✓ 選擇的 neighbors 越多，得到比較 smooth 的 pdf，對於雜訊也較不敏感，但因為過多的 averaging 會使得較多的 local information 遺失；選擇的 neighbors 越少，則對於雜訊更敏感，得到的 pdf 就更 noisy，但保留較多的 local information。

- **Naïve Bayes classifier:** 將所有的 feature 視為獨立、能解決 feature 維度的問題。N 維的問題就可 reduce 成 N 個 1 維的問題，因此 pdf 則變為$P(x|wi) = \prod_{j=1}^{n} \mathrm{P}(xj|wi)$。

Classifier Evaluation

- Confusion matrix: 用來記錄分類錯誤與否個數的矩陣。
- Two class confusion matrix: $\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$。T:true, F: false, N:negative, P:positive。

  TN: 已知為 No 的情況下，判斷正確(即判斷為 No)。
  FP: 已知為 No 的情況下，判斷錯誤(即判斷為 Yes)。
  FN: 已知為 Yes 的情況下，判斷錯誤(即判斷為 No)。
  TP: 已知為 Yes 的情況下，判斷正確(即判斷為 Yes)。
  PD (probability of correct detect): TP/(TP+FN)。
  FA (probability of false positive/alarm): FP/(FP+TN)。
- ROC (Receiver Operating Characteristic) Curve: 橫軸為 FA，縱軸為 PD。

Linear Classifier

- Linear classifier 顧名思義就是一個擁有 linear decision boundaries 的分類器。
- 一個簡單的概念: 以 two class 為例，input 為 xi；output 為(+1)或(–1)，其中輸出的值代表分到兩類中的其中一類。我們可以用 Xw = d 表示，X 為 input、w 為權重，X 經過 w 作用後會得到 d，而 d 的結果就是分類的結果。因此我們想要得到的是 w，如果知道 w 就可以對 testing samples 做判斷。

Neural Network

- **Data normalization:** 會影響收斂程度。例如有一個 datatset 有兩個 feature，其中一個 feature 的數值較另一個 feature 大許多，則在 training 時就比較難 train 到數值較小的 feature。
- **Nonlinear activation function:**
  - ✓ Sigmoid/logistic and tanh: 皆為連續可微分函數，但在函數兩側有接近 0 的梯度(會使 weight update 緩慢)。
  - ✓ RELU (rectified linear unit): 非線性且無 vanishing gradients (梯度趨近 0)

- **Learning rate:** 選擇較大的，能加速訓練過程，但震盪程度(oscillation)較大；選擇較小的，雖然訓練速度慢，但震盪程度低。通常在調整時，加速慢，煞車快。
- **Momentum:** 用來 speed up 訓練過程與避免 oscillation ，也就是達到 acceleration 與 stabilization。對於一個高原地區(緩慢持續下降)，就能發揮 speed up 功效。Momentum 也可用來避免 local minima。
- **Bias-Variance Dilemma:** bias 越小表示 fitting 越好；variance 越小表示 high confident。可以透過使用較複雜的 model 來達到降低 bias，但會連 noise 也一起訓練到，使得 over fitting；可以透過使用較簡單的 model 來降低 variance，但學到的 samples 少，而 under fitting。
- **Regularization**: 透過 minimize E 來降低 bias 與 model complexity。前項為 fitting error，後項為 regularization。其中後項會選擇係數比較低的 model。

$$E = \sum_i \left| y_i - \sum_{k=0}^{d} a_k x_i^k \right|^2 + \lambda \sum_{k=0}^{d} a_k^2$$

- **Cross validation:** 將 samples 分成 k 堆，且每一堆有相似的資料分布。在訓練時會執行 k 次。每次都會拿第 k 堆做為 validation，其餘為 training。透過不同的組合(validation & training set) 來訓練與驗證，以找到適合的 model。
- **Cover's Theorem:** 將一個複雜的 pattern recognition 問題移至高為度空間處理，比起在低維度空間時，更有可能 linear separable。
- **RBF (Radial Basis Function) network:** 由 RBF unit 組成。RBF 的 output 為 $F(x) = \sum_{i=0}^{m} w_i * \varphi(|x - ci|)$，其中 Gaussian 形式的 $\varphi(x) = \exp(\frac{|x-c|^2}{-2\sigma^2})$。
- **SVM (Support Vector Machine):** 選擇一個 hyperplane 使得距離所有的 sample 最遠。
  - ✓ <u>linear separable cases:</u>
    ① margin = 2 * (hyper plane 到 closet sample 的距離)
    ② support vector 為最接近邊界的 samples
    ③ discriminant function 為 $g(x)=w^Tx + w0$
    ④ sample x 與 decision boundary (g(x) = 0) 距離為 $\| w^Tx + w0 \| / \|w\|$
    ⑤ $\| w^Tx + wo \| = 1$, x 為 support vector，則 margin = 2 / ||w||。要找到最大的 margin 就是找到最小的||w||。可以把此問題寫成數學式子:
    Goal: minimize $J = 1/2 * \|w\|^2$
    Subject to constraints: $yi * (w^Tx + w0) \geq 1$ (讓所有 samples 落在緩衝區外)
    yi = 1, if x 為 class 1；yi = –1, if x 為 class 2
  - ✓ <u>non-separable cases</u>:
    ① 將 constraint 改寫成:
    $yi * (w^Tx + w0) \geq 1 - \xi$ where $\xi \geq 0$

Goal: minimize J = $\frac{1}{2}|w|^2 + C\sum_{i=1}^{n}\xi_i$

其目標隱藏著 tradeoff (large margin & minimum classifier error)。前項會放大 separable band；後項會縮小 separable band。

✓ non-linear SVM:

① 使用 kernel trick，利用 function 來達成投影到高維度的轉換。

✓ SVM 可視為 MLP 或 RBF network 的結構。他們其中差異為:

① # hidden neurons: MLP 與 RBF 會先決定；SVM 則透過 optimization。

② problem complexity: MLP 與 RBF 根據 hidden neuron 數量而定；SVM 根據 hyperplane 而定。

③ computation cost: MLP 與 RBF 較低；SVM 較高。

Feature Selection

- **FDR (Fisher's Discriminant Ratio):** between class variance 越大，within class variance 越小，其 separability 越好。

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \propto \frac{S_b}{S_w}$$

- **Sequential Feature Selection:** 分為 sequential forward 與 backward。
- **FLD (Fisher's Linear Discriminant):** 透過投影降維度，使得 class separability 最大。

投影後 class i 的 mean: $\mu_i = w^T\mu_i$

投影後 class i 的 variance: $\sigma_i^2 = w^T\sum_i w$

$$FDR(w) = \frac{w^T S_b w}{w^T S_w w}$$

對 w 做偏微分: $S_b w = \frac{w^T S_b w}{w^T S_w w} S_w w = \lambda S_w w$，其中 w 延著$(\mu_1 - \mu_2)$方向。

- LDA (Linear Discriminant Analysis): 維度降低則 separability 下降，因此選擇保留 eigenvalue 大的 eigenvector。