Intro to ML Course Review

Machine learning basics

- point estimator: a single estimation $\hat{\theta}$ of fixed but unknown parameter θ, with expectation over the whole dataset
  - **bias($\hat{\theta}$) =** $E[\hat{\theta}] - \theta$**,** difference between prediction and true function
  - **Var($\hat{\theta}$) =** $E[(\hat{\theta} - E[\hat{\theta}])^2]$**,**sensitivity of the model to the individual data points
- **under-fitting:** high bias low variance
- **over-fitting**: low bias high variance. too many # of features compared to # of training data. test error high
- estimate f(x) for a test point x:
  - **MSE** = $E[(y - \hat{f}(x)^2] = (f(x) - E[\hat{f}(x)])^2 + E[\hat{f}(x) - E[\hat{f}(x)])^2] + \sigma^2$=Bias^2 + Variance + Irreducible error
- l**ikelihood function,** a function of θ, Find an estimation of θ that maximizes the probability of the observed data set →$L(\theta|x) = probability(X = x; \theta) or f(X = x; \theta)$
- **Bayes Rule:** $Pr(Y_k|X) = \frac{Pr(Y_k,X)}{Pr(X)} = \frac{Pr(X|Y_k)Pr(Y_k)}{Pr(X)}$
- **MAP: argmax_**θ $P(\theta|X)$**, knowing** $P(X|\theta)$
- **MLE: argmax_**θ $P(X|\theta)$
- train(model fitting), validation(model selection, i.e.find λ), test(evaluate generalization error)
  - cross-validation: Repeatedly partition dataset into different training and validation sets, then average performance for model selection
  - leave-one-out CV (computational expensive)
  - k-fold CV (e.g, k = 5,10)

Linear regression

- $\min_{w,b} E[(w^T x + b - y)^2] -> \min_{w,b} 1/N \sum_N [(w^T x_i + b - y_i)^2]$ since E approaches mean given large data
- $W = (X^T X)^{-1} X^T y$
- nonlinear regression: nonlinear basis function
  $\sigma_0(x) = 1, \sigma_1(x) = x, \sigma_2(x) = x^2 -> \min_{w,b} \sum_N [(w^T \sigma x_i + b - y_i)^2]$

Ridge regression

- $\min_{w,b} \sum_N [(w^T x_i + b - y_i)^2] + \lambda ||w||^2$, as λ→∞ w→0
- $W = (X^T X + \lambda I)^{-1} X^T y$
- sparse solution: ||w||

KNN:

- Complexity ∈ $O(Nd)$
- $\hat{y} = mode(\{y_{x'} | x' \in N_k(x)\})$, select k with lowest **validation error**(cross validation)

- weighted distance function, bigger weight less importance
- when k=1, zero training error, high variance, over-fitting
  - Training error tends to increase with K. Test error and CV error tend to first decrease then increase

Classification

Given L(Y,Y')→1 (incorrect)or 0, Expected error = $E[L(Y,\hat{Y})] = E_X E_{Y|X}[L(Y,\hat{Y})|X]$→
$\hat{y} = argmin_{g\in Y} \sum_k^K L(Y_k,g)P(Y_k|X) = argmin_{g\in Y} 1 - Pr(g|X) = argmax_{g\in Y} Pr(g|X)$

Generative classifier:

- model **Pr(Y_k , X)** under distributional assumptions and use Bayes rule to predict class
- Can generate synthetic data X by drawing samples from Pr(X)
- Can be wasteful if we only need a classification decision (since only Pr(Y_k|X) is required).

Naive bayes:

- assume inputs are conditionally independent, $Pr(Y_k|X) = Pr(X|Y_k)Pr(Y_k)$ $Pr(Y_k) = \frac{class_K}{total}$, $Pr(X|Y_k)$ depends on the event situation, should be given

Gaussian mixture models

- $Pr(x|\theta) = \sum_k Pr(z=k)Pr(x|z=k) = \sum_k \pi_k p_k(x|\theta) = \sum_k \pi_k N(x|\mu_k,\sigma_k)$
- $Pr(x|z=k) = N(x|\mu_k,\sigma_k)$, $P(x) = \sum_z P(x|z)p(z) = \sum_k \pi_k N(\mu_k,\sigma_k)$
- responsibility: $P(z_k=1|x) = \frac{P(z_k,x)}{p(x)} = \frac{\pi_k N(\mu_k,\sigma_k)}{\sum_k \pi_k N(\mu_k,\sigma_k)} = \frac{P(z_k=1)p(x|z_k=1)}{\sum_j^k P(z_j=1)p(x|z_j=1)} = \gamma(z_k)$
- P(X|z)→we know which Gaussian generates the point, P(z_k=1|x)→ probability
- **EM**:
  - log likelihood $lnP(x|\pi,\mu,\sigma) = \sum_n ln(\sum_k \pi_k N(x_n|\mu_k,\sigma_k))$, no closed form
  - initialize $\mu_k = \frac{1}{N_k}\sum_n \gamma(z_{nk}x_n)$ weighted mean of all the points in the data set
    $N_k = \sum_n \gamma(z_{nk})$ # of points assigned to k
    $\sigma_k = \frac{1}{N}\sum_n \gamma(z_{nk})(x_n-\mu_k)^2$ weighted sum of covariance
    maximize log likelihood wrt to $\pi_k = \frac{N_k}{N}$ the # of data assigned to K/ total
  - E step: evaluate $\gamma(z_{nk})$ using current π,μ,σ
  - M step: re-estimate parameters π,μ,σ using new $\gamma(z_{nk})$

K-means Clustering

- Hard assignment to each cluster, a particular limit of EM for Gaussian mixtures (covariance matrix of the mixture components are given by ϵI, I→0)

EM for Gaussian mixtures

- Soft assignment based on responsibilities (assign to cluster with the highest value of responsibility) hence more flexible
- Can use K-means for initialization

Discriminative classifier

- model conditional probability Pr(Y_k | X) directly based on data with no distributional assumptions.

Logistic regression

- $y = \sigma(X) = \frac{1}{1+exp(-x)} = \frac{exp(x)}{1+exp(x)}$, if x>0, then exp(x)>1, $\sigma(x)$>0.5

- $Pr(y=1|x)=\sigma(w_0+w^Tx)$,$Pr(y=0|x)=1-\sigma(w_0+w^Tx)$
- multi-class classification: one vs rest(train k classifiers) one vs one(train K(K+1)/2)

SVM: maximize margin boundary for $\hat{y}=sign(w^T\sigma(x)+b)$

  - Sparse solution: prediction of output only requires evaluation of kernel functions at a subset of the training data point

- **hard margin**
  - distance between a point x and plane $w^T\sigma(x)+b=0$ is $\frac{(w^T\sigma(x)+b)}{||x||}$, maximum margin is found by solving max w,b {1/||w|| min n for $y_n(w^T\sigma(x_n)+b)$ } and we assume the point closest to the boundary$y_n(w^T\sigma(x_n)+b)$=1, then margin becomes 1/||w||
  - objective: $min_{w,b}=\frac{1}{2}||w||^2$ st $y_n(w^T\sigma(x_n)+b)\geq 1,\forall n$
  - Lagrangian: L(w, b,a) = $\frac{1}{2}||w||^2-\sum a_n[y_n(w^T\sigma(x_n)+b)-1]$
  - KKT: $w=\sum_n a_n y_n\sigma(x_n)$, $0=\sum_n a_n y_n$
  - sparse solution:
    - $a_n\geq 0, y_n(w^T\sigma(x_n)+b)-1\geq 0, a_n[y_n(w^T\sigma(x_n)+b)-1]=0$
    - support vectors x_n : $y_n(w^T\sigma(x_n)+b)=1$
  - prediction $\hat{y}=sign(w^T\sigma(x)+b)=sign(\sum_{n\in S}a_n y_n k(x,x_n)+b)$, $k(x,xn)=\sigma(x)^T\sigma(x_n)$
- **soft margin**
  - Introduce slack variable $\epsilon \geq 0$ for each data point
  - $\epsilon = 0$ for data points that are on or inside the correct margin boundary
  - $\epsilon = |y - w\phi(x) - b|$ for other data points (e.g., misclassified data points have $\epsilon > 1$)
  - $min_{w,b,\epsilon}=\frac{1}{2}||w||^2+C\sum_n^N \epsilon_n -> y_n(w^T\sigma(x_n)+b)\geq 1-\epsilon_n, \epsilon_n\geq 0 \forall n$
  - Large C often results in a smaller-margin hyperplane; small C often results in a larger-margin hyperplane

Perceptron

- binary classification, $y\in\{-1,1\}$, x has d features
- prediction $\hat{y}=sign(\sum_d^D w_d x_d+b)=sign(w^Tx+b)$, sign(t)= 1 if t>=0, -1 if t<0
- algorithm: if y$\hat{y}\leq 0$ the prediction is wrong then $w_d=w_d+yx_d, b=b+y$
- Suppose the perceptron algorithm is run on a linearly separable data set D with the margin $\gamma > 0$. Assume that $| x | \leq 1, \forall x \in D$. Then the algorithm will converge after at most $\frac{1}{\gamma^2}$ updates. Larger the margin faster it converges. If the data is linearly non-separable, the algorithm will never converge.
- Multi-layer: learned nonlinear feature representation+ linear predictor($h^{(n)}=f_n(W^{(n)}h^{(n-1)+b^{(n)}}$)