Bias and Variance

- Given an estimator $\hat{\theta}$ for population parameter θ, we define the bias of $\hat{\theta}$ as $\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$
- $\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$

Bias-Variance Decomposition (expectation of the error) $\mathbb{E}[(y - \hat{f}(x))^2] = \left(\text{Bias}(\hat{f}(x))\right)^2 + \text{Var}(\hat{f}(x)) + \sigma^2$,

Linear Smoother (weighted KNN) For real valued targets, $\hat{y}(x) = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$

$$\text{For categorical valued targets, } \hat{y}(x) = argmax_{v \in V} \sum_{i=1}^{n} w_i \delta(v, y_i)$$

where w_i = inverse-distance. (e.g. $1/Dist(x, x_i)$) and v = target values.

p-norm of 2-dimension vector $\|x\|_p = (|x_1|^p + |x_2|^p)^{1/p}, \quad p \geq 1.$
Note: $\|x\|_p \geq \|x\|_q$ whenever $p < q$.

Min-max Normalisation $x'_{jr} = \frac{x_{jr} - min(x_{jr})}{max(x_{jr}) - min(x_{jr})}, \quad x'_{jr} \in [0, 1]$

Logistic Regression (Sigmoid function) Discriminative classification

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-x^\top \beta}}$$

Decision rule:
If $P(y = 1 \mid x) \geq 0.5$ (same as saying $x^\top \beta \geq 0$), then predict class 1
If $P(y = 1 \mid x) < 0.5$ (same as saying $x^\top \beta < 0$), then predict class 0
Loss function:
Let $\hat{P}(y = 1|x) = h_\beta(x)$, then $J(\beta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(h_\beta(x^{(i)})\right) + (1 - y^{(i)}) \log \left(1 - h_\beta(x^{(i)})\right)\right]$

Bayesian Expected Loss $\mathbb{E}[L(\alpha_i)] = R(\alpha_i|x) = \sum_{h \in H} \lambda(\alpha_i|h) P(h|x)$ where α_i = action, h = hypothesis and λ = cost

Bernoulli Naive Bayes Classification Here, a is the feature

$$P(a|+) = \frac{\text{number of email that are + and have a}}{\text{number of email that are +}}$$

If test data = aabb, then $e = (1, 1, 0)$ and $P(x|+) = P(a|+)P(b|+)(1 - P(c|+))$, and $P(+|x) \propto P(x|+)P(+)$

Multinomial Naive Bayes Classification $P(a|+) = \frac{\text{total number of } a \text{ that are +}}{\text{total number of words that are +}}$ If test data = aabbc, then $e = (2, 2, 1)$ and $P(x|+) = \frac{n!}{x_1! x_2! x_3!} P(a|+)^{x_1} P(b|+)^{x_2} (1 - P(c|+))^{x_3}$ where x_1 =number of 'a' in test data $P(+|x) \propto P(x|+)P(+)$

Tree Learning $Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$
$GainRatio(S, A) = \frac{Gain(S,A)}{SplitEntropy(S,A)}$
$SplitEntropy(S, A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$

Perceptron Training Unsupervised **linear classifier** (unlike basic linear classification by finding centroids of each class and using vector methods to find weight 'w') $w = \sum_i^n \alpha_i y_i x_i$ Here, need to learn α_i (number of misclassification on instance i. On dual view, $\hat{y} = \text{sgn}\left(\sum_i^n \alpha_i y_i \langle x_i, x\rangle\right)$

Perceptron learning algorithm

Initialize $\alpha_i \leftarrow 0$ for all i
converged $\leftarrow$ false
while not converged **do**
  converged $\leftarrow$ true
  for $i = 1$ to $|D|$ **do**
    if $y_i \left(\sum_{j=1}^{|D|} \alpha_j y_j \langle x_j, x_i\rangle\right) \leq 0$ **then**
      $\alpha_i \leftarrow \alpha_i + 1$
      converged $\leftarrow$ false
    end if
  end for
end while

SVM algorithm A linear classifier

- $\arg\max_{\boldsymbol{\alpha}} \left(-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j G'[i,j] + \sum_{i=1}^{n}\alpha_i\right)$ subject to: $\sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \text{for } i = 1,\dots,n$ where $G' \equiv X'(X')^T$ and $X' = \begin{bmatrix} x_1^T y_1 \\ x_2^T y_2 \\ \vdots \\ x_n^T y_n \end{bmatrix} \in \mathbb{R}^{n\times p}$ (feature vector x_i is aligned horizontally), $n =$ number of samples
- reduce one α term using $\sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \geq 0$
- Compute partial derivative on the above equation w.r.t each α and set them to zero to solve for **all** α. (for support vectors, $\alpha_i \neq 0$)
- find **w** by $w = \sum_{i=1}^{n}\alpha_i y_i x_i$ where $x_i \in$ support vectors
- find **t** by $y_i(\langle w, x_i\rangle - t) = 1$ where x_i is one support vector
- find **margin** $m = \frac{1}{\|w\|}$

Prediction $\hat{y} = \text{sgn}(w\cdot x - t)$
Prediction is fast coz of spare support vectors (i.e. not all $\alpha_i > 0$)
For **non-linear** SVM, can use Kernel trick $\hat{y} = \text{sgn}\left(\sum_{\alpha_i>0}\alpha_i y_i K(x_i, x) - t\right)$

AdaBoost Input: data $D = (X, y)$, ensemble size T, learning algorithm A (decision stump) Output: weighted ensemble of models

1. Initialise weights: $w_{1i} \leftarrow 1/|D|$ for all $x_i \in D$
2. for $t = 1,\dots,T$:
   - run A on D with weights w_{ti} to produce a model M_t
   - calculate weighted error ϵ_t where $\epsilon_t = \sum_{i=1}^{n} w_{t,i}\mathbb{I}\{y_i \neq \hat{y}_i\}$ ($\mathbb{I}\{y_i \neq \hat{y}_i\}$ is equal to 1 if $y_i \neq \hat{y}_i$ and zero otherwise.)
   - $\alpha_t \leftarrow \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$
   - $w_{(t+1)i} \leftarrow \frac{w_{ti}}{2\epsilon_t}$ for misclassified instances $x_i \in D$
   - $w_{(t+1)j} \leftarrow \frac{w_{ti}}{2(1-\epsilon_t)}$ for correctly classified instances $x_j \in D$
3. return $M(x) = \text{sign}\left(\sum_{t=1}^{T}\alpha_t M_t(x)\right)$ (i.e. $M(x) \geq 0 \implies$ (+) class and $M(x) < 0 \implies$ (-) class)

Neural Learning

Cross-Entropy Loss For binary classification with $y_i \in \{0,1\}$, $L(w) = -\sum_{i=1}^{N}\left[y_i\log(p_i) + (1-y_i)\log(1-p_i)\right]$,where p_i is probability function which contains w

Gradient Descent at j-th layer i-th perceptron (node in NN): $w_{ji}^{(t+1)} = w_{ji}^{(t)} - \eta\left[\frac{\partial L(w)}{\partial w_{ji}^{(t)}}\right]$

PAC Learnable $P(error_D(h) \leq \epsilon) > 1-\delta$

Probability that Version Space is not ϵ exhausted $P(error_D(h) > \epsilon) < |H|e^{-\epsilon m} < \delta$ where m is the number of samples.
For probability to be below δ, need to fulfill
$m \geq \frac{1}{\epsilon}\left(\ln(|H|) + \ln\left(\frac{1}{\delta}\right)\right)$ (for finite Hypothesis space)
$m \geq \frac{1}{\epsilon}\left(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon)\right)$ (for infinite Hypothesis space with finite VC dimension)
where $m \propto H$, $m \propto \frac{1}{\epsilon}$ and $m \propto \frac{1}{\delta}$

For hypotheses that are not consistent (i.e. not part of Version Space) $P(testError_D(h_{best}) > trainError_D(h_{best}) + \epsilon) \leq |H|e^{-2m\epsilon^2} < \delta$
$m \geq \frac{1}{2\epsilon^2}\left(\ln(|H|) + \ln\left(\frac{1}{\delta}\right)\right)$ Need to fulfill

For linear classifiers For d dimensions (features), number of parameters for classifier is $d+1$, and VC dimension is also $d+1$.

For finite Hypothesis space For d data points where $VC(H) = d$, $|H| \geq 2^d \implies d \leq \log_2|H|$

Theorem H is PAC-learnable if and only if VC dimension is finite