COMP 551 Applied Machine Learning - Midterm Crib Sheet

Differentiation: $\partial(\mathbf{x}^T a)/\partial\mathbf{x} = \partial(a^T\mathbf{x})/\partial\mathbf{x} = a^T$, $\partial(\mathbf{x}^T A\mathbf{x})/\partial\mathbf{x} = \mathbf{x}^T A^T + \mathbf{x}^T A = \mathbf{x}^T(A^T + A)$, $\partial\|\mathbf{x}\|^2/\partial\mathbf{x} = 2x$
Overfitting: Overfitting models have high variance (and usually low bias); less training data ⇒ more likely to overfit (higher variance) and vice versa
k-fold cross validation: k partitions of training set. Train on $k-1$ subsets, validate with k^{th} subset. Repeat k times. Average errors over the k rounds/folds. **Complexity:** multiply by k
Leave-one-out cross validation: $k = n$. For each model/hyperparams setting, repeat n times { hold out 1 training ptn, get $\mathbf{w}$ from training data, get prediction error on held out training ptn}, average the pred. err. over all n subsets, choose setting with lowest estimated true pred. err.. **Complexity:** multiply by n
Accuracy: TP + TN / all predictions. TP = m11, TN = m00
Precision: TP / all declared pos = TP / (TP + FP). FP = m01 = type 1 err.
Recall/sensitivity: TP / all actual pos = TP / (TP + FN). FN = m10 = type 2 err.
Specificity: TN / all actual neg = TN / (TN + FP)
False positive rate: FP / (TN + FP)
F1 measure: $2\frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$
Linear regression:
Closed form: $\partial\text{Err}(\mathbf{w})/\partial w = -2X^T(y - X\mathbf{w}) = 0 \Rightarrow \hat{\mathbf{w}} = (X^TX)^{-1}X^TY \Rightarrow \hat{Y} = X\hat{\mathbf{w}}$, **complexity** ($n$ ptns & m feats): $O(m^3 + nm^2)$
SGD: $\text{Err}(\mathbf{w}) = (y - X\mathbf{w})^T(y - X\mathbf{w}) = \sum_i (y_i - X_i\mathbf{w})^2$, $\partial\text{Err}(\mathbf{w})/\partial w = 2(X^TX\mathbf{w} - X^Ty)$, $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha\partial\text{Err}(\mathbf{w}^{(t)})/\partial\mathbf{w}^{(t)}$. **Complexity:** $O(mn)$ for each weight update
Regularization: introduce penalty to error ⇒ introduce bias & reduce variance. High penalty ⇒ high bias & low variance and vice versa
L2 regularization/ridge regression: $\text{Err}_{L2}(\mathbf{w}) = \text{Err}(\mathbf{w}) + \lambda\|\mathbf{w}\|_2$, $\partial\text{Err}_{L2}(\mathbf{w})/\partial\mathbf{w} = \partial\text{Err}(\mathbf{w})/\partial\mathbf{w} + 2\lambda\mathbf{w}$
L1 regularization/lasso regression: Same as L2 with $\lambda\|\mathbf{w}\|_1$ and $\lambda\times\text{sign}(\mathbf{w})$, $\text{sign}([-10, 0, 10]) = [-1, 0, 1]$
Linear classification:
Discriminative learning: Directly estimate $\text{P}(y|\mathbf{x})$
Logistic regression: Probabilistic model/model defining a decision boundary. Log-odds $a = \ln\frac{\text{P}(y=1|x)}{\text{P}(y=0|x)} = b + w_1x_1 + ... + w_mx_m$, **decision boundary:** set of points s.t. $a = 0$, $\text{P}(y = 1|x) = \sigma(\mathbf{w}^T\mathbf{x})$, for $y \in \{0, 1\}$, **likelihood** (want maximize) $L(D) = \text{P}(y|\mathbf{x}, \mathbf{w}) = \prod_{i=1}^n \sigma(\mathbf{w}^T\mathbf{x})^{y_i}(1-\sigma(\mathbf{w}^T\mathbf{x}))^{1-y_i}$, **log-likelihood** $l(D) = \ln L(D) = \sum_i^n \left[y_i \ln\sigma(\mathbf{w}^T\mathbf{x}) + (1 - y_i)\ln(1 - \sigma(\mathbf{w}^T\mathbf{x}))\right]$, **cross-entropy loss** (want min) $\text{CE}(D) = -l(D)$, $\partial\text{Err}(\mathbf{w})/\partial\mathbf{w} = \sum_i^n \mathbf{x}_i(y_i - \sigma(\mathbf{w}^T\mathbf{x}_i))$, $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha_t\partial\text{Err}(\mathbf{w})/\partial\mathbf{w}$, **# params**: 1/ft + bias term
Generative learning: Separately model $\text{P}(\mathbf{x}|y)$ and $\text{P}(y)$, use Bayes' rule to estimate $\text{P}(y|\mathbf{x})$: $\text{P}(y = 1|\mathbf{x}) = \frac{\text{P}(\mathbf{x}|y=1)\text{P}(y=1)}{\text{P}(\mathbf{x})}$
LDA: $\text{P}(\mathbf{x}|y = k) = \frac{\exp\{-(x-\mu_k)^T\sum^{-1}(x-\mu_k)/2\}}{(2\pi)^{1/2}|\sum|^{1/2}}$. k classes, m fts.: k mean vectors + $m \times m$ shared covariance mat. $= km + m^2$ params.
Bin. class.: log-odds ratio $\delta(\mathbf{x}) = \ln\frac{\text{P}(y=1)}{\text{P}(y=0)} - \frac{1}{2}(\mathbf{x} - \mu_1)^T\sum^{-1}(\mathbf{x} - \mu_1) + \frac{1}{2}(\mathbf{x} - \mu_0)^T\sum^{-1}(\mathbf{x} - \mu_0)$
$= \left(\ln\frac{\text{P}(y=1)}{\text{P}(y=0)} - \frac{1}{2}\mu_1^T\sum^{-1}\mu_1 + \frac{1}{2}\mu_0^T\sum^{-1}\mu_0\right) + \mathbf{x}^T\sum^{-1}(\mu_1 - \mu_0) = b + \mathbf{w}^T\mathbf{x}$, b scalar and $\mathbf{w}$ a linear parameter vector
QDA: Like LDA but has unique cov. mat. $\sum_k$ for each class $y = k$. $km + km^2$ params
Bernoulli Naive Bayes: $\delta(x) = \ln\frac{\text{P}(y=1|x)}{\text{P}(y=0|x)} = \ln\frac{\text{P}(x|y=1)\text{P}(y=1)}{\text{P}(x|y=0)\text{P}(y=0)}$ (expnd w. θ prms) $= \ln\frac{\text{P}(y=1)}{\text{P}(y=0)} + \sum_j\left[\ln\frac{1-\theta_{j,1}}{1-\theta_{j,0}}(1 - x_j) + \ln\frac{\theta_{j,1}}{\theta_{j,0}}x_j\right]$
$= \ln\frac{\theta_1}{1-\theta_1} + \sum_j \ln\frac{1-\theta_{j,1}}{1-\theta_{j,0}} + \sum_j\left(\ln\frac{\theta_{j,1}}{\theta_{j,0}} - \ln\frac{1-\theta_{j,1}}{1-\theta_{j,0}}\right)x_j = \left(\ln\frac{\theta_1}{1-\theta_1} + \sum_j w_{j,0}\right) + \sum_j(w_{j,1} - w_{j,0})x_j = b + \mathbf{w}^Tx$
$\text{P}(y = 1|x) = \frac{\text{P}(x,y=1)}{\text{P}(x)} = \frac{\text{P}(x|y=1)\text{P}(y=1)}{\text{P}(x|y=0)+\text{P}(x|y=1)}$. **# params:** 2/ft
Laplace smoothing: Add bias/reduce variance (or overfitting) for NB, $\theta_{j,1} = \frac{(\#\text{ ptns w. } x_j=1, y=1)+k}{(\#\text{ ptns w. } y=1)+k+1}$, $k = 1$ "add-one", else "add-k"
Gaussian Naive Bayes: Like QDA, unique **but diagonal** cov. mats.. $2km$ params
Naive Bayes: k classes ⇒ **# params**: $k - 1 + km$, $k - 1$ for $\text{P}(y)$ and km for all $\text{P}(x_i|y = k)$
Decision tree: Depth can't be > training examples: every valid test must split the data ⇒ in worst case there's 1 test/training ptn. 2 steps: top-down growing, pruning. **Top-down (recursive) induction: 1.** If all training ptns have same class, create a leaf w. that cls. label and exit. Else **2.** Pick best test to split data on **3.** Split the training set according to the value of the outcome of the test **4.** Recursively repeat steps 1 - 3 on each subset of the training data. **Choosing best test:** for classification choose test w. highest info gain, for regression lowest MSE. **Advantages:** provide gen rep of classification rules (normalization not needed: no "distance" b/w ptns), learned fn is easy to interp, fast learning alg, good acc in practice. **Disadvantages:** output sensitive to small change in train data, tests may not be meaningful with many features, not good for learning fns w. smooth, curvilinear boundaries (but good for non-linear piecewise axis-orthogonal dec bounds.)
Information content: $I(E) = \log_2\frac{1}{\text{P}(E)}$ bits of information for event E. Amt. of "surprise" from outcome e.g. fair coin flip $\log_2 2 = 1$ bit of info., fair dice roll $\log_2 6 \approx 2.58$
Entropy: $H(S) = H(P) = \sum_i p_iI(s_i) = \sum_i p_i\log_2\frac{1}{p_i} = -\sum_i p_i\log_2 p_i$ S info. source, s_i symbol, p_i prob. of symbol s_i. Avg. amt. of info. per symbol **e.g.** loaded die 75% to roll 6 gives $-.75\log_2.75 + (5)(-.05\log_2.05) \approx 1.392$
Bin. classification: $H(D) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$ entropy of dataset, p pos ptns, n neg ptns
Conditional entropy: $H(y|x) = \sum_v \text{P}(x = v)H(y|x = v)$
Information gain: $IG(x) = H(D) - H(D|x)$. Higher is better
Early stopping: Stop growing tree when further data splitting doesn't improve info. gain of validation set
Post-pruning: Reduce variance/combat overfitting for dec. trees. **1.** Split dataset into train and validation sets **2.** Grow large tree e.g. until each leaf is pure **3.** For each node { eval valid set acc of pruning node's subtree, greedily remove node that most improves

valid acc and its subtree, replace rmved node by leaf w the majority class of the corres examples } **4.** Stop when pruning hurts valid acc

Perceptron: $\hat{y} = h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{w}) = \{1 \text{ if } \mathbf{w}^T\mathbf{x} \geq 0; -1 \text{ otherwise}\}$, decision boundary $\mathbf{w}^T\mathbf{x} = 0$, prediction correct iff $y_i\mathbf{w}^T\mathbf{x}_i \geq 0$, $\text{Err}(\mathbf{w}) = \sum_{i=1:n}\{0 if y_i\mathbf{w}^T\mathbf{x} \geq 0; -y_i\mathbf{w}^T\mathbf{x} \text{ otherwise}\}$, **linearly sep.** iff $\forall \mathbf{x}_i, y_i\mathbf{w}^T\mathbf{x}_i > 0$, **perceptrion cvg. th.:** linear sep. $\Rightarrow \exists$ correct perceptron for whole dataset after finite updates i.e. not linear sep. $\Rightarrow \nexists$ correct perceptron

Alg: Init $\mathbf{w}$ randomly; While $\exists$ misclassified ptns { $\forall$ misclassified x_i { $\mathbf{w} \mathrel{+}= \alpha y_i x_i$ } }

Hard SVM:

Soft SVM: dual and primal problem and what slack and variables do to the soft clustering.

Kernel trick:

kNN: Reduce variance/combat overfitting by **increasing** k

Distance weighted (kernel-based) NN:

Activation functions: $\sigma(x) = 1/(1+e^x)$, $\partial\sigma(x)/\partial x = \sigma(x)(1-\sigma(x))$

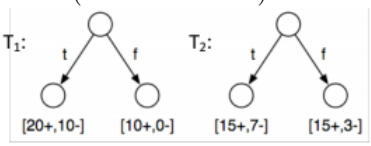
FFNN:

CNN:

RNN:

LSTM:

Misc.: *cos* **loss fn guarantees gradescent cvg?** No. Deriv. *cos* is $-sin$, which oscillates between -1 and 1 as x increase/decreases, so has no unique extremum (i.e. non-convex).



What is the entropy of the dataset?

- $H(D) = -(3/4)\log_2(3/4)-(1/4)\log_2(1/4) = 0.811$

What is the conditional entropy of the dataset given the different tests?

- $H(D|T_1) = (30/40)[-(20/30)\log_2(20/30)-(10/30)\log_2(10/30)]+(10/40)[0] = 0.688$
- $H(D|T_2) = (22/40)[-(15/22)\log_2(15/22)-(7/22)\log_2(7/22)]$
  $+ (18/40)[-(15/18)\log_2(15/18)-(3/18)\log_2(3/18)] = 0.788$