CS434 Machine Learning and Data Mining Final

Study online at https://quizlet.com/_9x55w3

How does multiclass classification differ from binary classification?	Binary classification is a special class of multiclass classification when the number of classes equals two.
What is a one-vs-all classifier?	A binary classifier that learns a decision boundary separating one class (the class of interest) from all other classes.
What are pros of one-vs-all classifiers?	Only need to train C classifiers, and probabilistic classifiers allow taking the argmax to determine class in overlap region.
What are cons of one-vs-all classifiers?	If the classifier is binary, then overlap regions must be decided by randomly choosing a class.
What is an all-vs-all classifier?	A composite of (C choose 2) classifiers used to separate each pair of classes.
How is an all-vs-all classifier used to make predictions?	Run all models on a new input and see which class is predicted the most frequently.
What are pros of all-vs-all classifiers?	Simple to implement, and rarely ties between classes.
What are cons of all-vs-all classifiers?	Weaker error bound than one-vs-all classifiers, and you have to learn and run many classifiers.
What is a tree classifier?	Train half-vs-half classifiers to form a binary tree of decision boundaries. Follow the decision path to classify a new input
What are pros of tree classifiers?	Strong error bound (log_2 K)*µand not too many classifiers to learn
What are cons of tree classifiers?	Have to select splits somehow (good or bad)
What does multiclass logistic regression learn?	Multiple linear (piecewise) decision boundaries in input space, where the max probability is taken as the class label
What is multiclass logistic regression?	Take the maximum entry (corresponding to class c) for the vector of probabilities. This is a single layer of neurons with softmax activation.
What other methods can do multiclass classification directly?	kNN, neural networks, multiclass SVMs, multiclass perceptrons, decision trees, naive Baye
What is a neuron?	A linear function of its input followed by a (typically) nonlinear activation function producing output.
What are the hyperparameter(s) of a single neuron?	The activation function, Ã
What learnable parameters does a single neuron have?	w R^d and b R
What do we do to form a neural network?	Stack multiple layers of neurons together
What type of neural network did we study in depth?	Multilayer, feedforward neural networks
What activation do we use in the output layer for regression?	Linear activation
What activation do we use in the output layer for classification?	Sigmoid or softmax
How do we compute the output for a single neuron?	Take the activation function of the summed, weighted input plus bias. Thus, simply compute Ãi(Σwi*xj) + bi)
How do we train a NN?	Forward pass, backward pass, then update.
What happens in a forward pass?	For each training example, (i) compute and store all activations, then (ii) compute loss
What happens in a backward pass?	Compute gradient of the loss w.r.t. all network parameters. Perform this efficiently using backpropagation
What happens during the update step?	A step of gradient descent is performed to minimize loss
What can we use NNs for?	Learning nonlinear functions, learning theoretically universal approximators to arbitrarily low error µand they're flexible enough to be used for classification or regression
What is a loss function?	Some function measuring how bad a network's output is relative to optimal performance
What sorts of loss functions are used for regression?	Squared error (most common), absolute error, or Huber error
What sorts of loss functions are used for classification?	Cross entropy or hinge error

What observations does backpropagation use to be so effective?	1) NNs generally reduce dimensionality: high-->low. 2) We shouldn't recompute something we already know (memoization via DP)
When do we use forward-mode differentiation?	When the input dimension is much smaller than the output dimension
When do we use reverse-mode differentiation?	When the dimension of output is much smaller than the dimension of the input
Where do we compute loss?	At the very END of the neural network (output)
What is a computational graph?	A directed acyclic graph (DAG) with vertices corresponding to computation and edges to intermediate results
What operations need to be defined for each node in backpropagation?	The forward computation y1,...,yk = f(x1,...,xk) and the backward computation yi/ xj i,j
Why is backpropagation an efficient way to compute gradients?	Computes each gradient only once!
What is stochastic gradient descent?	Random initialization of weight vectors when starting a NN. Gradient descent, but using a small sample of points to estimate gradient
Why is initialization important in neural network traning?	Depending on the activation, init might make the gradient == 0, so now you aren't training!
How does training behavior change with different levels of learning rate?	Maximum training optimality at an intermediate learning rate. Too low, learning becomes very slow. Too high, system jumps around and cannot fall into minima
How do we perform discrete convolutions?	For an Nd system, take an up-to-Nd filter/kernel and slide it around the system, recording features such as the sum of filter product elements, etc. Can also do this across the time domain to compute the activation for that feature
What assumptions do convolutional kernels make?	1) Locality: only need to look at certain blocks of the input to find the cat face, and 2) Translational invariance: Doesn't matter where it occurs: it will look the same
What type of computation do RNNs model?	Memory retention about a problem
What do internal nodes do in a decision tree?	Perform tests against a single input feature value.
What does each branch from an internal node represent?	A splitting by feature values or ranges of feature values
What is the function of leaf nodes in decision trees?	Predicts output: either class or continuous outputs
What are some benefits of decision trees?	Their resulting models are highly human-interprable, they can fit arbitratily complex functions, and they can use a mix of discrete and continuous inputs
How many features can DTs use?	Up to all of the input features, although not all features must be included in the DT
May a single feature appear in multiple DT branches?	Yes, this is possible if a feature is split incompletely or on continous values. Repeatedly splitting on discrete features does nothing, though
Are DTs unique?	No, many trees represent the same logical equation, but they may not have the same size. Thus, it's best to select the smallest possible DT with proper logical separation, although this is NP-hard
Why do we want a small decision tree?	Small trees embody the prior of simplicity, and thus they're less likely to overfit
Apart from NP-hard implementations, what should we do?	Build the tree greedily!
What heuristic should we use when greedily constructing a decision tree?	Information gain on conditional splits on labels
How do we deal with splits on continuous features?	There are INFINITELY-many such splits. IG only changes when a threshold crosses at a datapoint. Thus, we sort the dataset values and consider thresholds between consecutive datapoints.

	we compute IG for each threshold and select the threshold with maximum IG
How do we avoid overfitting on DTs?	1) Add hyperparameters to control tree size (depth, #nodes, min_split_number), 2) Stop early based on validation performance (when validation saturates), 3) Post pruning
How does post pruning work for DTs?	Grow the full DT on the training set, then consider the impact of removing each node on validation performance. Greedily prune the node that improves validation performance the most
Why are decision boundaries composed of axis-aligned segments?	Decision trees separate based on single variables, not functions of multiple variables.
What is the maximum depth of a decision tree?	The number of unique splits
When would a leaf of a fully-expanded DT have more than one datapoint in the leaf?	If Bayes error is NOT zero; need to add another variable to split on!
How are decision trees trained?	Taking a dataset and splitting on certain variables. Variables to split on are selected with information gain (highest IG of all the given splits, selected greedily)
Does the DT training algorithm find the optimal tree every time?	NO, it's greedy though and should get a decently good one.
What is entropy?	A measure of inhomogeneity of class labels in a dataset
What is conditional entropy?	A measure of inhomogeneity of class labels in a dataset when splitting the dataset based on a variable, with respect to the variable that is being split on
What is information gain?	Entropy minus conditional entropy
Why do we rank splits by information gain?	It maximizes the likelihood of a split separating fully based on class labels
How can decision trees handle continuous variables?	Split between each individual continuous value, and determine which split is highest
What is bias?	Error due to assumptions in the model not matching the problem (modelling error). Really, average error between function and the model over all possible datasets.
What is variance?	Error due to sensitivity to changes in the dataset (estimation + opt error). Variance of error between model and the true function over all possible datasets
What sort of model is a second-order polynomial fit?	Low variance, high bias
What sort of model is a 20th-order polynomial?	High variance, low bias
Examples of LV,HB learners ("weak")	Naive Bayes, LogReg, Shallow DTs
Examples of HV,LB learners ("strong")	Kernel methods, NNs, deep DTs, kNN
Tradeoffs of weak learners	Don't often overfit (low variance), but can't represent complex functions (high bias)
Tradeoffs of strong learners	Can learn complex functions (low bias), easy to overfit (high variance)
Why do "weak" models tend to have high bias but low variance?	They can't fit the data well because they make simplifying assumptions about the model
Why do "strong" models tend to have low bias but high variance?	They tend to overfit to a training dataset. Low-D parameters fit well, but dimensions irrelevant to fitting the dataset vary highly because there's nothing to fit them to.
How do bias and variance relate to sources of error we discussed earlier in the course?	Variance includes optimization and estimation error, while bias includes modelling error
How do ensembles work?	Final output of the "ensemble" of models is a combination of each model's output.
When do ensemble methods work well?	When each individual member of the ensemble is (i) accurate (better than chance) and (ii) diverse (uncorrelated errors on new examples)

How does bagging work?	Tries to reduce variance in strong learners. For uncorrelated errors, expected error goes down, and on average they do better than a single classifier
How does boosting work?	Tries to reduce bias in weak learners. Each model is good at different parts of input space, and on average they do better than a single classifier
How do we do bagging?	Given a dataset of N points, sample N training points with replacement and train a model. Repeat this M times
How does bagging work with regression?	At test time, run each model and average their output
How does bagging work with classification?	Take the majority vote
When should we use bagging?	When your strong learners overfit the dataset, and when you have a reasonably-sized dataset
What's a convenient implementation idea unique to bagging?	Each ensemble member can be trained in parallel, rather than in sequence. Boosting requires sequential training
How do random forests work?	1) Train an ensemble of decision trees with bagging. 2) During learning, at each split only consider a random subset of attributes/thresholds when selecting what to split on (d features used often). 3.) Select majority vote from forest
How do we re-weight training data when boosting?	Assign importance values, inversely proportional to the weighted sum of distances to points which are not fit well as you add new models.
How do we weight models in the "committee" when boosting?	Weight each model based on the error rate (really, classifier quality)
Does bagging help reduce bias or variance?	Bagging reduces variance, especially in strong learners
How does the correlation between model outputs effect the performance of a bagged ensemble?	As correlation goes down, ensemble performance increase goes up
Why would we want to introduce additional randomness in Random Forests?	The greedy decision tree learning algorithm will likely use the same attributes early on, despite resampling.
Does boosting help reduce bias or variance?	Boosting reduces bias, especially in weak learners
How does L2 boosting work?	We search for the models and weights that minimize error over the final model
How does Adaboost work?	Initialize importance weights to 1. For each model,{ train the classifier to minimize weighted exponential loss. Then compute the weighted error of the classifier, and the classifier quality. Update weights }
What two characteristics describe clustering techniques?	Grouping all given examples (exhaustive) into disjoint clusters (partitional) such that within-cluster examples are similar and without-cluster examples are different
What are partition algorithms?	"Flat" clusterings, whereby each point belongs to exactly one cluster. E.g.: k-means, k-medioids, Gaussian mixture models (GMMs)
What are hierarchical algorithms for clustering?	"Hilly" clusterings, where by each point often belongs to more than one cluster, and local internal clustering is common. Bottom-up is agglomerative, and top-down is divisive
What is the k-means clustering algorithm?	Initialize k centroids randomly. While the algorithm hasn't converged, take the average position of all the points in each centroid, and assign this point to be the new centroid center. Then, return the centroids and associations
How do we choose k for k-means?	Look for the elbow/knee on a graph of SSE vs k
What is a disadvantage of k-means?	It's highly initialization-dependent
How so we deal with the sentisitivity of k-means?	Run multiple times at the same value of k, then take the clustering with the lowest SSE
How do we initialize centroids "well"?	Make the cluster centers as far apart as possible

How does k-means handle outliers?	Extremely sensitive, because each datapoint needs to have a cluster.
How do we deal with k-means' sensitivity to outliers?	Run k-medioids instead of k-means
Where does k-means perform poorly?	When there's outliers, when data is not spherically distributed, and when clusters are of different sizes
Is SSE a good indicator of model convergence for k-means?	No! As k->N, SSE -> 0.
How long does it take k-means to converge, and whre does it converge?	Takes finite number of steps (few iterations in practice), to reach a local minimum
What is k-means' computational complexity?	O(mknd)
What are downsides of k-means?	Not outlier-resistant, may need to run several times to get a low SSE, and only works well on spherical clusters of similar size
What is the expectation maximization algorithm?	First, initialize probabilities of each point being in each gaussian, means of gaussians to random points, and covariances to identity matrices. In the E-step, compute fractional assignment of each point being generated from class c, and normalize to 1. In the M-step, update the parameters based on fractional assignment
What is a problem with using GMMs?	Because log-likelihood can go to infinity if one gaussian has only one point assigned. To solve this, monitor each component, reset to a random value if it starts to collapse (with large covariance), OR add a prior and do MAP in the M-step of EM
What is the identifiability problem for GMMs?	Log-likelihood in GMMs has multiple identical maxima. No easy fix, but of limited harm, although it may slow convergence.
How does EM behave when constructing GMMs?	Guaranteed to converge to a local maxima in finitely many steps, although may be slow. Not guaranteed to find global optima.
How do we deal with nonoptimality in GMMs?	Restart multiple times, and choose the one with the highest log-likelihood
How do we interpret GMMs?	They produce a full density model of the data, so you can sample new synthetic data or evaluate the probability at an untested point
How do we get from GMMs to k-means?	1) Assume a hard-assignment frather than fractional/probabilistic, and 2) assume all gaussians have the same isotropic covariance matrix
What is clustering?	An unsupervised method for roughly separating probable classes by groupings in input space
What is k-means?	An algorithm for performing unsupervised clustering
How is k-means implemented?	Initialize k centroids. Iterate through centroids, taking the mean of the data assigned to each as the new, updated position for each centroid ci. Continue until no updates to labels.
What does k-means optimize?	Sum of squared error for a given number of clusters, k
How is coordinate descent related to the k-means algorithm?	Coordinate descent optimizes two variables by switching between them and performing gradient ascent for log likelihood.
Is the k-means algorithm guaranteed to converge? If so, to local or global optima?	Yes, in finitely-many steps, but to a local minimum.
How do we pick hyperparameters for k-means?	By observing the "elbow" on the SSE-vs-k graph
Is k-means sensitive to outliers?	Yes, very much so!
Is k-means sensitive to initialization?	Yes, to find an optimal clustering, run multiple times and find the one with lowest SSE
What types of clusters does k-means work best for?	Isotropic, nonoverlapping spherical clusters
What sort of model does GMM assume generated the data?	Gaussian model! Not necessarily isotropic, and not necessarily the same size
What can GMM do that k-means can't?	Model wide/long distributions, model a collection of distributions of different sizes/radii

CS434 Machine Learning and Data Mining Final

Study online at https://quizlet.com/_9x55w3

Why is maximum marginal likelihood difficult to optimize?	Exact solutions are only known for a small class of distributions, so numerical integration techniques are often needed
What is the Expectation Maximization algorithm do at a broad level?	Attempts to search for the global maximum of log likelihood for generating the data that's provided according to some mixture of gaussians
What are some challenges in GMM optimization?	Identical distributions can create a very challenging optimization
What assumptions must be made in GMM in order to recover the k-means algorithm?	Isotropic distributions, spherical distributions
How does HAC work?	Initialize each point as its own cluster. While more than one cluster remains, merge the two closest clusters.
How do you measure distance between different clusters?	Apply a link function to the two clusters of interest
What is a "link function"?	A class of functions which measures the distance between multiple clusters. Examples are single-link, complete-link, centroid, and average-link. Average link is most robust and most often used
How do the different link functions behave?	Single link tends to make long, snaky clusters. Complete link tends to ??? Centroid works well with spherically-shaped data. Average link is highly robust, and thus most commonly used
What is a dendrogram?	A visualization of a clustering process using hierarchical agglomerative clustering.
How to read dendrograms?	By selecting different height cutoff values we can see clear clusters. Distance has a natural notion for separating outliers and far-apart clusters.
How to create "flat" clusterings from HAC?	Cutting at a desired value when clusters are well-separated
How do we evaluate whether a clustering is "good" without external validation?	Visual inspection (CAUTION) and an internal criterion like within-cluster and between-cluster similarities. These metrics depend on the dataset and the distance measure used
How do we evaluate clustering quality if we were spontaneously given labels after clustering?	Rand index or purity.
How do you compute Rand Index?	RI = (a+d)/(a+b+c+d) (a: in the same group in both P and G (same cluster, same labels); b: in the same group in P but different in G (same cluster, different labels); c: in different groups in P but same in G (different cluster, same labels); d: in different groups in both P and G (different clusters, different labels))
How is purity computed in HAC?	Purity = fraction of points that would be labelled correcly by a majority vote per cluster where all points get the cluster label
What is PCA? What sort of subspaces can it find?	A dimensionality-reduction technique that is capable of finding linear projections
How does PCA choose directions for the lower dimension representation?	PCA keeps the dimension with the greater variance, i.e. the dimension that varies most, it selects this dimension to retain
How can PCA be solved?	Solve the eigenvector problem using the covariance matrix: £x@w = »@w. Sort both eigenvalues and eigenvectors in descending order by eigenvalues, return eigenvectors corresponding to the top k.
How can we see what fraction of variance a dimension of PCA's output captures?	Plotting a graph of fraction of variance explained by a certain principal component to judge how represenative the visualization is
What sort of relationships can PCA not identify?	Parallel relationships where data points have labels
Why might applying PCA as a preprocessing step lead to issue?	It may reduce variance of interest in the dataset, because it was not examined by a person beforehand
What is a sequential decision process and how does it differ from standard machine learning problems?	Sequential decision making tasks' outputs changes the next input that is received.
What is a Markov Decision Process?	

	A MDS is a decision process defined by a set of possible states S, a set of possible actions A, a reward function R:S' , and a transition function T:SxA©S. It is defined by the Markov assumption
What is a state in a MDS?	A place where the learning agent can exist or occupy in its environment
What is an action in a MDS?	A possible decision the agent can take, not necessarily but frequently impacting the environment and the success or failure of the agent
What is a transition function in a MDS?	A mapping between state-action pairs and a distribution over next states
What is a reward function in a MDS?	A mapping between states and rewards
What is a behavior policy in a MDS?	A mapping from states to actions (or distributions over actions)
What is model-free vs model-based reinforcement learning?	Model-free directly learns a policy, while model-based learns the transition function and reward function from observation and then use these to find best policy
What is the principal behind policy-gradient methods like REINFORCE?	If trajectory reward is positive, push up the probabilities of all the actions leading to that trajectory. If negative, push down those action probabilities
What is imitation learning and how does it differ from reinforcement learning?	Rather than using feedback from the environment, IL attempts to mimic an expert demonstration of a trajectory set
What is behavior cloning and what are its drawbacks?	A type of imitation learning. Very simple, and works well when minimizing one-step deviation is sufficient. Unfortunately, errors compound and data distribution mismatch can interfere with training
When should behavior cloning be used?	When the state space is well-covered by the demonstrator, when recovering from 1-step deviations is easy, and to pre-train before doing a full RL approach
Methods that can turn any binary classifier into a multiclass classifier	One-vs-all, all-vs-all, and tree classifier. While general, this can require training quite a few models.
Softmax function	Takes in a d-dimensional vector of exponentiated activations which contains the activation for each of d classes, with each class activation divided by the sum of all d activations. This outputs a normalized d-dimensional vector such that all elements sum to one.
In multiclass logistic regression, the total number of learned weight vectors for a C class problem.	C or C-1
Artificial neurons	Behave very differently than their biological counterpart
Capability of single-activation neuron	Can represent non-linear boundaries in classification problems.
ReLUs	Tends to converge fast because they do not saturate for positive inputs like Sigmoid.
Loss function	Measures how big an error predicted by a network is from the ground truth. Generally decreases when plotted versus number of epochs when training a neural network.
Jacobian	A matrix filled with partial derivatives of each dimension of vector v with respect to each dimension of vector u. Thus, it's a high-dimensional gradient.
Backpropagation	An efficient way to compute gradients of the loss with respect to model parameters. Firstly, it is a reverse-mode differentiation and computes the product of intermediate Jacobians from the output of the network backwards -- reducing cost of matrix multiplication in computing gradients. Secondly and more important, the backwards pass allows us to store and reuse loss gradients as we work our way backwards through the network.
Neural networks	Universal approximators of finite size

Computational graph	A directed acyclic graph with node corresponding to units of computation and edges corresponding to the results of these computations. If each node implements both a forward and backward (i.e. computing a Jacobian of output with respect to input) operation, then the graph can be used to calculate derivatives of the output of arbitrary combinations of operations via backpropagation.
Locality assumption of convolutional neural networks	Relevant features in an input can be found by examining local regions of the input -- e.g. windows of time in a sequence or regions of an image.
Decision trees	Tree of nodes, where internal nodes implement tests of attributes to divide datapoints and leaves determine labels to be predicted.
Why should decision tree learning not terminate whenever all all attributes result in zero information gain?	Many functions (like XOR) may not show any information gain in initial variables but be able to be usefully split later only after multiple variables have been considered.
Tasks in unsupervised learning	Density estimation, clustering, and dimensionality reduction
Single-Link function	Able to generate very long clusters because it only considers the minimum distance between points in two clusters for merging.
The height of a joint in a Dendrogram	The distance between two merged clusters.