Introduction

Below, we outline some key concepts and ideas you should be comfortable with for the final exam. Under each topic, we have divided this into two or three sections:

1. Items to know
2. Things to be able to work through when given information/formulae
3. Items out of scope

Note that the third section is only defined for some topics. This list is not exhaustive, but we hope it is illustrative. We encourage you to review the course textbook, section materials, homework materials, and the practice exam problems for a full picture.

1 Regression

1.1 Items to know

- What is the "bias trick" for rolling the bias into $\mathbf{w}$?
- What is the least squares loss function?
- How is $\mathbf{w}^*$ derived in a linear regression problem?
- What is a basis function, and why would we use it?
- What is regularization, and when do we use it? How does the loss function change for Lasso and ridge regression?
- What is the bias variance tradeoff, and how does it relate to overfitting, underfitting, regularization, and increasing the size of the training set?
- In Bayesian linear regression, how are priors related to regularization?
- What is a posterior distribution, posterior predictive distribution, and marginal likelihood?

1.2 Things to be able to work through when given information/equations

- You should be familiar with the mathematical techniques required to find $\mathbf{w}_{ridge}$. Sketch the derivation, and how is this similar to the derivation for $\mathbf{w}^*$ for OLS?
- Given a likelihood function (you do *not* need to remember PDFs), how is the MLE computed?
- Given a Bayesian linear regression setup, how would the MAP be computed?
- Given expressions for conjugate distributions, and forms of posteriors, work through finding a posterior parameter or the posterior predictive for a given setting.

1.3 Items out of scope

- Second-order or other optimization methods (anything other than simple closed-form optimization and the gradient descent setup)
- Deriving conjugacies or using conjugacy statements from memory (e.g. Normal-Normal).

2 Classification

2.1 Items to know

- Role of hinge loss vs 0/1 loss vs logistic loss?
- What is the idea behind gradient descent (i.e. intuitively, why does it work)? What happens if the learning rate is too high? Too low?
- What are batch gradient descent and stochastic gradient descent?
- If the data are linearly separable, are there any guarantees about perceptron's behavior? What is the intuition behind this?
- What can we say about the shape of the decision boundary of a logistic regression?
- Can you explain all the parts of the likelihood function for two-class and multi-class logistic regression?
- How do we work with generative models for classification? What is the Naive Bayes model?
- What is the difference between a discriminative model like Logistic Regression and a generative model like Naive Bayes?

2.2 Things to be able to work through when given information/equations

- Gradients for negated log likelihood of logistic regression
- For a generative model and give class priors and class-conditional distributions, work with the (full) log-likelihood (using Lagrangian method where needed)
- Given parametric as well as non-parametric classifiers (such as kNN) and a particular dataset, how do we expect different classifiers to perform?
- Performing multiple, numerical steps of an iterative optimization such as gradient descent.
- Deriving basis functions to represent data in a space where it is separable (this is in scope for simple cases where such a transformation is easily discernible).

3 Neural Networks & Model Selection

3.1 Items to know

- Why do we need activation functions?
- What is backpropagation, and why is it useful?
- How can model selection methods such as cross-validation, and regularization be useful in supervised learning?
- How can neural networks be used for regression tasks and how can they be used for classification tasks?

3.2 Things to be able to work through when given information/equations

- Given a particular task for a neural network, what is a good choice for a loss function (e.g., least squares vs softmax)?
- Given a particular neural network architecture, how might changes to architecture (e.g. adding a layer, changing the structure) intuitively affect the model's performance?
- Given a simple neural network, determine whether the model is able to perform a particular classification task succesfully.
- Given a simple (not necessarily named) loss function for a neural network, deduce why it is intuitive for the task at hand.
- Deriving backpropagation computations by hand

4 Support Vector Machines (SVMs)

4.1 Items to know

- What is the maximum margin classifier, and how does it relate to SVMs?
- What is the role of the hinge loss function in SVMs?
- What is the difference between hard-margin and soft-margin SVMs?
- How does the choice of the regularization parameter C affect the classifier?
- What is the kernel trick, and why is it useful?
- What are some common kernels (e.g. linear, polynomial, RBF), and how do they work?
- How does an SVM decision boundary compare to that of logistic regression?
- What are support vectors, and why are they important in SVMs?

4.2 Things to be able to work through when given information/equations

- Given a set of data points, determine the maximum margin classifier.
- Given a soft-margin SVM formulation, explain the role of slack variables and how they impact classification.
- Given a kernel function, compute the transformed feature space dimensions and explain its effect.
- Understand how SVMs handle non-linearly separable data using kernel functions.

4.3 Items out of scope

- Work through the dual formulation of an SVM and interpret the meaning of Lagrange multipliers.
- Solving the full quadratic optimization problem manually.
- Derivation of the Karush-Kuhn-Tucker (KKT) conditions for SVMs.

5 Clustering

5.1 Items to know

- K-means objective, Lloyd's algorithm
- Hierarchical agglomerative clustering, dendrograms, and the various linkage criteria. How do the different linkage criteria affect what kinds of clusters are learned?
- Comparing K-means to HAC

5.2 Things to be able to work through when given information/equations

- Modifications/extensions to K-means and HAC, e.g. K-means++ and K-medoids

6 Mixture Models and Topic Models

6.1 Items to know

- Expectation maximization: understand the motivation (MLE problem), the E-step, and the M-step. Be familiar with the ELBO, but no need to memorize its derivation
- Understand the typical kinds of applications of mixture models
- Understand the general setup for mixture models
- How to express a model as a directed graphical model (DGM); idea of latent vs. observed variables, handling parameters as random variables, and the plate notation (see Graphical models lecture).

6.2 Things to be able to work through when given information/equations

- Understand how to work with the Gaussian mixture model (no need to memorize its specific form).
- Perform EM for mixture models in general (when given update equations). Example: factor analysis.

7 Dimensionality reduction

7.1 Items to know

- Variance preservation view of PCA
- Reconstruction loss view of PCA
- Know conceptually how to find principal components: eigenvectors of the empirical covariance matrix
- Understand motivations (dimensionality reduction) and typical applications
- Understand the components of the SVD and their interpretations

7.2 Things to be able to work through when given information/equations

- How to perform PCA for toy examples, e.g. working with only 2 dimensions or when given a SVD.

8 Graphical models and Bayes nets

8.1 Items to know

- Understand the graphical representation of a Bayesian network (BN), as well as the role of conditional probability tables (CPTs)
- Idea of polytree requirement
- Plate notation
- d-separation rules

8.2 Things to be able to work through when given information/equations

- Can construct a BN for a given variable ordering (e.g., add A, then B and required edges to A, then C and required edges to A and B...), understand the effect of different orderings
- Know how to use variable elimination for inference, and understand the use of leaves-first ordering and the importance of a good elimination order
- How to deduce independence / dependence rules between variables via d-separation, given a specific Bayes Net

9 Hidden Markov Models

9.1 Items to know

- Know the form of the HMM, the distribution it defines, the conditional Independence properties, and understand the typical kinds of applications
- Inference: understand the inference questions of interest

9.2 Things to be able to work through when given information/equations

- Learning: the use of EM (no need to memorize the E or M step rules).
- The forward-backward algorithm (no need to memorize the α- and β definitions), the Viterbi algorithm for finding the max probability sequence of hidden states (no need to memorize the recurrence), and can understand how to work with α and β values for inference.
- Kalman filters

10 Markov Decision Processes

10.1 Items to know

- Know the form of the MDP model, understand the typical kinds of applications of MDPs
- Finite horizon planning: understand the planning objective, the MDP value function, policy evaluation, and the use of value iteration for planning. No need to memorize formulas.
- Infinite horizon: understand the planning objective, the MDP value function, policy evaluation, Bellman equations, value iteration (VI), policy iteration (PI) including policy evaluation, and how VI and PI compare. No need to memorize formulas.

10.2 Things to be able to work through when given information/equations

- How to apply policy iteration, policy evaluation, or value iteration to a specific toy problem, given the update formulas
- Computational complexity of different update steps

11 Reinforcement Learning

11.1 Items to know

- Understand typical applications, the difference between RL and planning, and the idea of exploration vs exploitation
- Understand the difference between model-based and model-free learning
- Understand the Q-function, the alternate form of Bellman equations using the Q-function

- Understand the structure the formulations of SARSA and Q-learning, as well as their differences.

11.2 Things to be able to work through when given information/equations

- How to apply the update rule for SARSA or Q-learning to a specific toy problem