**Required**: Kevin Murphy, *Machine Learning, a Probabilistic Perspective*, 1st edition, MIT Press (2012) [author disavows the eBook version sold on Amazon, only buy an eBook from MIT Press]

**Supplemental:** Christopher Bishop, *Pattern Recognition and Machine Learning,* Springer (2011 printing)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weekly Schedule** | | | | |
| **Week** | **Topics** | **Learning Goal** | **Evidence Required to Demonstrate Goals Are Achieved.**  **Be precise!!** | **Learning Activities That Will Produce the Evidence (quizzes, coding homework, etc.)**  **Be precise!! If it is a quiz, give some sample questions. If it is a coding assignment, give the specifications for the inputs, outputs, and algorithms to be implemented.** |
| 1 | Overview of Machine Learning: history, relation to classical statistics **Text:** Chapter 1; | 1. Explain the idea of intelligence especially as it relates to computers.  2. Explain what it means for a machine to “learn”.  3. Explain common learning paradigms  4. Be able to formulate a well-posed learning problem for a real-world task by identifying the task, performance measure, and training experience  5. Be able to identify Machine Learning in real-world scenarios | Interact and contribute to in-class group activities and discussions  Articulate answers for (in-class) questions:  What is intelligence and what is the difference of human intelligence versus computer intelligence?  What Does it Mean to Learn? | Video about how to use Machine Learning in Real World Applications  Play the intelligent paper game  Ask students to give other examples of supervised and unsupervised learning situations (in class discussion)  Ask students to formulate a learning problem (in class discussion)  Ask students to give other applications of Machine Learning (in class discussion) |
| 2 | **Linear Regression:** model specification issues, maximum likelihood estimation, ridge regression **Text:** Chapter 7, except 7.4 and 7.6 | 1.  To know simple linear regression, intercept slope coefficient parameter, least squares, sum of squares, population regression line, bias and unbiased, standard error, residual standard error  2. Design k-NN Regression and Decision Tree Regression  3. To Understand how to estimate the regression coefficients  4. Implement learning for Linear Regression using three optimization techniques: (1) closed form, (2) gradient descent, (3) stochastic gradient descent  5. To be able to choose a Linear Regression optimization technique that is appropriate for a particular dataset by analyzing the tradeoff of computational complexity vs. convergence speed | Explain difference between bias and variance? | Application cards. Record two or three different ways to apply regression to a real-world situation.   * Use a Naïve Bayes model to predict Sports vs Non Sports messages. * Non-graded quiz. * Expected Learning Outcomes Statements (end of class survey) |
| 3 | **Logistic Regression:** model fitting methods (steepest descent, Newton’s, L1 and L2 regularization, etc.) **Online learning algorithms**: structured prediction, regret minimization, stochastic optimization, risk minimization, LMS algorithm, perceptron algorithm, relation to Bayesian techniques. **Text:** Chapter 8.1-8.3.6, 8.5, 13.3, | Structured prediction Regret minimization  Stochastic optimization  Risk minimization  LMS algorithm  Implement the structured perceptron algorithm for sequence labeling.  Bayesian | How do you interpret the odds ratios of logistic regression?  Explain difference between train and test error  What is structured prediction? | Implement a spam detection logistic regression model  “Muddiest Point” Activity. conducted at the end of the week. Students are anonymously asked to report on a piece of paper what idea about regression was confusing or unclear. |
| 4 | **Support Vector Machines:** uses in regression and classification, optimization, choice of parameter C, probabilistic interpretation **Kernel Methods:** RBF kernels, Mercer kernels, Matern kernels, Linear kernels, String kernels, Kernels derived from probabilistic generative models; the “kernel trick” and its uses: nearest neighbor classification, K-medoids clustering, ridge regression, and PCA **Text:** Chapter 14.1, 14.2, 14.5, supplemental notes by Ng | Motivate the learning of a decision boundary with large margin  Compare the decision boundary learned by SVM with that of Perceptron  Derive the hard-margin SVM primal formulation  Derive the Lagrangian dual for a hard-margin SVM  Describe the mathematical properties of support vectors and provide an intuitive explanation of their role  Employ slack variables to obtain the soft-margin SVM  Employ the kernel trick in common learning algorithms  Use the "kernel trick" to obtain a computational complexity advantage over explicit feature transformation  Implement the K-Means algorithm  Connect the nonconvexity of the K-Means objective function with the (possibly) poor performance of random initialization  PCA and Dimensionality Reduction |  | Think-pair-share. Having students turn to the person next to them, discuss the question, and then vote again  **Think-pair-share. A**llow a couple of minutes for each individual student to thinkit through. Next, each student turns to the student next to him/her to discuss the question/answer as a pair. ask student pairs to sharetheir response with the class.  One Minute Paper.List your major questions related to Support Vector Machines. |
| 5 | **Decision Trees:** basis in information theory, when appropriate to use, growing and pruning, detecting and avoiding over-fitting; pros and cons with respect to other ML techniques; random forests; ID3 and related algorithms; use of bias; training with incomplete data **Boosting:** why it works so well, specific types of boosting, ensemble learning **Text:** Chapters 2.8, 16.2, 16.4, 16.6 , supplemental notes by Mitchell, Yoav & Schapire | 1. Implement Decision Tree training and prediction  2. Use effective splitting criteria for Decision Trees and be able to define entropy, conditional entropy, and mutual information / information gain  3. Describe the inductive bias of a decision tree  4. Judge whether a decision tree is "underfitting" or "overfitting"  5. Explain the difference between true error and training error  6. Implement a pruning or early stopping method to combat overfitting in Decision Tree learning  7. Identify when a model is overfitting  8. Add a regularizer to an existing objective in order to combat overfitting |  |  |
| 6 | **Deep Learning:** background in feed-forward neural networks (multilayer perceptrons) and Restricted Boltzmann machines; deep generative models (sigmoid networks, Boltzmann machines, belief networks) **Text:** Chapters 16.5, 27.7, 28.1-28.2 | Combine simpler models (e.g. linear regression, binary logistic regression, multinomial logistic regression) as components to build up feed-forward neural network architectures  Explain the reasons why a neural network can model nonlinear decision boundaries for classification  Identify (some of) the options available when designing the architecture of a neural network  Implement a feed-forward neural network |  | Video. Amazing applications of Deep Learning  build a simple neural network using tensorflow.  One Minute Paper:What you believe to be the most important/significant concept from Deep Learning. |
| 7 | **Deep Learning (continued):** training of deep networks, applications of deep networks **Midterm Review** **Text:** Chapters 28.3-28.4 | Construct a computation graph for a function as specified by an algorithm  Carry out the backpropagation on an arbitrary computation graph  Construct a computation graph for a neural network, identifying all the given and intermediate quantities that are relevant  Instantiate the backpropagation algorithm for a neural network  Instantiate an optimization method (e.g. Stochastic Gradient Descent (SGD)) and a regularizer (e.g. L2) when the parameters of a model are comprised of several matrices corresponding to different layers of a neural network  Apply the empirical risk minimization framework to learn a neural network  Use the finite difference method to evaluate the gradient of a function  Identify when the gradient of a function can be computed at all and when it can be computed efficiently |  | Student-Generated Exam Questions. |
| 8 | **Midterm Exam Clustering:** measures of dissimilarity; k-means clustering, clustering in mixture models **Text:** Chapters 25.1, 11.1-11.3 | Distinguish between coordinate descent and block coordinate descent  Define an objective function that gives rise to a "good" clustering  Apply block coordinate descent to an objective function preferring each point to be close to its nearest objective function to obtain the K-Means algorithm  Implement the K-Means algorithm  Connect the nonconvexity of the K-Means objective function with the (possibly) poor performance of random initialization |  |  |
| 9 | **Expectation Maximization:** basic idea, Jensen’s inequality, the EM algorithm, EM applied to various distributions (mixture of Gaussians), theoretical basis **Text:** Chapters 11.4, supplemental notes by Ng | Explain the relationship between parameters and hidden variables.  Construct generative stories for clustering and dimensionality reduction.  • Draw a graph explaining how EM works by constructing convex lower bounds.  • Implement EM for clustering with mixtures of Gaussians, and contrasting it with k-means.  • Evaluate the differences between EM and gradient descent for hidden variable models. | Aside from the fact that GMMs (Gaussian Mixture Models) use soft assignments and K-means uses hard assignments, there are other differences between the two approaches. What are they? | Homework. Textbook problems 11.1 to 11.15  Prove Jensen’s inequality using the definition of concavity and induction. |
| 10 | **Dimensionality Reduction:** classical Principal Components Analysis, PCA theorem; Singular Value Decomposition, Probabilistic PCA, EM algorithm for PCA, PCA for categorical data, PCA for paired and multi-view data **Text:** Chapters 12.2, 12.4, 12.5 | Define the sample mean, sample variance, and sample covariance of a vector-valued dataset  Identify examples of high dimensional data and common use cases for dimensionality reduction  Draw the principal components of a given toy dataset  Given a set of principal components, project from high to low dimensional space and do the reverse to produce a reconstruction  Explain the connection between PCA, eigenvectors, eigenvalues, and covariance matrix  Use common methods in linear algebra to obtain the principal components |  |  |
| 11 | **Graphical Techniques:** Bayesian nets, Directed Gaussian Models (DGM), plate notation, d-separation, Markov random fields, Hammersley-Clifford theorem and its effects, training maxent models, pseudo-likelihood, stochastic maximum likelihood learning **Text:** Chapter 10, 19.1-19.5 | Understand basic principles and techniques of probabilistic graphical models  Create suitable models for any given problem  Derive the algorithm (equations, data structures etc) needed to apply the model to data |  |  |
| 12 | **Graphical Techniques (continued):** belief propagation (serial, parallel, Gaussian, and other variants), variable elimination, junction trees, intractability of exact inference, approximate inference methods, max sum and max product **Text:** Chapter 20 | Implement the algorithm  Demonstrate your skills by doing a reasonably challenging project |  | Homework. Textbook problems 19.1 to 19.3, 20.1 to 20.4 |
| 13 | **Structured Prediction:** Markov and Hidden Markov (HMM) models, inference algorithms (forward, Viterbi, backward filtering, etc.), training methods, Baum-Welch algorithm, model selection, Conditional Random Fields (CRF), label-bias problem, uses in handwriting recognition, protein analysis, and stereopsis **Text:** Chapters 17.1-17.5, 19.6-19.7, CRF tutorial paper by Sutton & McCallum | 1. Show that structured prediction problems yield high-computation inference problems  2. Define the three key problems for an HMM: evaluation, decoding, and marginal computation  3. Compare the reinforcement learning paradigm to other learning paradigms  4. Cast a real-world problem as a Markov Decision Process  5. Derive a dynamic programming algorithm for computing the marginal probabilities of an HMM  6. Interpret the forward-backward algorithm as a message passing algorithm  7. Implement supervised learning for an HMM  8. Implement the forward-backward algorithm for an HMM  Implement the Viterbi algorithm for an HMM  9. Implement a minimum Bayes risk decoder with Hamming loss for an HMM | How Hard Is Inference for Structured Prediction? | One Minute Paper: What you believe to be the most important/significant concept from Deep Learning. |