COMP 551 Practice Questions

Yutong Yan

Fall 2018

1 Lecture 1

1. Question: In the class, we have seen that both classification and regression are supervised learning problems. What is the difference between these two supervised learning problems?
   Answer: That classification is the problem of predicting a discrete class label output for an example. $\hat{y} \in \{c_1, ..., c_n\}$
   That regression is the problem of predicting a continuous quantity output for an example. $\hat{y} \in R$
2. Question: Give five real life applications for classification and five real life applications for regression.
   Answer: Classification problems: Email Spam; Handwritten Digit Recognition; Image segmentation; Speech Recognition; DNA Sequence Classification Regression problems: Housing Price Prediction; Stock Price Prediction; ...
3. Question: When would you say that a particular model is a linear model?
   Answer: The model is linear in terms of the parameter.
4. Question: What is the significance of w0 (bias) in the linear model: y = w_0 + w_1x? Why not use a model like y = w_1x?
   Answer: If we do not add the bias term, we only cover the lines pass the origin.

2 Lecture 2

1. Question: What is overfitting? How can we find if a model is overfitting to a particular dataset?
   Answer: Overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably. (Like the m = 9 example in class)
   We can use a separate test set to recognize overfitting.
2. Question: Suggest at least 3 approaches to solve the overfitting problem.
   Answer:
   Solution 1: Add more data points so that the model cannot overfit.
   Solution 2: Regularization: Add a penalty term to the error function in order to discourage the coefficient from reaching large. (In the case we cannot obtain more data.)
   Solution 3: Model Selection: We can partition the data set into three parts, a training set, a validation set and a test set. The training set is used for training the model to fit the data, the validation set is used to measure differences in performance between models in order to select the best one and the test set to assert that the model selection process does not overfit to the first two sets.
3. Question: What is the hyperparameter of a model? How is it different from the parameters of the model? How can we choose these hyperparameters?
   Answer: a hyperparameter is a parameter whose value is set before the learning process begins. By contrast, the parameters of the model are derived via training. (The regularization term in error

function for linear regression model is the hyperparameter.)
Repeat training with different λs, and pick the one with the best perfomance in valid data set. (Model Selection)

4. Question: While doing model selection, we choose the best hyperparameter based on the validation set performance. What will happen if we choose the best hyperparameter based on training set performance? What will happen if we choose the best hyperparameter based on the test set performance? Do we really need a separate validation set?
   Answer: Training set performance: Overfitting. The parameters you learn during training are optimized to the training set. If you're not careful, you can over-optimize the parameters, leading to a model that's really good on the training set.
   Test set performance: The model has seen the test set so the test accuracy will not be proxy for the real performance of the model.
   If we use a set during your model fitting, the results we get on that set will not be fully indicative of the general results we'll obtain on completely new data. That's why we hold out a fraction of the data till the very end, past the point where we are making any decisions on what to do.

5. Question: What are the hyperparameters of the linear regression model? What are the hyperparameters of the k-NN classier?
   Answer: Linear regression model: M(degree of polynomial) and regularization term (if exists)
   K-NN classier: k; metric to compute NN (L1 vs. L2)

6. Question: Compare and contrast least squares approach and nearest neighbor approach in terms of bias and variance.
   Answer:
   Decision boundary: very smooth (least squares) vs. wiggly and depends on a handful of input points and their positions (nearest neighbors)
   Stability: stable vs. less stable
   Assumption: assumes the boundary is linear (strong) vs. does not have strong assumptions
   Least squares: high bias, low variance
   KNN: low bias, high variance

3 Lecture 3

1. Question: In the class, we have seen that if we use squared error loss, then the expected prediction error is minimized by the conditional mean. Explain how nearest neighbor approach and least squares approach are trying to approximate this conditional mean.
   Answer:
   Both k-nearest neighbors and least squares end up approximating conditional expectations by averages. But they differ dramatically in terms of model assumptions.
   Least squares assumes f(x) is well approximated by a globally linear function.
   k-nearest neighbors assumes f(x) is well approximated by a locally constant function.

2. Question: What is Bayes rate? We have seen in the class that Bayes rate is the best possible performance any classier can achieve. What does a classier require in order to achieve this optimal error rate?
   Answer:
   The error rate of the Bayes classifier.
   To achieve the optimal rate, we need to take a few data points first to learn the density distribution first, and then proceed.

4 Lecture 4

1. Question: What is the advantage of using non-linear basis functions with a linear model like linear regression?

Answer: We are trying to predict y from x, for some future test case, but we are not trying to model the distribution of x. Suppose also that we dont expect the best predictor for y to be a linear function of x, so ordinary linear regression on the original variables wont work well. So we need to allow for a non-linear function of x.

2. Question: What is the pseudo-inverse of a matrix? How is it different from the inverse of a matrix? When will the psuedo-inverse and inverse be equivalent?
   Answer:
   $A^+ = (A^T A)^{-1} A^T$
   Difference:
   Recall the definitions of both inverse and pseudo-inverse of a matrix:
   Let A $\in M_n(R)$, then matrix B $\in M_n(R)$ is called the inverse of A, if A $\cdot$ B = B $\cdot$ A = I_n, where I_n is identity matrix of size n.
   Let A $\in M_{n,m}(R)$, then matrix B $\in M_{m,n}(R)$, is called the pseudo-inverse of A, if the following 4 conditions are satisfied:
   A $\cdot$ B $\cdot$ A = A
   B $\cdot$ A $\cdot$ B = B
   $(AB)^T$ = AB
   $(BA)^T$ = BA
   If A is square and invertible, then $A^+ = A^{-1}$.

3. Question: Explain the geometrical interpretation of least squares approach.
   Answer:
   `https://ccrma.stanford.edu/~jos/sasp/Geometric_Interpretation_Least_Squares.html`

4. Question: When can one resort to gradient descent to minimize the objective function?
   Answer: The situation that it is not tractable to compute the analytical solution.

5. Question: What happens when the step size is too large in gradient descent? What happens when the step size is too small?
   Answer: Too small: the gradient descent can be slow.
   Too large: gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

6. Question: What is the difference between gradient descent and stochastic gradient descent?
   Answer: Stochastic gradient descent: Approximate the gradient of this total loss by the gradient of individual data point loss.

$$w = w - \alpha \frac{\mathrm{d}loss_k}{\mathrm{d}w}$$

   Here, k means k random data points.
   The stochastic step only tries to approximate the true gradient. The approximation error can help in escaping local minima.
   Gradient descent:

$$w_j = w_j - \alpha \frac{\mathrm{d}L(w_0, w_1)}{\mathrm{d}w_j}$$

   Here, $L(w_0, w_1) = \frac{1}{2N} \sum_{i=1}^{N} (w_0 + w_1 x^{(i)} - y^{(i)})^2$.
   In SGD, we are using the cost gradient of 1 example at each iteration, instead of using the sum of the cost gradient of all examples.

7. Question: Gradient descent can always find the global minimum. True or False? If false, is there any scenario when it is guaranteed to find the global minimum?
   Answer: False. For non-linear models, gradient descent might get stuck in any of the local minimum. It is guaranteed when the model is linear.

5 Lecture 5

1. Question: What is inductive bias? What is the inductive bias of linear regression and nearest neighbor algorithms?
   Answer: Inductive Bias is the set of assumptions a learner uses to predict results given inputs it has not yet encountered.
   Linear regression: The relationship between the attributes x and the output y is linear. The goal is to minimize the sum of squared errors.
   KNN: The classification of an instance x will be most similar to the classification of other instances that are nearby in Euclidean distance.

2. Question: A hypothesis which minimizes the empirical risk is also guaranteed to minimize the true risk. True or False?
   Answer:
   False.
   True risk: R(f) = E[(L, f(X))]
   Empirical risk: $R_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(Y_i, f(x_i))$
   Using Law of Large Number, empirical risk will converge to bayes risk, however, it is a terrible idea to optimize over all possible f: X $\rightarrow R$ functions!
   Empirical risk could be zero, and true risk could be larger than zero, which the model could extremely overfit the dataset.

3. Question: Define bias and variance. Explain the bias-variance trade-off.
   Answer: Bias is the difference between the expected or average prediction of our model and the correct value which we are trying to predict. (Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.)
   Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. (Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasnt seen before. As a result, such models perform very well on training data but has high error rates on test data.)
   Low variance + high bias = underfitting
   High variance + low bias = overfitting
   This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm cant be more complex and less complex at the same time. (If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then its going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.)

4. Question: Define Occam's razor.
   Answer: "One should not increase, beyond what is necessary, the number of entities required to explain anything."
   "Seek the simplest model."

5. Question: Adding regularization controls overfitting. True or False?
   Answer: True.
   Regularization helps to choose preferred model complexity, so that model is better at predicting. Regularization is nothing but adding a penalty term to the objective function and control the model complexity using that penalty term.

6. Question: Can we use L1 regularization and L2 regularization for feature selection? If so, explain how will you do that. Will there be any difference in the feature selection procedure based on whether the regularizer is L1 or L2 regularizer?
   Answer: A regression model that uses L1 regularization technique is called Lasso Regression and model which uses L2 is called Ridge Regression.
   L2 (Ridge regression): $w = (x^T x + \lambda I)^{-1} x^T y$
   -The solution adds a positive constant to the diagonal of $x^T x$ before inversion.

-This makes the problem non-singular even if $x^T x$ is not of full rank.

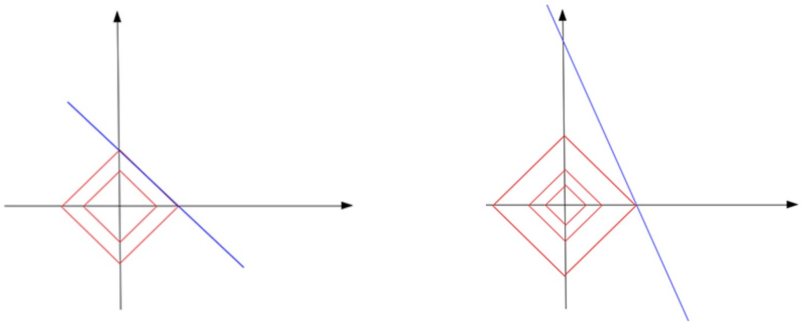
$$\min_w (y - xw)^T (y - xw)$$

subject to $w^T w \leq n$ where n is larger when $\frac{1}{\lambda}$ is larger
L1 (Lasso Regression): $E_w(w) = |w|$

$$\min_w (y - xw)^T (y - xw)$$

subject to $|w| \leq n$ where n is larger when $\frac{1}{\lambda}$ is larger
-There is no closed form solution.
-This function is not differentiated at w = 0.
-L1 does feature selection by setting weights of irrelevant features to zero.
-L1 prefers sparse models.

7. Question: L1 regularization prefers sparse models. Justify.
   Answer:
   Why sparsity can avoid over-fitting?
   By L1 regularization, we essentially make the vector x smaller (sparse), as most of its components are useless (zeros), and at the same time, the remaining non-zero components are very useful.
   As when your coordinate system has more axises, the L1 norm shape should have more spikes or tips.



   Left: Not only we cant have a unique solution this time, most of the regularized solutions are still not sparse (other than the two tip points.)
   Right: At the touch point, the constant c is the smallest L1 norm you could find within all possible solutions.

8. Question: Compare the geometrical views of L1 regularization and L2 regularization and argue why L1 regularizer sets more weights to zero than L2 regularizer.
   Answer: Comparison is above.
   For L2 regularization, we penalize weights with large magnitudes. However, the solutions are qualitatively different: with L1 regularization some of the parameters will often be exactly zero, which doesnt usually happen with L2 regularization. (For L1, we ignore some of the features, either by explicitly removing them, or by making any parameter weights connected to these features exactly zero.)

6 Lecture 6

1. Question: What are the three approaches to solving classification problem? Sort them in ascending order of procedure complexity.
   Answer:
   → First solve the inference problem of determining the class-conditional densities P($x|C_k$) for each class C_k individually.
   And use Bayes theorem.

Or we can learn the joint distribution $P(x, C_k)$ and then normalize to obtain $P(C_k|x)$.
Once we find $P(C_k|x)$, use decision boundary to determine the class
This is known as "generative model."
→ Secondly, directly model $P(C_k|x)$ and then use decision theory to determine the class.
Approaches that model $P(C_k|x)$ directly are called discriminative models.
→ Thirdly, find a function f(x) (called discriminant function) which maps x directly to the class label.
-There is no probability here.
-discriminant based models
⇒ Generative models; Discriminate models; Discriminant based models

2. Question: Why are generative models called as generative models?
   Answer:
   We can generate synthetic data points in the input space by using the learned distribution.

3. Question: What are linearly separable problems? Give cartoon examples for linearly separable 2-class problem and not linearly separable 2-class problem.
   Answer:
   (In the notes.)

4. In a linear discriminant model, decision surface is perpendicular to the weight vector. Prove.
   Answer:
   Consider two points x_A and x_B both of which lie on the decision surface.
   y(x_A) = 0 → $w^T x_A + w_0 = 0$
   y(x_B) = 0 → $w^T x_B + w_0 = 0$
   w^T (x_A - x_B) = 0
   ⇒ w vector is orthogonal to every vector lying within the decision surface.
   ⇒ w determines the orientation of the decision surface.

5. Question: Explain the difference between one-vs-rest classifier and one-vs-one classifier.
   Answer:
   One-vs-rest classier: k-1 classifiers each of which solves a two-class problem of separating points in class C_k from points not in that class.
   One-vs-one classifier: consider k(k-1)/2 binary discriminant functions, one for every possible pair of classes.

6. Question: Explain various ways of solving multi-class classification problem using discriminant functions.
   Answer:
   One-vs-rest classier; One-vs-one classifier; Single k-class discriminant

7 Lecture 7

1. Question: Least-squares solution lacks robustness to outliers when used for classification. Justify.
   Answer: The sum of squared error function penalizes predictions that are "too correct" in that way they lie a long way on the correct side of the decision boundary.

2. Question: List down the applications of PCA.
   Answer: dimensionality reduction; lossy data compression; feature extraction; data visualization

3. Question: Why should we constrain the norm of the projection vector in PCA to 1?
   Answer: Else you can maximize the variance by letting $||P|| \to \infty$.

4. Question: PCA and LDA project data from one space to another space. How can we use such algorithms for classification? Which projection will be more helpful to design a classier and why?
   Answer: PCA: Find the projection that maximizes the class separation.

LDA: Maximize a function that will give a large separation between the projected class mean while also give a small variance within each class, thereby minimizing the class gap.

Unsupervised vs. supervised learning:

LDA is very useful to find dimensions which aim at separating cluster, thus you will have to know clusters before. LDA is not necessarily a classifier, but can be used as one. Thus LDA can only be used in supervised learning.

PCA is a general approach for denoising and dimensionality reduction and does not require any further information such as class labels in supervised learning. Therefore it can be used in unsupervised learning.

5. Question: Which one of the following projection algorithms is supervised? PCA or LDA?
   Answer:
   (a) PCA is unsupervised (no label).
   (b) LDA is supervised (LDA makes use of the labels of data)
   (c) PCA projects data onto the direction(s) that yield the largest variance. Singular vector-1 gives the direction, which has the largest variance.
   (d) LDA projects the data to the directions that yield the largest variance R_{xy} and smallest variance R_{xx} and R_{yy} (say this is binary classification problem, class x and class y).

8 Lecture 8

1. Question: What is the difference between a linear model and a generalized linear model?
   Answer:
   Generalized linear model: decision surfaces are linear functions of x, even if activation function is non-linear. This type of model is called GLM.
   GLMs have complex analytical and computational properties than linear models.
   The general linear model requires that the response variable follows the normal distribution whilst the generalized linear model is an extension of the general linear model that allows the specification of models whose response variable follows different distributions.
2. Question: In GDA, covariance matrix of all the class conditional densities are shared. How is this affecting the decision boundary?
   Answer:
   Quadratic terms in x from the exponents of the Gaussian densities have canceled out due to the assumption of the common covariance matrices. The model is a linear function of x. Therefore, decision boundary is also linear function of x. Prior probability p(k) enters only through the bias parameter w_0 so that the changes in the prior have the effect of making parallel shifts of the decision boundary.
3. Question: Explain the i.i.d assumption.
   Answer:
   Assuming the examples are independent and identically distributed.
4. Question: What is the difference between GDA and QDA?
   Answer:
   GDA assumptions:
   1. Class conditional densities are Gaussian.
   2. All classes share the same covariance matrix.
   QDA: relaxes 2. and allows each class conditional density $P(X|C_k)$ to have its own covariance matrix, then we will obtain quadratic function of x.

5. Question: Define confusion matrix. How will an ideal matrix look like?
   Answer:
   A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.
   False Positives and False Negatives = 0.

6. Question: Define precision and recall. List two applications where precision is more important and two applications where recall is more important.
   Answer:
   precision = tp/(tp+fp)
   recall = tp/(tp+fn)
   Precision: one of all positive predictions, how many of them are truly positive?
   Recall: one of all positive examples in the data, how many of them are correctly identified as positive by the classifier?
   Precision is important in face password systems.
   Recall is important in email spam classifier.

7. Question: Explain the trade-off between precision and recall.
   Answer:
   The tradeoff is that it is easy to achieve high precision at the cost of low recall.
   1. The classifier which says default "yes." (100 percent recall, bad precision)
   2. The classifier which says default "no." (vice versa)

8. Question: Define F1-measure. What is the advantage of using F1-measure as an evaluation metric?
   Answer:
   F1 = 2pr/(p+r)
   Harmonic mean of precision and recall.

9 Lecture 9

1. Compare GDA and QDA in terms of parameter complexity.
   Answer:
   If there are M features, then
   GDA: kM parameters for the means, M(M+1)/2 parameters for the shared covariance matrix, k-1 parameters for P(C_k).
   QDA: kM parameters for the means, kM(M+1)/2 parameters for the shared covariance matrix, k-1 parameters for P(C_k).

2. Explain the naive Bayes assumption.
   Answer:
   Features are conditionally independent.

3. Gaussian Naive Bayes has linear decision boundary. True or False?
   Answer:
   False. It is linear only if the class conditional variance matrices are the same for both classes.

4. What is Laplace smoothing? Why do we need to smooth our Naive Bayes estimates?
   Answer:
   A correction is applied to conditional probabilities calculations to ensure that none of the probabilities is 0.
   To avoid P(C_k|x) = 0.

5. Laplace smoothing is a biased smoothing. Justify.
   Answer:

If no example from a class, it reduces to a prior probability of 1/2. Since M = (number of examples where x_1=1 and t=1) + 1/ ((number of examples where t=1) +2)

6. What are the advantages of discriminative approach over generative approach for classification?
   Answer:
   Generative:
   p($C_1|x$) = $\sigma(w^T x + w_0)$ (for 2-class problem)
   p($c|x$) = softmax($w^T x + w_0$) (for k-class problem)
   where w and w_0 are functions of mean, variance, and prior probabilities.
   Discriminative:
   Assume (w, w_0) as a vector of parameters and learn them directly by using maximum likelihood.
   Advantages:
   1. Fewer number of parameters.
   2. Improved predictive performance when the class-conditional density assumptions give a poor approximation to the true distributions.

7. Explain the relationship between maximum likelihood and least squares.
   Answer:
   For linear regression example, t = y(x;w)+ϵ, p(t|x,w,β) = $N(t|y(x;w), \beta^{-1})$, with dataset $\{x^n, t^n\}_{n=1}^N$

$$p(t|x, w, \beta) = \prod_{i=1}^{N} N(t^{(n)}|w^T x^{(n)}, \beta^{-1})$$

$$\ln p(t|x, w, \beta) = \sum_{i=1}^{N} \ln N(t^{(n)}|w^T x^{(n)}, \beta^{-1})$$

$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta \mathbb{E}_D(w)$$

   where $\mathbb{E}_D(w) = \frac{1}{2} \sum \{t^{(n)} - w^T x^{(n)}\}^2$ Thus maximizing log-likelihood w.r.t w is equivalent to minimizing $\mathbb{E}_D(w)$ w.r.t w.

$$\frac{\partial \ln p(t|x, w, \beta)}{\partial w} = \sum_{i=1}^{N} \{t^{(n)} - w^T x^{(n)}\} x^{(n)}$$

10 Lecture 10

1. Is there a closed form solution for logistic regression? If not, why?
   Answer:
   The error function is a non-linear function which is not quadratic (due to the sigmoid function). Error function for logistic regression is not quadratic but still a convex function. So there is a unique minimum.

2. What is the difference between gradient descent and Newton-Raphson method?
   Answer:
   Newton-Raphson method uses a local quadratic approximation to the error function. Specially, it uses a second-order Taylor series expansion to approximate E(w) near some point w_0 ignoring the derivatives of higher order.
   Note: when the error function is quadratic, Newton-Raphson method will find the solution in one step.

3. Derive Newton-Raphson update rule for least squares problem.
   Answer:
   H = $\nabla\nabla$E(w) = $\phi^T \phi$
   $w^{(new)} = w^{(old)} - (\phi^T \phi)^{-1}(\phi^T \phi w^{(old)} - \phi^T t)$
   $w^{(new)} = w^{(old)} - w^{(old)} + (\phi^T \phi)^{-1} \phi^T t$

$w^{(new)} = (\phi^T\phi)^{-1}\phi^T t$
which is the standard least squares solution

4. Explain the geometric view of gradient descent and Newton-Raphson method.
   Answer:
   In the notes.

5. Prove that in the absence of regularization, the maximum likelihood training for logistic regression can exhibit severe overfitting for datasets that are linearly separable.
   Answer:
   Maximum likelihood solution occurs when the hyperplane corresponding to $\sigma = 0.5$, equivalent to $w^T x = 0$. separates two classes and the magnitude of w goes to infinity.
   (This problem will happen even if we have more data points than the number of parameters in the model.)
   (How to solve this issue?: 1. Add regularization 2. Better estimates)

6. What are the advantages of generative models over discriminative models?
   Answer:
   We can compute p(x) as $\sum_k$ p(x|C_k)p(C_k). This can be useful for detecting new data points that have low probability under the model, for which prediction may be of low accuracy. (outliers detection)

7. What are the advantages of discriminative model over generative model?
   Answer:
   If we only wish to make classification decisions, we can just use discriminative models. class conditional densities may have lots of structure that has little effect on the posterior probabilities.
   Less demanding.
   No need for a large training set.

8. What are the advantages of discriminative models over discriminant based models?
   Answer:
   We have access to p($C_k|x$) using discriminative models.
   Advantages of p($C_k|x$):
   1. Minimizing risk
   2. Compensating for class priors
   3. Combining models
   4. Reject option

9. What is perception error criteria?
   Answer:
   If $x^{(n)}$ is in class C_1, $w^T x^{(n)} > 0$
   If $x^{(n)}$ is not in class C_1, $w^T x^{(n)} < 0$
   t $\in$ {-1, +1}
   For any $x^{(n)}$, $w^T x^{(n)} t^{(n)} > 0$
   For a data point is correctly classified, zero error.
   For a data point is wrongly classified, we want to minimize (-$w^T x^{(n)} t^{(n)}$).

$$E_p(w) = -\sum_{n \in M} w^T x^{(n)} t^{(n)}$$

10. What are the issues with the perception algorithm?
    Answer:
    1. 'Finite' number of steps can be very large. The smaller the gap, longer the time to find it.
    2. When the data is linearly separable, there are many solutions, and which one is found depends on the starting values.
    3. When the data is not separable, the algorithm will not converge, and cycles develop. The cycles can be long and therefore hard to detect.

In short,
If the data is linearly separable, the solution found by perception algorithm depends on
1. Initial values of w and b
2. Order in which data is represented

11 Lecture 11

1. Define margin of a decision boundary.
   Answer:
   Perpendicular distance between the decision boundary and the closest of the data points.

2. What are active constraints and inactive constraints in a constrained optimization problem?
   Answer:
   All data points will satisfy the constraint:
   $t^{(n)}(w^T\phi(x^n) + b) \geq 1$
   For data points for which equality holds, constraint is said to be active.
   For data points for which equality does not hold, constraint is said to be inactive.

3. Argue that there will be at least one active constraint in the max-margin problem and that there will be at least two active constraints when the margin is maximized.
   Answer:
   Because there will always be a closest point.
   When the equity holds, aka the margin is maximized, there will be two closest points.
   (Pictures in the notes can help understand this)

4. In SVMs, when will you solve the primal problem and when will you solve the dual problem?
   Answer:
   Primary: N>>M
   Dual: M>>N
   N is the number of data points.
   M is the number of parameters.

5. What is the error function that max-margin classier is trying to minimize?
   Answer:

$$\sum_{n=1}^{N} E_\infty(y(x^{(n)})t^{(n)} - 1) + \lambda\|w\|^2$$

   where $E_\infty = 0$ if z ≥ 0, $E_\infty = \infty$ otherwise.

6. What is the motivation for introducing slack variables in the max-margin optimization problem?
   Answer:
   Max-margin classifier assumes that the data is linearly separable. It gives infinite error if the data point was misclassified. (When data is not linearly separable, we can modify the approach so that the data points are allowed to be on the 'wrong side' of the margin boundary, but with a small penalty that increases with the distance from the boundary.)

12 Lecture 12

1. Compare the characteristics of squared error, cross-entropy loss, and hinge loss. How are they trying to approximate the misclassification error? Would you prefer one loss over the other? If so, why?
   Answer:
   Squared error: penalize points that are in the correct side if they are too far away from boundary.
   $\rightarrow$ But we need monotonically decreasing error function

Cross-entropy loss: continuous approximation of misclassification error.
Hinge loss: similar to cross-entropy, however, hinge loss is flat, which can lead to sparse solutions.
→ Squared error would not be a good one, because the error increases for large values of z. Hinge loss and logistic regression error are possibilities because they are monotonic.

2. What are the differences between parametric methods and non-parametric methods? Give examples for both methods.
   Answer:
   Parametric:
   Finite number of parameters
   Number of parameters is independent of dataset.
   (Logistic regression, GDA, Primal SVM)

   Non-parametric:
   Infinite number of parameters
   Number of parameters is dependent of dataset.
   (KNN, Dual SVM)

3. Why is K-NN called a lazy classifier
   Answer:
   -There is no learning. Just store every training example.
   -Most of the computation happens during prediction stage.

4. What is the advantage of using basis functions?
   Answer:
   One can take any linear model or GLM and use non-linear basis functions to get non-linear decision boundaries.
   Careful construction of basis functions can help us to project the data, which is not linearly separable in the original feature space, to a new projected space where it becomes linearly separable.

5. What are the various ways of splitting continuous valued attributes in a decision tree?
   Answer:
   1. Discretization: To form an ordinal categorical attribute.
   2. Binary Decision: Consider all possible splits and finds the best cut.

6. What are the different measures of node impurity?
   Answer: Gini index, entropy, and Misclassification error.

7. What is GINI index? How would you compute GINI index for a continuous valued attribute?
   Answer:
   Gini(t) = 1 - $\sum_j [p(j|t)]^2$, where $p(j|t)$ is the relative frequency of class j at node i.
   Use binary decision based on one value.
   Several choices for the splitting value. (Number of possible splitting value = Number of distinct values)
   Each splitting value has a count matrix associated with it. (Class counts in each of the partitions, A < V, and A ≥ V.)
   Simple method to choose v (more efficient):
   Sort the attributes on values.
   Linearly scan these values each time updating the count matrix and computing gini index.
   Choose the split position that has least gini index.

8. What is the disadvantage of using entropy as a criteria for attribute test selection in a decision tree? How can you overcome it?
   Answer:
   Tends to prefer splits that result in large number of partitions, each being small but pure.(Student ID will get high gain)
   Adjust info gain by the entropy of the partitioning (split INFO). Higher entropy partitioning (large number of small partitions) is penalized.

9. When can a decision tree overfit? What are the various ways to avoid overfitting in decision trees?
   Answer:
   -As tree grows deeper, model can overfit to the errors or noise.
   -Even if there is no noise, model can still overfit to coincidental regularities in the limited training data.

10. Explain reduced error pruning and rule post pruning. What are the advantages of rule post pruning over reduced error pruning?
    Answer:
    Reduced error pruning:
    -Consider each of the decision nodes in the tree to be candidate for pruning.
    -Running a decision node consists of remaining subtree rooted at that node, making it a leaf node, and assigning it the most classification of the training examples affiliated with that node.
    -Nodes are removed only if the resulting pruned tree performs no worse than the original over the validation set.
    Rule post-pruning:
    -Infer the tree from training set. Allow overfitting.
    -Convert the learned tree into an equivalent set of rules by creating one rule for each path from the root to a leaf node.
    -Sort the pruned rules by their estimated accuracy and consider them in this sequence while classifying subsequent instances.
    Advantages:
    1. Each node is used in different context. Reduced error pruning removes all the patterns through that node. Rule post-pruning prunes at fine graduality.
    2. Converting to rules removes the distinction between test that occur near the root of the tree and those that occur near the leaves.
    3. Converting to rules improves readability.

11. What are the advantages of a decision tree classier?
    Answer:
    1. Inexpensive to construct.
    2. Extremely fast at classifying unknown records.
    3. Easy to interpret for small sized trees.
    4. Can handle all types of attributes exactly.

13 Lecture 13

1. What is a mixture of expert?
   Answer:
   Mixture of experts refers to a machine learning technique where multiple experts (learners) are used to divide the problem space into homogeneous regions.

2. Explain the procedure for bootstrapping datasets. Why is it useful?
   Answer:
   We can create a dataset by drawing N data points at random from x_B, with replacement, so that some point x may be replicated in x_B, whereas other points in x from x_B. We can repeat this process by L times to generate L datasets each of size N and each obtained by sampling from original dataset x.

3. Prove that bagging reduces variance when we bag classifiers whose errors are uncorrelated.
   Answer:
   We use the same model but with different datasets. So we keep the bias of the model, but reduces variance. The error reduction happens because of variance reduction.

4. What is the difference between bagging decision trees and random forests? Which one would you prefer and why?
   Answer:

Random forest:
-Each tree has high variance, but the ensemble uses averaging, which reduces variance.
-Random forests are very competitive in both classification and regression, but still subject to overfitting.
Bagging decision trees:
-Often errors due to individual models are not uncorrelated. In practice, the errors are typically high correlated, and the reduction overall is generally small.

5. What are the differences between bagging and boosting?
   Answer:
   Bagging:
   -Faster
   -Small error reduction
   -Works well with "reasonable" classifiers
   -Has some issues with lots of mislabeled data, but relatively less.
   -Reduces variance
   Boosting:
   -Slower
   -Could be more
   -Works with very simple classifiers
   -May have if a lot of data mislabeled, because it will focus on those examples a lot, leading to overfitting
   -Reduces bias of the extremely weak learners which have high bias

6. What is the error function that AdaBoost is trying to minimize?
   Answer:

$$J_m = \sum_{n=1}^{N} w_n^{[m]} I(y_m(n^{(n)}) \neq t^{(n)})$$

14 Lecture 14

1. What is the optimal solution for exponential error function? How is AdaBoost trying to approximate it?
   Answer:

$$y(n) = \frac{1}{2} \ln \frac{p(t=1|x)}{p(t=-1|x)}$$

   The Adaboost algorithm is seeking the best approximation to the log-odds ratio, within the space of fns represented by the linear combination of base classifiers, subject to the constrained minimization resulting from the sequential optimization strategy.

2. What is k-fold cross validation? What are the advantages and disadvantages of doing k-fold cross validation?
   Answer:
   -Divide the data into k folds. (disjoint)
   -Use k-1 folds for training and last fold for testing.
   -Repeat previous step to test with all folds.
   -Average all 'k' performance.
   Advantages: One split may be not representative.
   Disadvantages: Very costly.

3. Explain stacking.
   Answer:
   Use the output of multiple classifiers as input to a meta-model.
   We 'stack' the meta-model on the top of base models.

Unlike bagging and boosting, base classifiers can be completely different.
Algo:
1. Divide the algorithm into k-folds.
2. For each fold, train all M classifiers using training part and make predictions for the validation part.
3. Create a new dataset:
i/p: predictions of M classifiers
0/p: target class
datapoints: For each fold, include model predictions in validation set as datapoints to new dataset
4. Train a meta classifier with this new dataset

4. What is representation learning?
Answer:
Learning the basis functions.
Basis functions project the data from the original space to a new space where it might be easy to learn to classify. In other words, they change the representation of data.

5. Can we add multiple linear layers in a neural network? If not, why?
Answer:
No. g has to be some non-linear activation function. If g is a linear function (like identity function), then
$$g = W^{(2)}(W^{(1)}n + b^{(1)}) + b^{(2)}$$
is equivalent to y = Wn + b for some W matrix and b vector. There is no use in projecting.

6. What type of output activation function will you use for the following scenarios:
Answer:
   - When the output is a regression target: identity function
   - When the output is a probability value: sigmoid function
   - When the output is a probability distribution: softmax function

7. State universal approximation theorem.
Answer:
"A single hidden layer neural network with a linear output unit can approximate any continuous function arbitrarily well, given enough hidden units."
This is a good result, but it does not mean there is a learning algorithm that can find the necessary parameter values.
The number of hidden units required grows exponentially as the complexity of the problem grows. (When we use gradient descent for learning).

8. What is the advantage of adding more hidden layers in a neural network?
Answer:
Increasing the number of neurons will allow you to decrease your training error but it also reduces the amount of generalization. When you add layers you increase the dimensional complexity of the data you can learn. Every time you add a layer, you change the shape of the discriminator.

15 Lecture 15

1. Backpropagation is an efficient way of implementing chain rule by using dynamic programming. True or False?
Answer:
True.

2. Explain the feed-forward stage and backpropagation stage in training a neural network.
   Answer:
   Feed-forward stage:
   Information comes from the left and each unit evaluates its primitive fn f in its right side as well as the derivative f' in its left side. Both results are stored in the unit, but only the result from the right side is transmitted to the units connected to the right.
   Backpropagation stage:
   Whole network is run backwards, whereby the stored results are now used. There are 3 cases to consider:
   Case 1: function composition
   Case 2: function addition
   Case 3: weighted edges

3. Explain how to compute gradient for the following cases using B-diagrams:
   In the lecture notes.
   - function composition
   - function addition
   - weighted edges

4. Backpropagation algorithm computes the derivative of the network function F w.r.t the input x correctly.
   Prove.
   Answer:
   We prove this by induction.
   The base case is: units in series, units in parallel, weighted edges. We proved this before with the three cases.
   The induction hypothesis is: The algorithm works for any feed-forward network with n or fewer nodes.
   Induction: if the algorithm works for an n-node network, then it will work for an (n + 1)-node network.

5. Derive backpropagation for a multi-layered feed-forward network with
   - sigmoidal hidden layers and softmax output layer.
   - tanh hidden layers and sigmoidal output layer.

   Answer:

16 Lecture 16

1. What is vanishing gradient problem? By using chain rule, explain why gradients shrink as the depth of the network increases.
   Answer:
   When we use sigmoid or tanh activation functions for hidden units, the range of values they can take are (0, 1) and (-1, 1) respectively. When we differentiate sigmoid or tanh activation function, the range of gradients w.r.t. their input are (0, 0.25) and (0, 1) respectively.
   This will shrink the gradient sent to $W^{(1)}$. Hence, as we increase the number of layers, gradients propagated to lower layers shrinks a lot!

2. What is an auto-encoder? How can we use auto-encoder for dimensionality reduction? How is it related to PCA?
   Answer:
   Autoencoder is a neural network with output being same as the input. The model is trained to

minimize the reconstruction error:
1. L2 loss in the case of real valued x

$$\frac{1}{2}(x - g(f(x))^2$$

2. Crossentropy loss if x is binary
h = f(x)
y = g(h)
Hence an autoencoder is an encoder followed by a decoder. However if the hidden layer size is greater than or equal to the size of the input layer, then the model would just learn to copy the input to output.
This model is more useful when the hidden layer size is less than the input layer size.
In this case, by minimizing the reconstruction error, model would learn to compress the data to a low-dimensional code and the decoder can use this code to reconstruct the original data. (Useful for dimensionality reduction)
When both hidden layer and output layers are linear layers and when trained with L2 loss, Autoencoder will learn to span the same subspace as PCA!
Hence Autoencoders can also be used for representation learning.
When we use non-linear hidden layers, the model learns a more powerful non-linear generalization of PCA.

3. Explain greedy layer-wise training procedure for training deep neural networks. How is it solving the vanishing gradient problem?
Answer:
Greedy layerwise pretraining: given a prediction task and a training set, we will use the following layerwise training procedure:
1. Given x, train an autoencoder.
2. Now throw away the decoder, and consider h1 as input and train another autoencoder.
3. Repeat this procedure 'k' times. You will end up with $W^(1)...W^(k)$ and $h_1...h_k$
4. Add a classification layer to the top of it and train the network for actual prediction task by using backprop.
In this stage, we will also update all previous weights.
Why should this work?: When we do supervised finetuning, gradients will still vanish. However, the lower layer weights are already initialized in reasonable region by the pretraining procedure. Hence you can consider this process as first learning better projections and then training a shallow classifier on the top.
This is better than plain backprop. But we can improve a lot by solving vanishing gradient problem!

4. Define ReLU activation function. What is the issue with ReLU activation? Explain how leaky ReLU solves that issue?
Answer:
ReLU is defined as follows:
f(x) = max(0, x)
Gradient of this function is 0 if $x < 0$ and 1 if $x > 0$.
This should greatly help in propagating gradients very deep.
The hidden layer activation are either zero or positive value. ReLU encourages the hidden layer activations to be sparse.
One issue with ReLU is when a unit receives negative value, it outputs zero and its gradient is also zero. Hence it might struck with zero output and especially it might die out (output zero). Often, many units might die out which is not desirable.
Leaky ReLU: f(x) = max(ϵx,x) where $\epsilon = 0.01$. Leaning will be small with negative inputs, but will exist nonetheless. In this sense, leaky ReLUs do not die.
With ReLU activation, we can train deep neural nets without pre-training.

5. Explain Batch Normalization procedure. How is it helpful in solving vanishing gradient problem? How will you use batch normalization during prediction time when you have only a single test instance?

Answer:
The main reason why gradients vanish is because the activation functions saturate (either to min value or max value). For example, when σ value is close to 0 or 1, then σ' value is close to 0. Sigmoid or tanh can easily saturate as weight value increases.
The idea of batch normalization is to normalize the pre-activation such that the hidden units activations will not saturate.

17 Lecture 17

1. Explain the advantage of CNNs over feed-forward neural networks for image data.
   Answer:
   -Neurons are arranged in 3 dimensions: width, height, and depth.
   -Each layer transforms an input 3D volume to an output 3D volume with some differentiable function that may or may not have parameters

2. What are the hyper-parameters of a convolutional layer? Explain the exact of each hyperparameter in terms of parameter complexity (how it affects the number of parameters required) and output complexity (how it affects the output volume).
   Answer:
   The conv layer's parameters consist of a set of learnable filters.
   Every filter is small spatially (along width and height) but extends through the full depth of the input volume.
   Example filter in first layer: $5 \times 5 \times 3$
   During the forward pass, we slide each filter across the width and height of the input volume and compute dot products between the entries of the filter over the width and height.
   As we slide the filter over the width and height of the input volume we will produce a 2D activation map that gives the responses of that filter at every spatial position.
   Intuitively, network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire wheel-like patterns on higher layers of the network.
   Now we will have an entire set of filters in each conv layer and each of them will produce a separate 2D activation map. We will stack these activation maps along the depth dimension and produce output volume.

3. What is zero-padding? Why should we do zero-padding?
   Answer:
   Sometimes it will be convenient to pad the input volume with zeros around the border. The size of this zero padding is a hyperparameter. Useful to control the spatial size of the output volumes.
   F-receptive field size
   S-stride
   P-mount of zero-padding
   W-input volume size
   Output volume size: $\frac{(W-F+2P)}{S} + 1$

4. What is pooling? What is the advantage of using a pooling layer? Can we substitute a pooling layer with a convolutional layer to get the same exact ? If so, how?
   Answer:
   It is common to insert pooling layer in between successive conv-layers in a convnet architecture.
   Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting.
   The pooling layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.
   Other types of pooling: average pooling and L2 norm pooling.
   Current trend is to just use bigger strides instead of pooling.

5. What is a sequential problem? Give at least 3 examples for sequential problem.
   Answer:
   Sequence classification; language modeling; conditional language modeling

6. Explain how can use use a feed-forward network to solve a sequential problem. Also explain the limitations of such an approach to solve the sequential problem.
   Answer:
   Solution 1:
   Feed forward network that takes x_t as i/p and predicts y_t
   -This model learns to predict y_t solely based on x_t.
   -Cannot capture the dependency of y_t with some previous x_t.
   -Stateless architecture.
   Solution 2:
   Consider previous k i/p while predicting the next output
   -Simulates state by explicitly feeding previous k inputs.
   -Can only model dependencies with previous k inputs.
   -Number of parameters grow linearly as the value of k increases.

18 Lecture 18

1. Explain how to unroll RNNs over time. What is the advantage of unrolling an RNN?
   Answer:
   In the notes.
   -Prediction for any y_t is dependent on all the previous inputs.
   -The number of parameters is independent of the length of the sequences.
   -RNNs can naturally handle variable length sequences.

2. Explain back-propagation through time (BPTT). What is the advantage of doing truncated BPTT over BPTT?
   Answer:
   Do backprop in the unrolled network.
   BPTT is costly if the sequence length is very long.
   Solution: to do truncated BPTT. Truncate the gradients for every 'k' steps.

3. Explain exploding gradient and vanishing gradient issues in training an RNN.
   Answer:
   Consider the recurrence relation:
   $h_t = W^T h_{t-1}$
   $h_t = (W^t)^T h_0$
   and if W admits an eigen-decomposition of the

   $$W = Q \wedge Q^T$$

   then

   $$h_t = Q^T \wedge^t Q h_0$$

   eigenvalues are raised to the power of t.
   Eigenvalues with magnitude less than one $\rightarrow$ decay to zero $\rightarrow$ gradients vanish
   Eigenvalues with magnitude greater than one $\rightarrow$ explode $\rightarrow$ gradients explode

4. Describe the LSTM architecture. Explain how it solves the vanishing gradient problem.
   Answer:
   LSTM has three states.
   Use gating(pass or block, or in other words, 1 or 0) function, not activation function. And train the combination of all those gates. Doing this, no matter how deep your network is, or how long

the input sequence is, the network can remember those values, as long as those gates are all 1 along the path.

5. What is the difference between the cell state and the hidden state in an LSTM?
   Answer:
   Hidden state: what info from past do I need to make my next prediction?
   Cell state: what info from past might I need to make future predictions?
   For regular RNNs, hidden state plays both of these roles.

6. Compare the advantages and disadvantage of

   - Batch gradient descent: computes the gradient of the cost function w.r.t. to the parameters θ for the entire training dataset

     $$\theta = \theta - \eta \nabla_\theta J(\theta; x^(1:T); y^(1:T))$$

     We need to calculate the gradients for the whole dataset to perform just one update.
     Can be very slow.
     Doesn't allow us to update the model online.
     Guaranteed to converge to the global minimum for convex error surfaces and to a local minimum for non-convex surfaces.

   - Stochastic gradient descent: SGD performs parameters update for each training example $x^(i)$ and label $y^(i)$
     Batch gradient descent performs redundent computations for large datasets, as it recomputes gradients for similar examples before each parameter update
     SGD does away with this redundancy by performing one update at a time
     Faster, can be used to learn online.
     Performs frequent updates with a high variance that causes the objective function to heavily fluctuate.
     These fluctuations help to skip local minima. Moreover, it complicates convergence to the exact minimum as SGD will keep overshooting.
     If we slowly decrease the learning rate, SGD shows the same convergence behavior as batch SGD.

   - Mini-batch gradient descent:
     Takes the best of both worlds
     Performs an update for every mini batch of n training examples
     Reduces the variance of the parameter updates, which leads to more stable convergence
     Can make use of highly optimized matrix operations.

7. Explain gradient descent with momentum. Why does momentum help in faster convergence?
   Answer:
   SGD has trouble navigating ravines i.e. areas where the surface curves much more steeply in one dimension than in another (which are common around local optima) In these scenarios, SGD oscillates across the slopes of the ravine while only making hasitnt progress along the bottom towards the local optimum.
   Momentum: helps accelerate SGD in the relevant direction and dampens oscillations.
   Momentum term increases for dimensions whose gradients point in the same directions and reduces updates for dimensions whose gradients change directions.

8. What is the difference between regular momentum method and Nesterov's accelerated gradient method?
   Answer:
   We can now effectively lookahead by calculating the gradient not w.r.t. to our current parameters but w.r.t. approx. future position of our parameters.

9. Explain Adagrad method. What are the issues in using Adagrad for training and how will you resolve them?
   Answer:

Adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters.
Well suited for sparse data.
Advantage: each parameter gets its own learning rate.
Disadvantage: If the initial gradients are large, the learning rates will be low for the remainders of training.
Due to the continual accumulation of squared gradients in the denominator, the learning rate will continue to decrease during the training, eventually decreasing to zero and stopping training.
To resolve the problem, instead of accumulating the sum of squared gradients over all time, use moving average

19 Lecture 19

1. What is the difference between frequentist approach and Bayesian approach for machine learning?
   Answer:
   Frequentist approach:
   -Treat δ as random
   -Empirical risk minimization and statistical decision theory that we have covered in earlier lectures.
   Bayesian approach:
   -Treat θ as random
   -Inference is made conditional on the current data
2. What is the difference between an estimation problem and a prediction problem?
   Answer:
   Estimation is always for unknown parameter whereas prediction is for random variable.
3. Explain how will you estimate the parameters when you use
   - Maximum Likelihood Estimation (MLE)
   - Maximum A Posteriori (MAP) Estimation
   - Bayesian Estimation
4. What are the advantages of using prior over parameters?
   Answer:
   Not only encodes this the maximum value of the data-generated parameters, but it also incorporates expectation as another parameter estimate, as well as variance information as a measure of estimation quality or confidence.
5. Under what circumstances would MAP reduce to MLE?
   Answer:
   If the sums of the counts and pseudo-counts become large, both expectation and maximum converge.
6. What is a conjugate prior? What are the advantages of using a conjugate prior.
   Answer:
   A conjugate prior of a likelihood is a distribution that results in a posterior distribution with the same functional form as the prior and a parameterization that incorporates the observations x.
   Advantages:
   1. determination of the normalizing term $\frac{1}{Z}$ is simple
   2. meaningful interpolation of hyper parameters.
   3. often allows to marginalize out the likelihood parameters in closed form and thus express the likelihood of observations directly in terms of hyper parameters.
7. Prove that Beta distribution is a conjugate prior for Bernoulli distribution.
   Answer:
   In the notes.

8. What are the advantages of using Bayesian estimation?
   Answer:
   - MLE can have large variance, like overfitting.
   - Bayesian approach gives an estimate of uncertainty.
   - Works well with small datasets
   - Use prior knowledge in a principled fashion
   - Warning: if a prior is wrong, posterior can be off