Lecture notes on Udacity's "Introduction to Machine Learning Class"

Lauren E. Blake

June 3, 2017

The class contains 16 lessons as well as a Final Project.

My code for the exercises and projects can be found on my GitHub Repo: https://github.com/Lauren-Blake/Udacity_Machine_Learning.

0.1 Lesson 1: Welcome

0.1.1 Introduction I

- Lots of applications for machine learning across diverse fields!

0.1.2 Introduction II

- Keep your eyes out for applications of machine learning and data sets that you could use machine learning on.

0.1.3 Introduction III

- Format: Lectures with quizzes, mini-projects at the end of each lesson.
- Final project at the end that ties together different aspects of the mini-projects.

0.2 Lesson 2: Naive Bayes

0.2.1 ML in the Google Self-Driving Car

- Train the self-driving car by showing car how to drive and giving the computer examples of humans driving. This is an example of a supervised classification problem.

0.2.2 Acerous versus non-Acerous Quiz

- Gave example of acerous animals and non-acerous examples. Then asked whether we thought a horse was acerous or not.
- Answer: A horse is acerous (lacking horns or antlers).

0.2.3 Supervised Classification Examples Quiz

- Need to have training data and then making predictions, recommendations, etc. based on the training set.

0.2.4 Features and Labels Musical Example Quiz

- With a song, features could be tempo, intensity, gender of person singing it, etc. Labels are whether a person likes the song or not.

0.2.5 Features Visualization Quiz

- Answer: She likes those because they are close to the other data points representing songs that the person also likes.

0.2.6 Classification by Eye Quiz

- Answer: Unclear because the new data point is close to two different labels.

0.2.7 Intro To Stanley Terrain Classification

- Features: speed and ruggedness

0.2.8 Speed Scatterplot: Grade and Bumpiness Quiz

- Answer: From the picture, we can see that the terrain looks flat and smooth, particularly relative to the other pictures.

0.2.9 Speed Scatterplot 2

- Answer: From the picture, we can see that the terrain looks very steep and medium bumpy.

0.2.10 Speed Scatterplot 3

- Answer: From the picture, we can see that the terrain looks flat and very.

0.2.11 From Scatterplots to Predictions

- Answer: The points look closer to the blue circles than the red X's.

0.2.12 From Scatterplots to Predictions 2

- Answer: Unclear because the points look equally close to the blue circles than the red X's.

0.2.13 Scatterplots to Decision Surface Quiz

- A decision surface parses out the training data into different features so that a data point falling on one side of the decision surface has one label and on the other side, a different label.

- Answer: Red cross because it is on the same side as the training data with red crosses.

0.2.14 A Good Linear Decision Surface

- When the decision surface is a straight line, it is a "linear" decision surface.

- Answer: Select the line that clearly and consistently separates the red crosses from the blue circles.

0.2.15 Transition into Using Naive Bayes

- Naive Bayes is a common algorithm to find the decision surface.

0.2.16 NB Decision Boundary in Python

- Have 750 training data points and make a decision boundary.

0.2.17 Getting Started with sklearn

- Documentation on Naive Bayes with derivation and code
- clf = GaussianNB() # Create Gaussian classifier
- clf.fit(features, labels) # Fit Gaussian classifier
- clf.predict #Give it a point and get out a label

0.2.18 Gaussian NB Example

- Goes through each line of code in the example section of http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.

0.2.19 GaussianNB Deployment on Terrain Data

- Answer: Add the following code:

```
#Specify classifier type
clf = GaussianNB()


# Fit the decision boundary
clf.fit(features_train, labels_train)
```

0.2.20 Finding Naive Bayes Accuracy

- accuracy = number of points classified correctly / all points in the test set
- Answer: See Lesson_2_Section_20_Quiz:Calculating_NB_Accuracy on my Udacity Machine Learning GitHub repo.

0.2.21 Training and Testing Data

- Important to train and test on different data sets (need to generalize to new data sets)

0.2.22 Unpacking NB Using Bayes Rule

- What is Naive Bayes?

0.2.23 Bayes Rule

0.2.24 Cancer Test

- Answer: We can see in from the diagram that the probability that someone with a positive cancer test actually has the disease is approximately 8%.

0.2.25 Prior and Posterior

- Answers: 0.009 and 0.099

0.2.26 Normalizing 1

- Answer: The normalizing constant is 0.009+0.099 = 0.108.

0.2.27 Normalizing 2

- Answer: 0.08333. Divide the cancer joint by the normalizing constant to get the posterior.

0.2.28 Normalizing 3

- Answer: 0.916666. Divide the non-cancer joint by the normalizing constant to get the posterior.

0.2.29 Total Probability Quiz

- Answer: 1. Add the answers from Normalizing 2 and Normalizing 3 together.

0.2.30 Bayes Rule Diagram

$$Total probability = 1 = P\left(\frac{C \mid Pos}{Normalizing constant}\right) + P\left(\frac{non - C \mid Pos}{Normalizing constant}\right) \tag{1}$$

0.2.31 Bayes Rule for Classification

- Who is the person that is sending the email based on the words that they used?

0.2.32 Chris or Sara Quiz

- Answer 1: Sara because the email contains words that she uses with higher probability than Chris.
- Answer 2: Chris because the email contains words that she uses with higher probability than Sara.

0.2.33 Posterior probabilities

- Answers: 0.5714; 1-0.5714; First find

  $$Prob\frac{\big(Chris \mid Email contains the words ``lifedeal''\big)}{Constant} \tag{2}$$

  Then, find its complement.

0.2.34 Bayesian probabilities on your own

- Prob (Chris | Email contains the words “love deal”) =

  Prob (“love deal” | C)*P(C) divided by a normalizing constant

  where the constant is Prob (“love deal” | C) P(C) + Prob (“love deal” | S) P(S)

  Then, take the complement.

- Answers: 0.5555; 0.4444.

0.2.35 Why Is Naive Bayes Naive

- Don't see underlying process (e.g. who is using the words) but get to see the outcome (e.g. the words that the person used).
- Answer: Word order is being ignored in Bayes Theorem whereas the words used and the length of the message (in terms of which words are used) are used.

0.2.36 Naive Bayes Strengths and Weaknesses

- Pros: Easy to implement, efficient
- Cons: Can break (e.g. phrases with distinct meanings)
- Good for text classification because can treat each word as a feature.

0.2.37 Congrats on Learning Naive Bayes

0.2.38 Lesson 2 Naive Bayes Mini-Project

- See code in "/Udacity_Machine_Learning/Lesson_2_Naive_Bayes_Mini_Project_Code.py"

0.3 SVM, Support Vector Machines

0.3.1 Welcome to SVM

- SVM is a very popular algorithm

0.3.2 Quiz: Separating a Line

- SVM takes in data from 2+ classes as input and draws a line to separate the classes
- Answer: The diagonal line that separates the Xs and the Os.

0.3.3 Quiz: Choosing Between Separating Lines

- Answer: The vertical line is the best separator.

0.3.4 Quiz: Choosing Between Separating Lines

- Want to choose a line that maximizes the distances to the nearest points in either class. Margin- maximizes distance to the nearest point.
- Answer: Something else because it maximizes the distances to the nearest points in either class (to be the most robust to classification errors).

0.3.5 Quiz: Practice with Margins

- Answer: The middle line maximizes the margin. It is the most robust to classification errors.

0.3.6 Quiz: SVMs and Tricky Data Distributions

- Answer: The line that fully separates the red from the blue points (diagonally downward).
- SVM prioritizes correct classification over maximizing the margin.

0.3.7 Quiz: SVM Response to Outliers

- What happens when no decision surface exists that completely separates the classes of data?
- Answer: Do the best it can

0.3.8 Quiz: SVM Outlier Practice

- SVM ignores extreme outliers.
- Answer: The line on the right is the best separator.

0.3.9 Handoff to Katie

- Making your own SVM

0.3.10 SVM in SKlearn

- Code: Import statement, training data, training features, create classifier, fit classifier, do a prediction.
- Important: import svm, classifier is svm.SVC()

0.3.11 SVM Decision Boundary

- Unlike with our Naive Bayes classifier, the SVM decision boundary will be a straight line.

0.3.12 SVM Coding Up the SVM

- Answer: See "Lesson_3_Section_12_Coding_Up_the_SVM for code. Also, the accuracy is 0.92, so it does better than the Naive Bayes classifier.

0.3.13 Nonlinear SVMs

- SVM can do complicated shapes in the decision boundary.

0.3.14 Quiz: Nonlinear Data

- Answer: No. Given our definition so far, SVMs will not separate this dataset.

0.3.15 Quiz: A New Feature

- Can put x squared and y squared into SVM in addition to x and y inputs.
- Answer: Yes, this is now separable.

0.3.16 Visualizing A New Feature

- Can do a linear transformation, which means we can look at it in a new coordinate system.

0.3.17 Quiz: Separating with the New Feature

- Answer: Yes, the data classes are now separable.

0.3.18 Quiz: Practice Making a New Feature

- Answer: Using the absolute value of X makes the data classes separable.

0.3.19 Kernel Trick

- Use kernels to change x,y input space to a much larger input (high dimensional) space. And this can lead to a non-linear separation for x, y

0.3.20 Quiz: Playing Around with Kernel Trick

- How to create an SVC and specify a kernel type
- Answer: All + more. There are many kernel types.

0.3.21 Quiz: Kernel and Gamma

- Parameters in machine learning- arguments passed when you create your classifier (prior to fitting)
- Parameters for an SVM- kernel (e.g. linear, rbf); C; gamma
- Answer: The plot on the far right best represents an SVM with a linear kernel and a gamma = 1.0.

0.3.22 Quiz: SVM C Parameter

- C- controls the tradeoff between a smooth decision boundary and classifying training points correctly.
- Answer: A smaller value of C will cause the optimizer to look for a larger margin (even if this means that the decision boundary will misclassify more points. Therefore, a large C means that you expect that you will get more training points correct.

0.3.23 Quiz: Gamma Parameter

- Gamma- defines how far the influence of a single training example reaches. Low values = far reach (even the points far from the potential decision boundary get taken into account when evaluating where to put the decision boundary) and high values = close.

0.3.24 Quiz: Overfitting

- Overfitting is a common problem in machine learning

- Answer: C, Gamma, and the Kernel type are all parameters that can impact fit/overfitting.

0.3.25 SVM Strengths and Weaknesses

- SVM works well in complicated domains where there is a clear margin of separation.

- SVM doesn't work well in large data sets or data sets with lots of noise (e.g. overlapping classes)

0.4 Lesson 4

0.4.1 SVM Strengths and Weaknesses

- Check