CS 374 Assignment #1

Introduction to Machine Learning
Linear Models, Perceptrons

Due the week of February 22, 2021

—

The readings and exercises for this first week of the tutorial were devised with two goals in mind. The first goal is to give you a general introduction to machine learning. The second goal is to begin to consider a simple yet powerful class of learning models: linear discriminant models. In particular, you will consider the perceptron learning algorithm, an online algorithm for learning a linear classification model from data. Finally, you'll see how perceptrons (and other methods) might be applied to the real world problem of identifying suspicious URLs.

Readings below are from the Alpaydin text, unless indicated otherwise. Complete references can be found by following the "Resources" link from the course homepage. If available, use the fourth edition of Alpaydin (the reddish orange book). If it isn't available, the third edition (green book) is fine for this assignment. You can find all non-Alpaydin readings on Glow. All readings except those from Alpaydin and Mitchell are available from the course website.

1 Introduction to Machine Learning

Your first objectives for the week should be to get a sense of the breadth of the field of machine learning and to become familiar with machine learning terminology.

1.1 Reading

Please read

- Chapter 1: Introduction
- Chapter 2: Supervised Learning

The first chapter in Alpaydin begins with a survey of some applications that can be addressed by machine learning. Because the types of applications we are concerned with are so diverse, no single algorithm can be expected to address them all. There are many factors that affect the design of machine learning algorithms including: the type of information available from which the algorithm can learn, the desired representation of the learned knowledge, expert theory about the domain of application, etc.

In reading Chapter 1, make note of the definitions of important terms such as *supervised learning* and *clustering*. These, as well as other terms highlighted in the chapter, will be part of our common vocabulary throughout the course.

The second chapter in Alpaydin introduces supervised learning in more detail. It discusses both classifier learning and regression, with a focus on the former.

A classifier considers examples, which it then designates as belonging to one class or another. There are many applications where classification is important. In medicine, for example, the process of diagnosis is one of classification. That is, the goal is to consider a patient (or, more specifically, a description of a patient, including history, test results, etc.) and to assign a "label" to the patient that gives the name of their disease. In banking, the process of loan approval is a classification problem. Here the goal is to consider an applicant and to assign a "label" indicating whether that applicant is low-risk (i.e., a good candidate for a loan) or high-risk (a bad candidate).

There are many applications for which the rules of classification are either unknown or are difficult to articulate. In these situations, we can use machine learning algorithms to automatically induce the knowledge that will allow us to perform classification.

Again, as you read this chapter, please note all new terms, such as *positive examples*, *negative examples*, *generalization*, *VC dimension*, *PAC learning*, *training set*, *test set*, *validation set*, *cross-validation*. Also note that Alpaydin tends to use the word "any" to mean "all". At times he also uses it to mean "at least one". The meanings should be clear from context, but if not, please ask me.

If you have any interest in reading other introductions to the topic of machine learning, you might consider Chapter 1 in Mitchell.

1.2 Exercises

In the tutorial meeting, I will expect you to introduce supervised learning. What is the goal? What information is available to a learning algorithm? What does it mean to search for a classifier in a space of hypotheses? What might such a space look like? Why does it matter? How does this relate to VC dimension?

I will also expect you to present your solution to the following problem:

1. Chapter 2, Exercise 9 in Alpaydin, fourth edition (Exercise 8 in Alpaydin, third edition).

2 Linear Models and the Perceptron Learning Algorithm

Our study of machine learning begins with algorithms for supervised learning of classifiers. A very simple type of classifier is the linear discriminant, and there are many different ways to learn one.

2.1 Reading

You were introduced to the notion of supervised learning of classifiers in Chapter 2. However, there the emphasis was on the use of rectangular regions to separate examples from different classes. Please read the following sections of Chapter 10 on Linear Discrimination:

- Section 10:1: Introduction (You can basically ignore up to the middle of page 244 (page 240 in the third edition). Start really paying attention with the paragraph that begins, "In the discriminant-based approach".
- Section 10.3: Geometry of the Linear Discriminant

It is important that you understand the basic geometry of the linear discriminant, and a number of the exercises below will guide you through this process. You might want to do the exercises that refer to this material before going on to the next set of readings.

Once you understand the geometry of the linear discriminant, you will explore one algorithm for learning a linear separator from data: the perceptron learning algorithm. While there are many algorithms for learning linear separators, the perceptron learning algorithm is interesting in that (1) it makes no assumptions about the underlying distribution of the data, and (2) it is an online algorithm (i.e., it does not require that you provide it with all the data at once; instead, it updates a linear model incrementally, with each new example that it sees).

Please read:

- Sections 11.1-11.4 of Chapter 11.

You might find it useful to read parts of Chapter 4 (through Section 4.4) in Mitchell. In particular, pages 90-91 in Mitchell give a nice visualization of the hypothesis space, as well a clear intuition for the gradient descent rule. You should certainly be sure you understand what the sigmoid (or logistic) function looks like and have some intuition for why it makes sense to use this as the output for classification, rather than the simple linear combination of weights and inputs.

Finally, to get a sense of the utility of linear discriminant models (even simple ones such as perceptrons) and more generally to get a sense of the process of applying machine learning to a practical problem, read

- "Identifying Suspicious URLs: An Application of Large-Scale Online Learning" by Ma, Saul, Savage, and Voelker.

You can find a link to the paper from the course assignment web page. You can also find it in Glow.

2.2 Exercises

On paper it is easy to draw a line separating positive-class and negative-class examples in a 2-dimensional feature space (assuming those examples are actually linearly separable). But what is the relationship between such a line and the weighted sum of input features: $\mathbf{w}^T\mathbf{x} + w_0$? Be prepared to explain this and to present your solutions to the following three problems.

1. In explaining the geometry of the linear discriminant, Alpaydin re-writes $\mathbf{x}$ as $\mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$. (See bottom of page 247 in the fourth edition or page 243 in the third edition.) Explain why $\mathbf{x}$ can be expressed in this way. You may assume that $\mathbf{x}$ is on the positive side of the hyperplane.

2. Starting with the alternate way to express $\mathbf{x}$, follow Alpaydin's explanation to derive r, the distance of $\mathbf{x}$ to the hyperplane, given by Equation 10.4. Be sure to fill in the details that Alpaydin omits.

3. Now show that the distance of the hyperplane to the origin is $\frac{w_0}{\|\mathbf{w}\|}$, as given by Equation 10.5. (This should be very easy.)

When learning a linear discriminant, our goal is to learn $\mathbf{w}$ and w_0 that give us the "best" classifier according to some criterion – for example, best with respect to some error function. One way to go about this is to perform stochastic gradient descent. Be prepared to explain stochastic gradient descent, to show how the update rule is derived, and to demonstrate the learning process:

1. Equation 11.8 tells us how to update the weights of a perceptron for each new training example. For this question, you will derive this update rule.

   As Mitchell (pages 90-91) tells us, it is helpful to visualize the entire hypothesis space of weight vectors and their associated error values. The error values make up an error surface. Because our goal is to minimize error, we want to find the point on the error surface that is minimal so that we can find the best (corresponding) weights.

   Gradient descent search determines a weight vector that minimizes error by starting with an arbitrary initial weight vector, then repeatedly modifying it in small steps. At each step, the weight vector is altered in the direction that produces the steepest descent along the error surface. This process continues until a minimum is reached.

   We calculate the direction of steepest descent along the error surface by computing the derivative of the error function, E, with respect to each component of $\mathbf{w}$. This is the *gradient* of the error function with respect to $\mathbf{w}$. When interpreted as a vector in weight space, the gradient specifies the direction that produces the steepest increase in E. The negative of this vector therefore gives the direction of steepest decrease.

   Use the information above to derive the update rule given in Equation 11.8, assuming that error is measured by cross-entropy. Cross-entropy is defined by Alpaydin just before Equation 11.8 on page 280. Be sure to use the version for two classes (about one third of the way down the page). If you are using the third edition of Alpaydin, you'll find the definition at the bottom of page 275.

2. Now use the update rule to learn appropriate weights for the following classifier-learning problem, which was inspired by a real application:

   You have been asked to learn a linear discriminant that will help distinguish normal cell phone usage from fraudulent usage. The data you have been given are described by two boolean features, x_1 and x_2, in addition to the class label. x_1 represents *Call time = midnight* and x_2 represents *Call origin = TownA*. The training data are as follows:

Call time = midnight	Call origin = TownA	Fraud
0	0	0
0	1	0
1	0	0
1	1	1

   If the initial weights are: $w_0 = -0.5$, $w_1 = 1$, $w_2 = 1$, and if $\eta = 0.1$, trace through the updates for each of the training examples. You should assume that the data are augmented by an input, $x_0 = 1$. **To make the calculations easier, use the threshold function (Equation 11.3 on page 276 of Alpaydin, fourth edition, or page 272 of Alpaydin, third edition), rather than the sigmoid, as the output of the perceptron.**

   The function to be modeled in this exercise is the boolean AND function. Why are the final weights different from those given on page 282 in Alpaydin, fourth edition (278 in Alpaydin, third edition)?

In your first exercise, you showed that the VC dimension of a line is just 3, but that doesn't stop us from using linear discriminants for classification in practice. Give a high-level overview of the Ma *et al.* paper. What is the goal? The approach? The experimental methodology? What are the authors' conclusions? What does this work contribute to the science of machine learning?

1. Write a very brief (no more than 1-page) bullet-point summary of the main points, and have it handy for your tutorial discussion.