#h1 Chapter 10 - Different Perspectives

#h2 More than meets the eye

#h3 10.1 - Introduction

#pg We have looked at our data from a geometric point of view. And we have linked those geometric concepts of algebra that underlie machine learning algorithms. In addition, we explored quantifying the statistical processes generating our data with specific probabilities. In addition, we have introduced more complex mappings that use non-linear functions. Each of these methodologies offers a distinct perspective on envisioning machine learning. I have found that the geometric approach works particularly well for me in guiding my understanding of complex algorithms like deep belief neural nets. Others are more comfortable with pure algebra, and a third group prefers the viewpoint of functions, derivatives, and integrals. But in principle, we can use each perspective to frame our machine learning problems.

#h3 10.2 - Statistical Perspective

#pg From a statistical perspective, machine learning seeks to find a generative statistical model and parameters on such a model to best match our observations. We previously formalized machine learning, defining the output or prediction y ̂ of an algorithm as a function applied to input data X, parameterized by the weights θ. This formalization in statistics uses posterior, conditional, and prior probabilities. For example, in a classification problem, the prior probability tells us the probability Pprior(c) of observing each class c. We would use that probability as our prediction, referred to as the posterior probability. In other words: Pposterior(c) = Pprior(c). The conditional probabilities constrain the prior probabilities by our input X. it especially informs us of the probability of each class being observed, given some input X, P(c|X). In machine learning, we look for patterns in the input data X that make P(c|X) different from Pprior(c). In other words, by observing some input data, the probability we assign to predicting c is increased or decreased. If somebody asks you whether it will be sunny today, you can estimate that by simply computing the probability of any day being sunny. That is, our estimate of Pposterior (sunny today) simply equals the estimate of our prior Pprior (sunny on any day). But if you had an extra bit of information (was it sunny yesterday?), you can adjust your estimate of whether today will be sunny or not. And you will be more confident that today will be a sunny day, given that yesterday was also sunny. Or, in statistical terms: P(sunny today | yesterday was sunny) > Pprior (sunny on any day). In this context, machine learning requires us to estimate these parameters on these underlying prior and conditional distributions to be best of our abilities. We have already seen in chapter 10 what to look for theoretically: inputs x that make the probability of output y given x; P(y|x) is not P(y,x).

#im ../assets/figures/010/010-01.png 75 32

#im ../assets/figures/010/010-02.png 75 32

#im ../assets/figures/010/010-03.png 75 16

#im ../assets/figures/010/010-04.png 75 32 Example 1 - We have two independent probabilities when we throw two dice. The probability of dice 1 rolling a six is not dependent on dice 2 turning up an even number. Therefore, we define the probability of seeing both as the product between the two probabilities of each event, which equals 1/12 or 0.8333.

#br

#br

#im ../assets/figures/010/010-05.png 75 32

#im ../assets/figures/010/010-06.png 75 32

#im ../assets/figures/010/010-07.png 75 16

#im ../assets/figures/010/010-08.png 75 32 Example 2 - This probability is different when the events are not independent. For example, we roll a single die, and it comes up even. What is then the probability we will roll a 6? Since a die has three even numbers (2, 4, and 6), the probability of having rolled a 6, given it came up even, is thus 1/3 or 0.333.

#br

#pg In other words, the probability of observing y depends on observing x. One more example hopefully solidifies this intuition. Let us say I have two dice. I throw them both, one at a time. Would you argue whether one rolling an even number (2,4,6)? Is it dependent on the other rolling a 6? Hopefully not because these probabilities are independent, and the probability (sometimes also called the likelihood) of observing both events is the product of observing them in isolation. In Example 1, our prior probability of rolling a six was 1/6. In other words, if we had no extra information, we would have to rely on the baseline probability of a dice rolling a 6. However, we can look for additional information on which the outcome of rolling a six is conditioned. This conditional probability refines our final posterior probability: the probability that combines both prior and conditional probability. The likelihood can increase or decrease, depending on the evidence. Still, the net result is that we are more confident than before when we had nothing more than the prior probability.

#h3 10.3 - Algebraic Perspective

#pg From an algebraic perspective, machine learning is about mapping inputs onto outputs as best we can by optimizing parameters specifying such transformations. We try to find the best set of equations that provide an optimal result, also defined by a function (i.e., loss). Personally, the best way of understanding these mappings is through geometry. Imagining or visualizing the state space of our data and its regularities that allow us to predict specific attributes from it guides us to find an appropriate approach and choice of algorithm.

#h3 10.4 - Calculus perspective

#pg In machine learning, we use algebra primarily to operate on variables and numbers. In contrast, calculus deals with functions and their derivatives. From that perspective, calculus helps us understand the mathematical function(s) we need to transform inputs into predictive outputs. It also helps us understand and improve the behavior of our optimization algorithms during learning. Most, if not all, optimization algorithms used in machine learning today use partial derivates of our inputs and parameters with respect to some measure of performance to estimate the change in parameters needed to maximize this performance. From this optimization viewpoint, machine learning is an exercise in calculus.

#h3 10.5 - Conclusion

#pg After reading the above, I hope you are beginning to see the parallels between these different perspectives. We are finding the most optimal mapping between inputs and outputs from an algebraic perspective. From a statistical perspective, we set out to compute posterior probabilities from both prior and conditional probabilities. But they are simply two different high-level perspectives to examine the same problem. Here is one more toy example to clarify this similarity. Imagine I have a bow and a lot of arrows. In my sufficiently sized lawn, I stand in the center of one of the outer edges and aim my arrow at the same spot on the edge off the other side of my lawn. This edge is too far for the arrow to reach, so it will land somewhere on the lawn. If my bows were perfectly balanced, the arrows would land on the imaginary line between me and the aiming point. However, bow number one has a slight imperfection, making the arrows veer off slightly to the left. And bow number two, you guessed it, has the same imperfection but to the right. The problem is the bows still look identical to the naked eye. How do I predict which bow I am currently shooting with, given the landing location of the arrow I just shot with it? From the algebraic perspective, a conventional classification algorithm will do the following. We can compute two transformations using training data for which we know which bow we are shooting arrows with. The first is to find the midpoint between the arrows shot with bow one versus bow two. This midpoint allows us to transform the data so that, on average, the arrows shot from bow one will end up to the left of the center. And we map anything left of the center onto a value of < 0. In contrast, we map arrows shot from bow 2 to values of > 0. We could stop here and return our prediction based on whether the landing location was > 0 or < 0. However, we would rather have some certainty or confidence in this prediction, preferably expressed as a probability. So, in the transformation, the algorithm maps these new values onto a sigmoid function. For points far to the left or right, the sigmoid transforms them into probabilities close to 0 and 1, respectively. Should these values be small, the probabilities get less pronounced, 0.3/0.6, or even 0.4/0.6, reflecting lower confidence in the prediction. In the extreme case, an arrow lands right in the middle. Here, the algorithm admits it has no confidence in its prediction, given that each outcome (bow one or bow 2) equals probabilities (0.5, 0.5). From the statistical perspective of machine learning, we do something remarkably similar, using terminology that appears so much different. Now, we do not try to find the midpoint directly but instead, estimate the two underlying distributions that gave rise to the range of landing positions we observed. When we shoot a new arrow, we note its landing position. We then ask what distribution was more likely to have generated that landing position. From this, we take the ratio between the probabilities of each distribution for the landing position. If the probability of bow 1 has 0.003 chance of coming from distribution 1, and a 0.020 of coming from distribution 2, we compute the discriminating probabilities as 0.003 / (0.003 + 0.02) = 0.13 and 0.02 / (0.003 + 0.02) = 0.87. And here is where it all comes together. Imagine the two distributions side by side. One distribution is centered on the arrows of bow 1, the other on the arrows of bow 2. We can compute the relative probability of each point falling between the means of these two distributions. When we do, we get an interesting and familiar result. In the end, all three perspectives still lend themselves to a geometric interpretation: the fact that we can think of our data as a set of points occupying an N-dimensional space. Statistics inform us of the distribution of these points and the statistical process (aka generative distributions) that gave rise to our observed data. Algebra tells us what transformations we can apply to map the points in our state space defined by our machine learning task in hand. Sometimes we aim to find a mapping that will separate the input points based on their class membership (classification and clustering). At other times, we try to map onto a continuous dimension (or set of dimensions) where their location closely matches some ground truth (regression). Finally, calculus informs us on how to find this transformation by iteratively improving the parameters of our models. In the following few chapters, we will more closely define and formalize machine learning through the perspective of algebra and statistics, with an occasional detour into the field of calculus.

#h3 10.6 - References

#bs

#be

#bp Strang, G. (2021). Introduction to Linear Algebra (5th ed.). Wellesley-Cambridge Press.

#bp Shilov, G. E. (2012). Linear Algebra. Dover Publications.

#bp Freedman, D., Pisani, R., & Purves, R. (2007). Statistics (4th ed.). WW Norton.

#bp Thompson, D. W. (1992). Canto: On growth and form (J. T. Bonner, Ed.). Cambridge University Press.

#bp Ellenberg, J. (2022). Shape: The hidden geometry of information, biology, strategy, democracy, and everything else. Penguin.

#bp Beckman, M. (2022). Math Without Numbers. Penguin Books.

#be