CSE 475: Statistical Methods in AI **Monsoon 2019**

SMAI-M-2019 4: Mathematical Foundations of ML - IV

Lecturer: C. V. Jawahar *Date: 8 Aug 2019*

4.20 Problem Space - IV

In the last lecture we looked at the learning problem and

- Understood it as an optimization problem of an appropriate loss/objective function.
- We also defined the data/examples $\mathcal{D}$ into two subsets $\mathcal{D}_{Tr}$ and $\mathcal{D}_{Te}$ as the subsets used for "Training" and "Testing"

Given this background, let us ask a critical question? What are we optimizing over?

- $\mathcal{D}_{Tr}$ or $\mathcal{D}_{Te}$ or $\mathcal{D}$?.

Since our objective is to define a computational procedure, we work only on $\mathcal{D}_{Tr}$.

This means our problem is something like:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i \in \mathcal{D}_{Tr}} ||f(\mathbf{x}_i, \mathbf{w}) \leftrightarrow y_i||$$

Where $\leftrightarrow$ compares the predictions (i.e., $f(\mathbf{x}_i, \mathbf{w})$) with that of "truth" (i.e., y_i). It could be something as simple as difference.

The above formulation seems to be correct (and that is what we are going to use also!!). But there is a very important issue/puzzle.

If we use a simple Look Up Table (LUT), that stores $\mathbf{x}_i, y_i$) and give you y_i for your ($\mathbf{x}_i$, does it minimize the above loss function? Yes; Indeed this gives zero error (perfect function) on the training set. However, a completely useless solution for the test set or for the purpose of learning. What went wrong in our problem formulation?

Really speaking, the problem we want to solve is slightly different, we would like to get a minimally complex function $f()$ that satisfy our data.

Occam's Razor: Prefer shorter hypothesis that fits the data.

- How do we define the "shortness"/complexity for the hypothesis/function $f()$.
- How do we find the shortest/shorter one?

The real problem of our interest is then

$$\min_{\mathbf{w},f} \sum_{i \in \mathcal{D}_{Tr}} ||f(\mathbf{x}_i, \mathbf{w}) \leftrightarrow y_i|| + ||f||$$

where $||f||$ is a measure of complexity of $f()$. Without going to the details, we make certain notes here.

- The complexity of the function class $f()$ can be computed in different ways. (Say degree of a polynomial defines how complex is the polynomial. Or number of weights in a Neural network defines how complex is the neural network function.). In more formal machine learning literature, "VC" dimension is used as a complexity measure.
- Unfortunately, searching over a function class $f \in \mathcal{F}$ is not simple computationally. Searching/Trying out all neural networks and then picking the one that optimizes the above problem does not seem to be very attractive.
- In practice, we fix $f()$, and then look for $\mathbf{w}^*$. With experience you will pick a suitable function $f()$ and move forward to find the optimal $\mathbf{w}^*$. This also answers your worry about why are we trying out many solutions/hyperparameters.
- Also note that the optimization problem that we solve could be non-convex. This means that the chance of getting a good solution depends on how well we optimize and also what function class we pick.

4.20.1 Overfitting

We know that we split the data into training and testing and build the model based on the training data. There is a common serious pitfall, that is expected. The performance on training data could be very good. While performance on the test data could be very bad. This is popularly known as overfitting and should be avoided.

Overfitting Worry against overfitting is a serious concern among practitioners of machine learning. An over complex function class is more likely to overfit your training data (like LUT). This explains why do we need simple models that fit the data.

When performance of the algorithm is superior on training data and inferior on test data, we say that the algorithm is overfitting the training data.

Generalization: Though ML problems look to be very similar to the classical modelling/fitting problems with data, we always aim at doing well on the "unseen" data. Our performance is defined as the performance on the unseen data and not on the training data.

Generalization typically refers to a machine learning solution's ability to perform well on new or unseen samples rather than the training data or data that it has used/seen while trainig. It is also related to the concept of overfitting. If the model is overfitted, then it will not generalize well. We work hard to avoid overfitting.

4.21 Probabilistic View Point

Let us revisit our problem of classifying an email as spam or non-spam. It may be important to make a final classification as 0 or 1. However, in many situations what we would like to obtain is the probability of the email being spam or non-spam. This may be useful for situations like:

- a human to look closely and take the decision.
- our classification is any way under uncertainty. capture this uncertainty.
- results of this stage is used for many tasks in the subsequent stages.

Probabilistic view of the classification also allows us to incorporate the prior knowledge we have, along with the evidences we have to make optimal decisions. We will see some such examples later today.

Also a number of tools and techniques from statistics and probability theory helps us in formulating and interpretting the formulations solutions.

4.22 Terms and Definitions

We assume that a student of this course has gone through a basic course on probability theory. There are a number of terms you should recollect at this stage.

- Random Variables and Probability
- Probability Density Function
- Types of Probabilities
- Marginal Probability
- Conditional Probability
- Joint Probability
- Popular Distribution
- Normal Distribution
- Beta Distribution
- Popular Results
- Sum Rule of Probability
- Product Rule of Probability
- IID
- And many more

Do read a brief note on these associated concepts in the annexure at the end of this.

4.23 Bayes Theorem

4.23.1 Example

Let us start with an example of Bayes decision in discrete case.

You are captured by the *Sentinelese* tribe while on your excursion to the islands. You are brought to the chieftain for prosecution. You are blindfolded and the chief selects a fruit from a basket containing 85 green mangos, 5 yellow mangoes, 2 green pears and 8 yellow pears. If you guess the fruit correctly, you are set free. If not ..

- What is your guess?
- What is your chance of survival?

Simple Solution

$$P(\text{Mango}) = \frac{90}{100} = 0.9$$

$$P(\text{Pear}) = \frac{10}{100} = 0.1$$

So the safe bet is Mango. Isn't?

Evidence: Decisions are usualy not that simple. You will have more evidence to analyse the situation.

1. You get a glimpse through the blindfold and you see a slight yellow color in the chiefs hand.
2. Unfortunately you are colour-blind and you mistake green for yellow 20% of the time, but never yellow for green.

- What is your best guess?
- What will be your chance of survival now?

4.23.2 Bayes Theorem

Conditional probability : Conditional probability is the probability of observing an event, given the fact that a second event has occurred. Using formal notations, we write:
$P(fruit = mango/you\ saw\ yellow)$: read as $P(fruit = mango)$ given *you saw yellow*; or in short as: $P(mango/yellow)$.

Priori and posterior probabilities:

- Class prior probabilities $P(\omega_i)$
- In our example, this would be $P(mango)$ and $P(pear)$.
- The class-conditional probability density function $p(\mathbf{x}/\omega_i)$. The probability density function for x given the state of nature is ω_i
- In the example above the class conditional probabilities are $p(yellow/mango)$, $p(green/mango)$ etc.

Bayes Rule: Bayes rule states that the joint probability of $\mathbf{x}$ and ω_i, denoted as $p(\mathbf{x}, \omega_i)$ is given by:

$$p(\mathbf{x}, \omega_i) = p(\mathbf{x}/\omega_i).P(\omega_i) = P(\omega_i/\mathbf{x}).P(\mathbf{x})$$

We can rewrite the second equality as:

$$P(\omega_i/\mathbf{x}) = \frac{p(\mathbf{x}/\omega_i).P(\omega_i)}{P(\mathbf{x})}$$

Here the L.H.S is the posterior probability of class ω_i after observing $\mathbf{x}$. Bayes decision rule says to choose that ω_i which maximises the posterior probability. The above equation may also be written as:

$$P(\omega_i/\mathbf{x}) = \frac{p(\mathbf{x}/\omega_i).P(\omega_i)}{\Sigma_{j=1}^{c} p(\mathbf{x}/\omega_j).P(\omega_j)}$$

Bayes rules gives you a mathematical formula for combining the evidence (what you saw) with your prior knowledge (what you knew about number of fruits and their colours). The combined probability is usually called *posterior probability.*

Using Bayes rule we write:

$$P(mango/yellow) = \frac{p(yellow/mango).P(mango)}{P(yellow)}$$

Or

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Bayes formula helps to convert the prior probability $P(\omega_i)$ to the *a posteriori* probability $P(\omega_i|x)$

Given the priors are equal, the category ω_j for which $p(x|\omega_j)$ is large is more *likely*.

If you are not colour blind:

$$P(\text{Mango/Yellow}) = \frac{\frac{5}{90} \cdot \frac{90}{100}}{\frac{5}{90} \cdot \frac{90}{100} + \frac{8}{10} \cdot \frac{10}{100}} = 0.385$$

$$P(\text{Pear/Yellow}) = \frac{\frac{8}{10} \cdot \frac{10}{100}}{\frac{5}{90} \cdot \frac{90}{100} + \frac{8}{10} \cdot \frac{10}{100}} = 0.615$$

Evidence can chnage your apriori decision!!

If you are colour blind:

$$P(\text{Mango/Yellow}) = \frac{\frac{5+0.2*85}{90} \cdot \frac{90}{100}}{\frac{5+0.2*85}{90} \cdot \frac{90}{100} + \frac{8+0.2*2}{10} \cdot \frac{10}{100}} = 0.724$$

$$P(\text{Pear/Yellow}) = \frac{\frac{8+0.2*2}{10} \cdot \frac{10}{100}}{\frac{5+0.2*85}{90} \cdot \frac{90}{100} + \frac{8+0.2*2}{10} \cdot \frac{10}{100}} = 0.276$$

4.24 Normal Distribution

We are familiar with the Normal/Gaussian distribution with mean μ and variance σ^2 from the school

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

In this case, x is the single variable. As we had seen in the problems of interest, our $\mathbf{x}$ is a vector consisting of $x_1, \ldots, x_d$. This naturally, demand the multivariate case as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}[\mathbf{x}-\mu]^T\Sigma^{-1}[\mathbf{x}-\mu]\right)$$

Indeed when $d = 1$, both these equations become the same. Naturally, our mean will be a d dimensional vector. And the covariance Σ is a $d \times d$ matrix.

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

$$\Sigma = \frac{1}{N}[\mathbf{x}-\mu][\mathbf{x}-\mu]^T$$

- Q: What do the elements of Σ imply?
- Q: What are the properties of Σ?
- Q: How is this covariance matrix related to the correlation matrix ?
- Q: By looking at the covariance matrix, what all we can say?

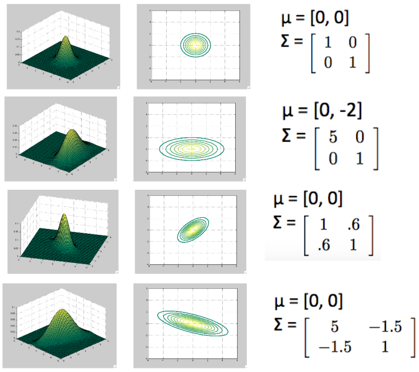

Figure 4.2: Appreciating the Covariance Matrix Structure

- Q: Why certain types of covariance matrices like $\Sigma = \sigma^2 I$ are of importance?
- We often model classes as multivariate Gaussians. Or we assume that there is an expected behaviour (measurement) for a class such as mean and there is a small deviation from the expected behavior that is modelled as Normal distribution.
- The quantity $[\mathbf{x}-\mu]^T\Sigma^{-1}[\mathbf{x}-\mu]$ is of special interest to us. This is called Mahalanobis distance.