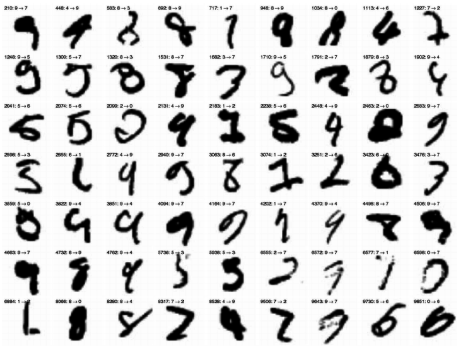
AIML Lecture 2 : Reading Material

1 Probability Primer

In most of the machine learning problems, ambiguity and uncertainty exists and cannot be fully avoided. For example, in the figure its very difficult to say with certainty which digit is written. However we still need to design algorithms which gives some reasonable outputs even in such corner cases. Probability theory, as we know, is a way of modeling uncertainty. Following are some of the basic terms in probability:



- **Sample Space:** The sample space, denoted as Ω is the set of all possible outcomes. In the case of the MNIST prediction, the sample space is $\Omega = \{0, \cdots, 9\}$.
- **Random Variable:** A random variable denotes a variable which can take values in the sample space. In the case of the MNIST prediction, the output of a machine learning algorithm on a specific input image can be represented as a random variable.
- **Probability Density:** The probability density (or just probability) is a function $p : \Omega \to \mathbb{R}^+$, which assigns a positive number less than 1, to every element in the sample space. The sum of probabilities should add upto 1.

$$\sum_{x \in \Omega} p(x) = 1.$$

- **Mean or Expectation:** The mean or expectation of a random variable X denoted as $\mathbb{E}X$ or μ is an equivalent of the average value, given by the formula

$$\mathbb{E}X = \sum_{x \in \Omega} p(x) \cdot x.$$

- **Variance:** The variance of a random variable X denoted as $\mathrm{Var}(X)$ represents how much the variable deviates from the mean value. It is given by the formula

$$\mathrm{Var}(X) = \sum_{x \in \Omega} p(x) \cdot (x - \mu)^2.$$

We often need to represent the input to a machine learning algorithm also as a random variable X. In this case, the input X will be considered to be *uniformly chosen* from the

dataset. That is if the data set is $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, the sample space of X will be the set $\{x_1, x_2, \cdots, x_n\}$ and the probability of each element x_i, $p(x_i) = 1/n$. The ground truth label for the input random variable is Y. As we saw for the MNIST classification task, the sample space of Y is $\{0, \cdots, 9\}$. Given X as input, a machine learning algorithm will predict a label $\hat{Y}$ and the accuracy of the algorithm is essentially the probability that $Y = \hat{Y}$. When we have multiple random variables we need to talk about the probabilities jointly.

Joint Probability Density: Given two random variables X, Y with samples spaces Ω, Ω', we can define a new random variable $Z = (X, Y)$ for which the samples space is $\Omega \times \Omega'$. That is the set of outcomes of Z is simply obtained by taking all pairs of outcomes (denoted as $\Omega \times \Omega'$) of X and Y. The joint probability density of X, Y is simply the probability density function of Z. Another way to think about joint probability density is as a matrix, where the rows are indexed by outcomes of X and the columns are indexed by outcomes of Y and for $x \in \Omega, y \in \Omega'$ the (x, y)th entry in the matrix is just the probability of $X = x$ and $Y = y$ denoted by $p(x, y)$. Joint probability densities must satisfy the following sum rule:

$$p(x) = \sum_{y \in \Omega'} p(x, y) \text{ and } p(y) = \sum_{x \in \Omega} p(x, y)$$

Conditional Probability: Given a specific input to a machine learning algorithm say $X = x$, the probabilities of the output $\hat{Y}$ and the ground truths Y will change. For example, suppose x is a handwritten image of 9, the ground truth label Y is equal to 9 with probability 1. We called this changed probability density due to the fixing of X to x, as the conditional probability density denoted by $p(Y = y|X = x)$ or simply $p(y|x)$. Note that in our example $p(9|x) = 1$ and $p(8|x) = p(7|x) = \cdots = p(0|x) = 0$. Such probability densities are called *one hot* probabilities. Conditional probabilities can be calculated by the following formula:

$$p(x|y) = \frac{p(x, y)}{p(y)}.$$

Bayes Rule: Bayes rule is an important relation in probability theory. It gives a way of "reversing" the conditional probabilities. It is given as follows:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x') \cdot p(x')}.$$

Independence: For two random variables X, Y, we say they are independent if the conditional probability is the same as the original probability density. That is:

$$p(x|y) = p(x) \text{ and } p(y|x) = p(y).$$

In this case, the joint probability densities reduces to the product of the probabilities. That is:

$$p(x, y) = p(x) \cdot p(y)$$

2 Dimensionality Reduction

In most real world problems, the raw input based on which predictions needs to be done, will be of a huge size. For example a typical color image of 1024×1024 resolution consists of $1024 \times 1024 \times 3$, 32 bit floating point numbers. For having fast algorithms, it is essential that the raw data must be processed such that the dimension is reduced to a reasonable size.

One way of doing dimensionality reduction is by selecting a few coordinates of the original input data that are the most relevant for the prediction. A more powerful way of doing dimensionality reduction is by multiplying the input (column) vector by a matrix which has very few rows. This results in an output which is low dimensional. We will be covering more advanced dimensionality reduction methods in the later lectures.

Another use for dimensionality reduction is visualization. We cannot visualize high dimensional vectors as we can perceive only 2D or at max 3D vectors. So if we reduce the dimension to 2, we can easily plot the input data in 2D and visualize them.

3 Performance Metrics

In a supervised machine learning problem, the labeled dataset is split into 2 parts known as training and testing data. The algorithm gathers insights from the training data and testing data is only used for evaluating the performance or accuracy. This is needed, since we can always have an algorithm which can be correct on the dataset, but completely wrong on unseen data. In this section, we review 2 types of supervised learning problems and the accuracy measures used to evaluate their performance.

3.1 Classification

Classification is a type of problem in which, we need to predict a class for a given input. The classes are disjoint, that is any input belongs to only one class. If the number of classes is 2, then it is called a *binary classification* problem. The accuracy in a classification problem is given by the fraction of the testing data for which the predicted output is the same as the ground truth label. That is, if the testing dataset is $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, and the predicted outputs for each of the inputs is given by $\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n$, then the classification accuracy is given by

$$\frac{\text{the number of i's for which } y_i = \hat{y}_i}{n}.$$

For the case of binary classification, we can further decompose the the above formula using the following terms:

- True Positive (TP) is the number of examples for which $y_i = \hat{y}_i = 1$.
- False Positive (FP) is the number of examples for which $y_i = 0$ and $\hat{y}_i = 1$.

- True Negative (TN) is the number of examples for which $y_i = \hat{y}_i = 0$.

- False Negative (FN) is the number of examples for which $y_i = 1$ and $\hat{y}_i = 0$.

In terms of the above, we can define the accuracy as:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Confusion Matrix: Given the dataset of (x_i, y_i) and predictions $\hat{y}_i$, we can obtain a *confusion matrix* which gives more information about the algorithms performance. If the number of classes is c, the confusion matrix is a $c \times c$ dimensional square matrix M. The (j, k)th entry of the matrix is given by

$$M_{j,k} = \frac{\text{number of i's for which } y_i = j \text{ and } \hat{y}_i = k}{n}.$$

That is, it the the number of testing examples for which the ground truth is j, but the prediction was k. For an ideal algorithm, the confusion matrix will be the identity matrix (1 along the diagonal). Note that every row as well as every column of the confusion matrix sums up to 1.

3.2 Retrieval

Retrieval problem models situations similar to web search. In this case, the input is like a query and we need to find an ordered list of relevant examples from the dataset. In the dataset, we will be given input x_i and a list of all relevant search results (or related items) $y_{i1}, y_{i2}, \cdots, y_{ik}$.

The performance evaluation of retrieval systems uses the following terms

- Precision is the fraction of relevant documents retrieved from the total number retrieved. Also called the true positive rate.

- Recall the the fraction of relevant items retrieved from the set of total relevant items.

4 References

1. Khan Academy Videos/Quizzes for Probability
   https://www.khanacademy.org/math/probability/probability-geometry
2. Books : Short Introduction to Probability
   https://people.smp.uq.edu.au/DirkKroese/asitp.pdf
3. University of Toronto lecture slides for Probability
   http://www.cs.toronto.edu/~urtasun/courses/CSC2515/Tutorial-ReviewProbability.pdf
4. Dimensionality Reduction (slides)
   https://www.cc.gatech.edu/~simpkins/teaching/gatech/cs4641/slides/dimensionality-redu
   pdf
   http://courses.washington.edu/css581/lecture_slides/17_dimensionality_reduction.pdf
5. Evaluating Machine Learning Methods (slides)
   http://pages.cs.wisc.edu/~dpage/cs760/evaluating.pdf
6. Evaluation Measures for Retrieval
   http://people.cs.georgetown.edu/~nazli/classes/ir-Slides/Evaluation-13.pdf
   https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)