Weekly report of lessons

Name: Mayank Kumar
Roll No: 19CS30029
The week: 20-09-2021 to 26-09-2021

The topics covered:

Bayesian Classification: Parametric and Nonparametric methods, Maximum Likelihood Estimation, Bernoulli distribution, Multi-density Function, Gaussian Density Function, Bias and MSE of estimators, Sample Mean and Variance, MSE of estimator, Bayes Estimator, Parametric Classification, Multivariate normal distribution, Multivariate discrete features, Application, Nonparametric Approach, Univariate nonparametric density estimation, Kernel Estimator, k-NN Estimator, Instance based learning, k-NN Regression, Locally weighted regression; Unsupervised Learning: Clustering, Class, Cluster, Motivation of clustering, K-Means Clustering, Exhaustive K-Means, Lloyd Algorithm, Conservative Approach, Weakness of algorithm, K-Means++, Determining optimal K

Summary topic wise:

Parametric Methods: Estimation of parameters used for computing class likelihood and posterior
Maximum Likelihood Estimation (MLE) of parameters:
Data X = $\{x^t\}$ for t = 1 to N. Here, x^t is an independent and identically distributed sample.
Likelihood: $l(\theta|X) = P(X|\theta) = \prod_{t=1}^{N} P(x^t|\theta)$, MLE of θ: $\theta^* = argmax_\theta\, l(\theta|X)$, Log-likelihood: $L(\theta|X) = \sum_{t=1}^{N} log\, P(x^t|\theta)$

Bernoulli distribution: Outcome 0 or 1. 1 has probability p and 0 has 1-p.
P(x) = px(1-p)(1-x), E(X) = p, var(X) = p(1-p)

$L(p|X) = log \prod_{t=1}^{N} p^{x^t}(1-p)^{1-x^t}$ and $\frac{\partial L(p|X)}{\partial p} = 0 \Rightarrow MLE\,(\widehat{p}) = \frac{\sum_{t=1}^{N} x^t}{N}$

Multi-density Function: Generalized Bernoulli Process. One of k mutually exclusive states occurs at every trial with probability p_i at i^{th} state. $P(x) = \prod_{i=1}^{K} p_i^{x_i}$ and MLE of $\widehat{p}_i = \frac{\sum_{t=1}^{N} x_i^t}{N}$

Gaussian Density Function:

Univariate Distribution: $p(x) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, $-\infty < x < \infty$, $E(x) = \mu$ and $var(x) = \sigma^2$

MLE of parameters: $\widehat{\mu} = m = \frac{1}{N}\sum_{t=1}^{N} x^t$ and $\widehat{\sigma^2} = s^2 = \frac{\sum_{t=1}^{N}(x^t-m)^2}{N}$

Bias and MSE of estimators: If d(X) = d is the estimator of parameter θ,
Bias = $b_\theta(d) = E(d(X)) - \theta$ and MSE = $E((d(X) - \theta)^2)$

Sample Mean m: unbiased and consistent estimator of μ
$m = \frac{1}{N}\sum_{t=1}^{N} x^t$, $E(m) = \frac{1}{N}\sum_{t=1}^{N} E(x^t) = \frac{N\mu}{N} = \mu$, $var(m) = \frac{1}{N^2}\sum_{t=1}^{N} var(x^t) = \frac{N\sigma^2}{N^2} = \frac{\sigma^2}{N}$

Sample Variance: $\widehat{\sigma^2} = s^2 = \frac{\sum_{t=1}^{N}(x^t-m)^2}{N}$ and $E(s^2) = \frac{\sum_t E((x^t)^2) - NE(m^2)}{N} = \frac{N-1}{N}\sigma^2$

MSE of an estimator d: MSE = $E((d(X) - \theta)^2)$ = variance of d + (bias of d)^2
The Bayes' Estimator: Use posterior probability P(θ|X) to estimate θ. For narrow posterior distribution, we use Maximum Posterior Estimate of θ, $\theta_{MAP} = arg\,max\, P(\theta \mid X)$

Mean of normal distribution: For $x^t \sim N(\mu, \sigma^2)$ and $\theta \sim N(\mu_0, \sigma_0^2)$, $P(X|\theta) = \frac{1}{(2\pi)^{\frac{N}{2}}\sigma^N} e^{-\frac{\sum_t (x^t-\mu)^2}{2\sigma^2}}$

$p(\theta) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} e^{-\frac{(\theta-\mu_0)^2}{2\sigma_0^2}}$, $E(\theta|X) = \frac{\frac{N}{\sigma^2}}{\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}}\mu + \frac{\frac{1}{\sigma_0^2}}{\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}}\mu_0$

Parametric Classification:
For each class i from 1 to K, compute posterior P(C_i|x) and assign the class which has the maximum posterior.
$P(C_i|x) = \frac{P(x|C_i)\,P(C_i)}{P(x)} = \frac{P(x|C_i)\,P(C_i)}{\sum_{i=1}^{K} P(x|C_i)\,P(C_i)}$

Multivariate normal distribution: $P(X) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-0.5(x-\mu)^T\Sigma^{-1}(x-\mu)}$

For $P(x|C_i) \sim N(\mu_i, \Sigma_i)$, $g_i(x) = -\frac{d}{2}log2\pi - \frac{1}{2}log|\Sigma_i| - \frac{1}{2}(x-\mu)^T\Sigma_i^{-1}(x-\mu) + log\,P(C_i)$

Multivariate discrete features: Given all features of x as independent, Bernoulli x_j for each j from 1 to d,

Class likelihood, $P(x|C_i) = \prod_{j=1}^{d} p_{ij}^{x_j}(1 - p_{ij})^{(1-x_j)}$

Discrimination function, $g_i(x) = \sum_{j=1}^{d} (x_j log\, p_{ij} + (1 - x_j)\, log(1 - p_{ij})) + log\, P(C_i)$

Application: Document Characterization

Nonparametric approach: Estimate probability density locally if similar inputs have similar outputs, by using kernel function with local support at x and bounded integral value of 1
Univariate nonparametric density estimation: $\{x^t\}$, t = 1 to N
Estimated cumulative prob.: $F(x) = \#(x^t<x)/N$, Estimated prob. density: $P(x) = [(\#(x^t<x+h) - \#(x^t<x)) / N]/h$
Naive Estimator: $P(x) = [(\#(x^t<x+h/2) - \#(x^t<x-h/2))/N]/h$

Kernel Estimator: Kernel Function - function of distance used to determine the weight of each example

Kernel Estimator (Parzen window): $P(x) = \frac{1}{Nh}\sum_{t=1}^{N} K(\frac{x-x^t}{h})$

K(u) = 1 for |u|<½, else 0 $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$

k-NN Estimator: k-nearest neighbor, density estimate = $\frac{k}{N(2d_k(x))}$, where $d_i(x)$ = distance of i^{th} NN from x

Instance-based learning: Instances are stored for training, and a set of similar related instances are retrieved for classification/regression for testing. Computes locally.
k-NN Regression: For target function f: R^d -> R and training examples $(x^t, f(x^t))$, t = 1 to N,

For i^{th} neighbor of x (x_i), $\hat{f}(x) = \frac{\sum_{i=1}^{k} f(x_i)}{k}$

The weight used for weighted regression is inversely proportional to the square of distance $d(x, x_i)$ and proportional to a kernel function
Locally weighted regression: For linear target function $f(x) = w_0 + w_1x_1 + \ldots + w_dx_d$, weight update rule for minimization is $\Delta w_i = \eta \Sigma_{x^t \epsilon S} K(x^t, x)\,(f(x^t) - \hat{f}(x^t))\, x_i^t$

Unsupervised Learning: Learning in the absence of labels of instances, with an aim to find regularities or patterns in the input instances available
Clustering: Task of organizing objects into groups of similar members
Class: a well-studied group of objects identified by common characteristics or properties
Cluster: Collection of similar objects which are dissimilar to other clusters, have potential to form a class
Motivation of Clustering: finding representatives for homogeneous groups, and detecting unusual objects
K-Means Clustering: K such partitions are computed, which minimize the sum of squares of distances between a datapoint and its cluster's centroid.
Exhaustive K-Means: No. of ways to partition a set of N objects into K non-empty groups. Is prohibitive since the number is of exponential order, and it is an NP-hard problem
The Lloyd Algorithm: For the given K initial centers, assign a datapoint to the center closest to it, update the centers of the clusters, and repeat these steps until the center positions converge.
Conservative Approach: greedily choose the transfer of such a datapoint from class i to j, which causes maximal cost reduction at that step
Weakness of the algorithm: Can detect well-separated hyperspherical clusters only, sensitive to outliers, and may get stuck at local minima. Slow convergence or empty clusters for improper initialization.
K-means++: Randomly choose the first center c_1 and choose i^{th} center c_i as x' with probability proportional to the square of minimum distance from selected i-1 centers.
Determining optimal K: By using validity index or checking stable clustering results with random initialization.

Concepts challenging to comprehend: None

Interesting and exciting concepts:

I found Unsupervised Learning an exciting concept as it could classify instances without having any label for training set. This seems similar to human intelligence, as the model takes time to learn and compute the results.

Concepts not understood: None.

Any novel idea of yours out of the lessons:

In Unsupervised Learning, since training is without any prior knowledge of labels, results might be less accurate. So, we should use a model built on the thin line between supervised and unsupervised learning, i.e., it uses a combination of labeled and unlabeled data, determines class of unlabeled data using labeled data points, and then, it is tried on the test dataset. This would build a much better model which yields high accuracy.
