An Application of Machine Learning to Grading

Gautam Singh

March 29, 2023

Outline

1. Introduction
2. Resources
3. Grading Using the Gaussian Method
4. Grading Using the K-Means Method
5. Results
6. Conclusions

Aim

Compare grade distribution obtained by using the K-means algorithm to the grade distribution obtained using a standard normal distribution.

1. Which method is fairer?
    1. Courses with skewed performance?
    2. Courses with less students :)?
2. Which method is faster to compute grades?
3. Which method reflects student efforts better?
4. What about failing students?
5. Which method can be extended to assess based on other factors?

Resources

Marks datasheet and relevant Python codes can be found here.

1. Marks of students: `marks.xlsx`
2. Python code using Gaussian method: `grades_norm.py`
3. Python code using K-means method: `grades.py`

Data Visualization

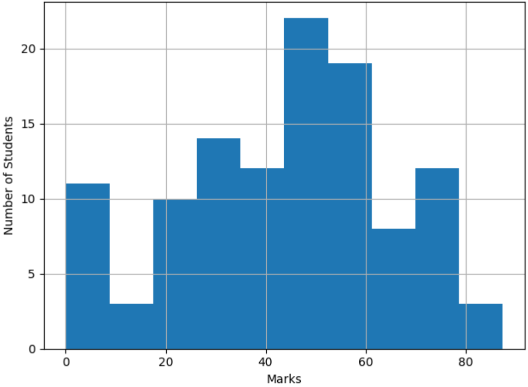


Figure: Histogram showing distribution of marks of the students.

Population Measures

Consider a dataset $\{\mathbf{x}_i\}_{i=1}^{N}$.

1. The **population mean** is given by

$$\boldsymbol{\mu} \triangleq \mathbb{E}\left[\mathbf{x}\right] \tag{1}$$

2. The **population covariance matrix** is given by

$$\boldsymbol{\Sigma} \triangleq \mathbb{E}\left[\left(\mathbf{x}-\boldsymbol{\mu}\right)\left(\mathbf{x}-\boldsymbol{\mu}\right)^{\top}\right] \tag{2}$$

Sample Measures

Consider a sample $\{\mathbf{y}_i\}_{i=1}^n$ drawn from the earlier dataset ($n \ll N$).

1. The **sample mean** is given by

$$\bar{\mathbf{x}} \triangleq \mathbb{E}[\mathbf{y}] \tag{3}$$

2. The **sample covariance matrix** is given by

$$\mathbf{s} \triangleq \frac{n}{n-1}\mathbb{E}\left[(\mathbf{y}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^\top\right] \tag{4}$$

Note that the sample measures are **unbiased estimators** of their corresponding population measures.

The Z-score

1. We assume that the number of students is large and the distribution of their parameters follows a normal distribution with population mean $\boldsymbol{\mu}$ and population covariance $\boldsymbol{\Sigma}$.
2. The Z-score of a student given their parameters $\mathbf{x}$ is given by

$$\mathbf{Z} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}} \left(\mathbf{x} - \boldsymbol{\mu}\right) \tag{5}$$

Statistical Note

If the population is large, computing population parameters directly is cumbersome. In this case, use the sample parameters to calculate the Z-score.

Application

In this case, data is one dimensional (marks of the student). Also, the population size is small enough to directly compute population measures.

1. The Z-score in this case will be

$$Z = \frac{x - \mu}{\sigma} \tag{6}$$

   where x denotes the marks of the student.

2. The runtime in this case is $O(N)$.
3. The marks were scaled relative to the highest scoring student.

Grading Scheme

Interval	**Grade**
$(-\infty, -3]$	F
(-3, -2]	D
(-2, 1]	C
(-1, 0]	B-
(0, 1]	B
(1, 2]	A-
(2, 3]	A
$(3,\infty)$	A+

Table: Grading scheme used for calculation of Z-scores

The K-Means Algorithm

1. It is an **unsupervised** learning algorithm.
2. It is a **classification** algorithm.
3. It is an **EM** algorithm (explained ahead).

Definitions

Consider a dataset $\{\mathbf{x}_n\}_{n=1}^{N}$ and K means $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$.

1. We define binary indicator variables r_{nk} for $1 \le n \le N,\ 1 \le k \le K$ as

$$r_{nk} \triangleq \begin{cases} 1 & k = \arg\min_j \left\|\mathbf{x}_n - \boldsymbol{\mu}_j\right\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

2. The cost function is given by

$$J \triangleq \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\|\mathbf{x}_n - \mu_k\right\|^2 \tag{8}$$

3. We are required to find $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ such that (8) is minimized.

Working of the K-Means Algorithm

The K-Means algorithm is an EM algorithm. Initally, we choose an arbitrary set of means. In each iteration, there are two steps.

1. *E-step*: Here, we calculate all the r_{nk} as defined in (7).
2. *M-step*: We set

$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_{n=1}^{N} r_{nk}\mathbf{x}_n}{\sum_{n=1}^{N} r_{nk}} \tag{9}$$

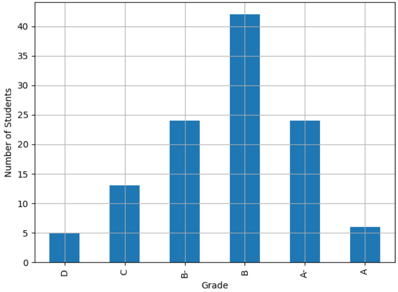
What if a Cluster is Empty?

If we encounter a k such that $r_{nk} = 0 \; \forall \; 1 \leq n \leq N$, we can either

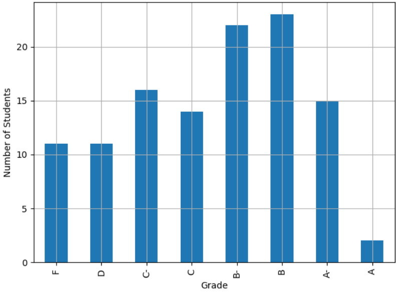
1. Discard the cluster (by setting $K \leftarrow K - 1$)
2. Selecting a point "far away" from all clusters.

Application

1. In this case, $K = 8$ and $N = 114$. The algorithm converged in 5 iterations.
2. The runtime in this case is $O(NK)$ per iteration.
3. The marks were scaled relative to the highest scoring student.



(a) Standard Normal Distribution



(b) K-Means Algorithm

Figure: Comparison of grading distributions using both algorithms.

Conclusions

1. Grading on a Gaussian curve failed less (in fact zero) students than in the case of grading using the K-means algorithm.
2. Grading on a Gaussian curve is faster for larger datasets, and both algorithms would have very little difference.
3. The K-Means algorithm gives a better idea of the performance of the class, especially when it is skew.
4. The K-Means algorithm can be extended to involve other factors such as attendance, prerequisites completed, and so on.