Mathematical Foundations for AI and Data Science

Nicolas Jorquera

This assignment delves into key mathematical concepts such as norms, inner products, convexity, and optimization techniques essential for machine learning and data analytics. Through a series of problems, it provides a hands-on approach to understanding how these mathematical tools are applied in AI and Data Science, from vector calculations to time-series forecasting.

Problem 1

Calculate the squared L-2 norm $(||x||_2)^2$ of $x = [2, 4, 6, 8]$?

$$\begin{aligned}(||x||_2)^2 &= (2^2 + 4^2 + 6^2 + 8^2)^2 \\ &= (4 + 16 + 36 + 64) \\ &= 120\end{aligned}$$

L-2 Norm Calculation: Norms are fundamental in machine learning algorithms like k-NN and SVM. The squared L-2 norm helps understand the "distance" between data points, aiding in clustering and classification tasks.

Problem 2

Calculate the L-1 norm $||x||_1$ of $x = [3, -6, 9, 12, 15]$? Show the steps you took to calculate this.

$$\begin{aligned}||x||_1 &= |3| + |-6| + |9| + |12| + |15| \\ &= 3 + 6 + 9 + 12 + 15 \\ &= 45\end{aligned}$$

L-1 Norm Calculation: The L-1 norm is widely used in feature selection and sparse modeling, which are essential in high-dimensional data problems often encountered in Data Science.

Problem 3

Calculate the L-infinity norm $||x||_\infty$, where $x = [-1, -11, 10, 4, 4]$? Show the steps you took to calculate this.

The L_∞ norm of a vector is the maximum absolute value of its elements.

$$\begin{aligned}||x||_\infty &= \max\{|-1|, |-11|, |10|, |4|, |4|\} \\ &= \max\{1, 11, 10, 4, 4\} \\ &= 11\end{aligned}$$

L-Infinity Norm: This norm is utilized in optimization problems and machine learning algorithms where you need to find the element with the maximum magnitude, such as in Chebyshev distance calculations.

Problem 4

Calculate the inner product, $x^T y$, for each of the following pairs of vectors. Please show your work.

Part A

$x = [2, 1, 2, 1, 2]$, $y = [1, 2, 1, 2, 1]$

$$\begin{aligned} x^T y &= 2*1 + 1*2 + 2*1 + 1*2 + 2*1 \\ &= 2 + 2 + 2 + 2 + 2 \\ &= 10 \end{aligned}$$

Part B

$x = y$, $y = [1, 1, 0, 1, 1]$

$$\begin{aligned} x^T y &= 1*1 + 1*1 + 0*0 + 1*1 + 1*1 \\ &= 1 + 1 + 0 + 1 + 1 \\ &= 4 \end{aligned}$$

Inner Product: Inner products are crucial in machine learning for tasks like similarity computation between vectors, which is vital for algorithms like k-NN and in techniques such as Principal Component Analysis (PCA).

Problem 5

Specify the domain of convexity. Identify whether or not the following functions are convex $\forall x \in \mathbb{R}$. If they are only convex on a certain domain, specify that domain, if it exists.

a. $f(x) = x^3 + 2x$

This function is not convex because its second derivative, $f''(x) = 6x$, changes sign from negative to positive at $x = 0$. Therefore, it is not non-negative for all $x \in \mathbb{R}$.

b. $f(x) = |x–10|^2$

This function is convex as its second derivative, $f''(x) = 2$, is non-negative for all $x \in \mathbb{R}$.

c. $f(x) = (\ln(x + 1))^2$

The function is convex for $x > -1$ since the second derivative is positive.

d. $f(x) = \tanh^2(x)$

This function is convex as its second derivative is non-negative for all $x \in \mathbb{R}$.

e. $f(x) = x \ln(x) + (1 - x) \ln(1 - x)$

This function is convex on the domain $(0, 1)$.

Convexity Analysis: Understanding the convexity of functions is essential for optimization problems in AI, especially when employing gradient-based methods to find minima or maxima, like in neural network training.

Problem 6

Determine whether or not the set is convex. Is the following sentence True or False? Why? Provide an informal proof substantiating your answer. The set of points $\{(x, y) : x^2 + y^2 \leq a\}$ is a convex set.

True. The set of points represents a disk (including the boundary) in the Euclidean plane. For any two points in the set, the line segment connecting them is also entirely within the set, which is the definition of a convex set.

Convex Sets: The concept of convex sets is foundational for understanding the feasibility and solution spaces in optimization problems, especially in linear programming methods used in operations research and AI.

Problem 7

Understanding the Softmax Function. The softmax function (shown below), $\sigma : \mathbb{R}^n \rightarrow [0, 1]$, is a generalization of the logistic function; it normalizes an input vector, x in $\mathbb{R}^n$, into a probability distribution.

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, \quad i = 1, \ldots, n$$

a. Simply write out the softmax function for when n = 5. (Hint: there should be five expressions.)

$$\sigma(x)_1 = \frac{e^{x_1}}{e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4} + e^{x_5}}$$
$$\sigma(x)_2 = \frac{e^{x_2}}{e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4} + e^{x_5}}$$
$$\sigma(x)_3 = \frac{e^{x_3}}{e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4} + e^{x_5}}$$
$$\sigma(x)_4 = \frac{e^{x_4}}{e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4} + e^{x_5}}$$
$$\sigma(x)_5 = \frac{e^{x_5}}{e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4} + e^{x_5}}$$

b. Use this function to transform the vector $x = [10, 5, 1, 2, 1]$ into a probability distribution. What is the output of the function?

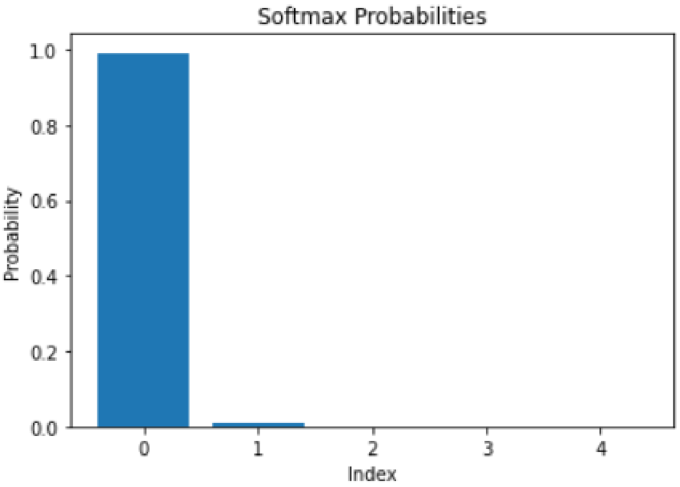
```python
import numpy as np

def softmax(x):
    return np.exp(x) / np.sum(np.exp(x))

x = np.array([10,5,1,2,1])
y = softmax(x)
```

$$
\begin{aligned}
y[0] &= 0.992733 \quad (99.27\%)\\
y[1] &= 0.006689 \quad (0.67\%)\\
y[2] &= 0.000123 \quad (0.01\%)\\
y[3] &= 0.000333 \quad (0.03\%)\\
y[4] &= 0.000123 \quad (0.01\%)
\end{aligned}
$$

c. Plot the output probability distribution (histogram).



Softmax Function: The softmax function is pivotal in classification problems, often used in the output layer of neural networks for multi-class classification, transforming raw scores into probabilities.

Problem 8

Set up the optimization problem your ML algorithm would solve. You are given time series data on purchases of cups of vanilla ice cream (assume only one, standard size) made by the end of each day, t_i, at an ice cream shop; $T = t_0, t_1, \ldots, t_n$, $Y = y_0, y_1, \ldots, y_n$ where t_i is the total number of days the shop has been open, and y_i is the total number of cups of ice cream purchased by the i^{th} day. Your goal is to forecast the number of cups of ice cream y_{i+1} on day t_{n+1}. **Key Assumptions** : There is seasonality (it's an ice cream shop…), The data has a non-stationary average, The data has been cleaned, and there are no measurement issues, There are no refunds

a. **Determine the model.** What kind of model would you use to forecast y_{i+1}? Why?

Given the information above; the SARIMA model would perform the best. The SARIMA model can capture both trend and seasonality in the data. Seasonality is crucial in this context because ice cream sales are likely to have strong seasonal patterns (e.g., higher sales in summer, lower in winter). It can also handle non-stationary data, which is one of the assumptions above. The integrated part of SARIMA can also help make the time series stationary by differencing the series before applying the ARMA model. Differencing is a method of transforming a time series dataset. It can be used to remove the series dependence on time, and is computed by subtracting the previous observation from the current one.

b. **Determine the function to optimize.** What function would you minimize or maximize? Why?

The function to minimize in this case would be the Mean Squared Error (MSE) loss function. The MSE function is used to measure the average of the squares of the errors. It is one of the most commonly used loss functions in regression problems, including time series forecasting, due to its ability to heavily penalize larger errors.

The loss function is defined as:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(t_i;\theta))^2$$

where:

$L(\theta)$ is the loss function, n is the total number of data points, y_i is the actual number of ice cream cups sold on day i, $f(t_i;\theta)$ is the predicted number of ice cream cups sold on day i, θ represents the parameters of the SARIMA model.

c. **Determine constraints.** What constraints do you have on the variables? Why? How does each constraint represent or capture real world constraints?

- *Non-negativity:* y_i must be non-negative ($y_i \geq 0$), as it represents a count (number of cups of ice cream sold), and it's impossible to sell a negative number of cups.
- *Integer values:* Another constraint is that y_i must be an integer ($y_i \in \mathbf{Q}$), as it's impossible to sell a fraction of a standard size ice cream cup.
- *Upper limit:* There might be an upper limit to the number of cups of ice cream that can be sold in a day (e.g., due to production or supply constraints). If such an upper limit L exists, we would have the constraint $0 \leq y_i \leq L$.

Time Series Forecasting: Forecasting is a critical component of data science, especially in fields like finance and supply chain management. Understanding seasonality and optimization techniques like SARIMA is vital for predictive modeling.

Problem 9

You are given the function $f(x) = \sum_{j=0}^{n} a_i x^{2i}$ where $x_i, a_i \in \mathbb{R}$, $i, n \in \mathbb{N}$, and x^i means "x to the power of i". For example,

$$f(x) = \sum_{n=2}^{i=0} a_i x^{2i} = a_0 x^0 + a_1 x^1 + a_2 x^2$$

I believe the example above is wrong, since the given function should only produce even powers.

Part A

For what values of n would you use Gradient Descent vs Stochastic Gradient Descent, given that you know the function? Why?

Gradient Descent (GD) and Stochastic Gradient Descent (SGD) usually depends on the size of the dataset rather than the degree of the polynomial (the value of n). However, if n is directly correlated with the size of the data (needing to store the values of each element in the summation: a_0x^0, a_1x^1, etc...); then the complexity of the function increases with n. For smaller values of n, it is feasible to use traditional Gradient Descent because it's less computationally expensive and the problem size is manageable.

However, as n becomes large, each iteration of GD becomes more computationally expensive because GD requires computation over the entire dataset to update the model parameters. On the other hand; Stochastic Gradient Descent can be a more efficient choice since SGD updates the parameters using only a single observation (or mini-batch) at each iteration, which can be computationally less expensive when n is large.

Therefore, for smaller values of n, Gradient Descent could be preferable, whereas for larger n, Stochastic Gradient Descent might be a better choice due to its efficiency on large datasets.

Part B

What is the global minimum of this function when n is odd vs when it's even? (Hint: x is NOT a random variable. Use a graphing calculator and common mathematical sense.)

Since all the powers of x are even in the polynomial, this means that the function is always non-negative; as $x^{2i} \geq 0$ for all real x and natural i). Therefore for both even and odd n values; the highest power in the polynomial is 2n, an even number. This function behaves like a parabola for large enough x and goes to infinity as x approaches positive or negative infinity.

However, if the function behaves like the example at the top of the page; where $f(x)$ can be either an even or odd function; then the value will not have a global minimum when $f(x)$ is negative since the function approaches negative infinity when x approaches negative infinity. It would however have a local minimum.

Gradient Descent Variants and Global Minimum: Understanding the optimization techniques like Gradient Descent and Stochastic Gradient Descent is crucial for training machine learning models efficiently. Knowing when to use each can significantly impact the performance and speed of a learning algorithm.