

# Lecture 4: Playing with gradient methods

## 一、背景及问题描述

本报告旨在通过实现并分析几种常见优化算法的数值表现，深入探讨梯度下降法及其变种（如梯度投影法、最小二乘法、正则化等）的应用。优化问题在多个领域（如机器学习、数据拟合、支持向量机等）中具有重要作用，尤其是在高维数据处理中，求解最优解或者近似最优解对模型的表现至关重要。

在本文中，主要通过实现以下优化算法进行数值实验：

- 梯度下降法 (Gradient Descent, GD)
- 梯度投影法 (Projected Gradient Descent, PGD)
- 最小二乘问题的求解 (Least Squares)
- Tikhonov正则化 (L2 Regularization)
- 套索 (Lasso)
- 支持向量机 (SVM) 优化

每种算法都通过不同的目标函数来测试其性能，重点分析目标函数的性质、优化方法及其在数值实验中的表现。

## 二、模型分析

### 1. 目标函数与其性质

- 二次函数 (Quadratic Function) :**

- 目标函数：

$$f(x) = \frac{1}{2} x^T x$$

- 特性：这是一个凸函数，并且是光滑的。梯度为

$$\nabla f(x) = x$$

，具有唯一的最小值。最优解为  $x = 0$ ，且是全局最小值。

- 最小二乘问题 (Least Squares Problem) :**

- 目标函数：

$$f(x) = \frac{1}{2m} \|Ax - b\|^2$$

- 特性：这是一个凸函数（若A是满秩矩阵）。梯度为

$$\nabla f(x) = \frac{1}{m} A^T (Ax - b)$$

，当目标函数为平凡线性模型时，存在唯一解。若加上噪声，解趋向最小二乘解。

- L2正则化 (Tikhonov Regularization) :**

- 目标函数：

$$f(x) = \frac{1}{2m} \|Ax - b\|^2 + \frac{\alpha}{2} \|x\|^2$$

- 特性：这是一个强凸函数，因为正则化项增加了函数的严格凸性。梯度为

$$\nabla f(x) = \frac{1}{m} A^T (Ax - b) + \alpha x$$

，确保了较好的数值稳定性。

- **套索 (Lasso) :**

- 目标函数:

$$f(x) = \frac{1}{2m} \|Ax - b\|^2 + \alpha \|x\|_1$$

- 特性：套索函数的目标函数包含L1范数，属于非光滑函数，其梯度是次梯度。该问题会产生稀疏解，能有效地执行特征选择。

- **支持向量机 (SVM) :**

- 目标函数:

$$f(x) = \frac{1}{m} \sum_{i=1}^m \max(1 - y_i(A_i x), 0) + \frac{\alpha}{2} \|x\|^2$$

- 特性：这是一个非光滑目标函数，因为存在最大函数。目标函数在一定的约束条件下最小化，支持向量机求解时需要考虑次梯度。

## 2. 迭代方法与分析

- **梯度下降法 (Gradient Descent, GD) :**

- 方法：**根据目标函数的梯度信息来迭代更新解。每次更新的步长由当前的梯度方向和步长决定。
- 收敛性：**对于凸目标函数，梯度下降法能够找到全局最优解。在强凸函数上，梯度下降法的收敛速度是线性的。
- 步长选择：**步长的选择对梯度下降法的性能有重要影响。若步长过大，可能导致发散；若步长过小，可能导致收敛过慢。因此，通常采用逐步减小的策略或者根据目标函数的光滑性来选择合适的步长。

- **梯度投影法 (Projected Gradient Descent, PGD) :**

- 方法：**在每次梯度更新后，将结果投影到可行域内。若目标函数有约束，梯度下降更新后通过投影操作确保解仍然在约束域内。
- 投影问题：**在此实验中，投影操作是将更新后的解映射回一个仿射子空间，该子空间是由矩阵U的列生成的子空间。

- **最小二乘问题:**

- 该问题可通过直接求解正规方程得到最优解。使用梯度下降法解决时，每次更新解时的步长需要根据目标函数的特性进行调整。

## 三、数值实验与结果分析

在本实验中，我们通过梯度下降法及其变种（如梯度投影法）解决了不同的优化问题，并对算法的性能进行了评估。

- **误差曲线：**我们绘制了每次迭代的误差曲线（即目标函数值或残差的变化）。从图中可以看出，随着迭代次数的增加，误差逐渐减小，表明算法有效地逼近了最优解。

- **梯度下降法的步长选择**：在优化过程中，步长的选择显著影响了算法的收敛速度。对于二次函数，较大的步长有时会导致发散，特别是在高维数据的情况下。通过逐步减小步长，能够避免这种情况的发生。
- **梯度投影法的效果**：对于有约束的优化问题（如仿射子空间约束），梯度投影法能够有效地将解限制在约束域内，从而提高了优化的稳定性和准确性。

从结果图中可以看到，最小二乘法和套索方法的收敛速度较快，而梯度投影法在高维空间中的表现也较为稳定。SVM优化则由于目标函数的非光滑性，其收敛速度较慢，但通过次梯度法依然能够获得良好的结果。

## 四、问题与解决方案

在实验过程中，我们遇到了一些问题并采取了相应的解决措施：

1. **步长选择问题**：过大的步长导致了优化过程中的不稳定，甚至是发散。解决方案是通过逐步减小步长或者使用自适应步长方法来提高收敛性。
2. **非光滑问题**：如Lasso和SVM优化问题中的L1范数和最大函数，直接使用梯度下降法会导致问题。我们通过使用次梯度方法（subgradient method）解决了这些问题，并取得了较好的结果。
3. **计算代价问题**：对于大规模问题，计算梯度和投影操作可能会变得计算量大。解决方案是采用批量更新（batch update）和近似方法来减少计算代价。