THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

SEAS 8510
Analytical Methods for Machine Learning

Lecture 4

Dr. Zachary Dennis

Agenda

9:00 – 9:15	Discussion Groups (15 min)
9:15 – 10:00	Homework 3 Review (45 min)
10:00 – 10:15	ML Applications of Optimization (15 min)
10:15 – 10:25	*BREAK (10 min)*
10:25 – 11:45	Gradient Descent, and Lagrange Multipliers, and Convex Optimization (80 min)
11:45 – 12:00	Test Material Overview (15 min)

Assignments

Last week: Discussion 3/Homework 3

This week: No Homework or Discussion Due

Midterm opens on 4/20 at 8 PM Eastern

Discussion 3

Prompt:

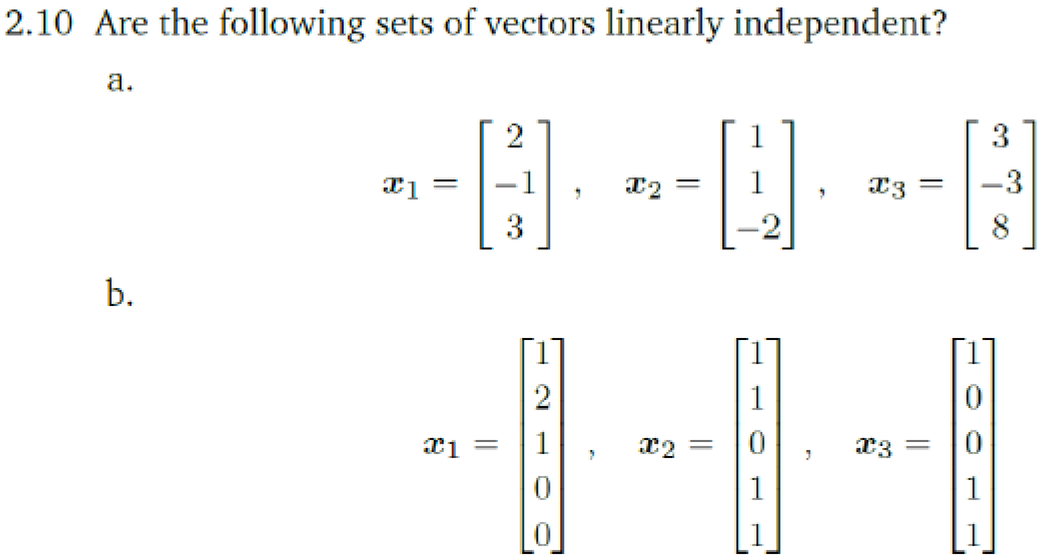
Describe how each of the following algorithms work from a high level and describe at least one use case for each – (a) singular value decomposition, (b) principal component analysis, (c) linear least squares regression, and (d) network analysis.

For people with practical experience in machine learning, feel free to provide real world examples of implementations on real or fake data sets besides what was shown in class.

Instructions:

- Be randomly assigned to breakout rooms
- In small group, meet and discuss your responses (15 min)

Homework 3



2.10 Are the following sets of vectors linearly independent?

a.

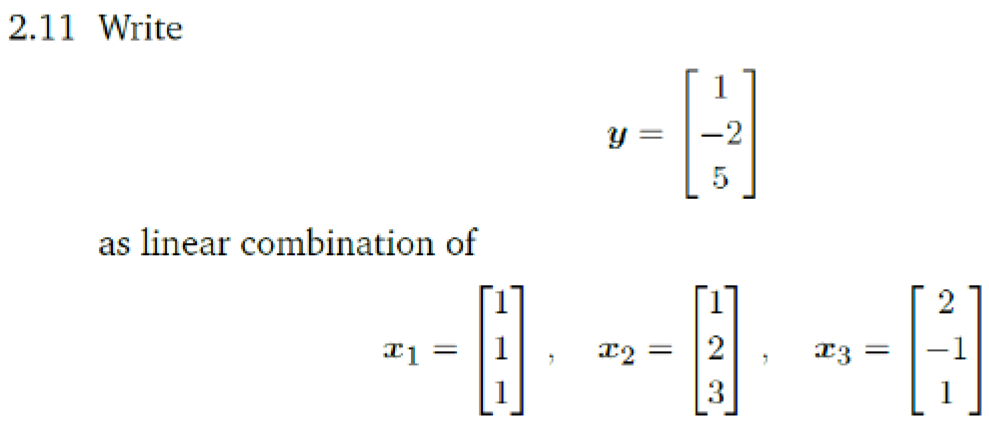
$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

b.

$$x_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

1. Check if the 0-vector can be non-trivially represented as a linear combination of x_1,x_2,x_3
    a) Solve $\sum_{i=1}^{3} \lambda_i x_i = \mathbf{0}$
2. If it can, then the three vectors are linearly dependent.

Homework 3



2.11 Write

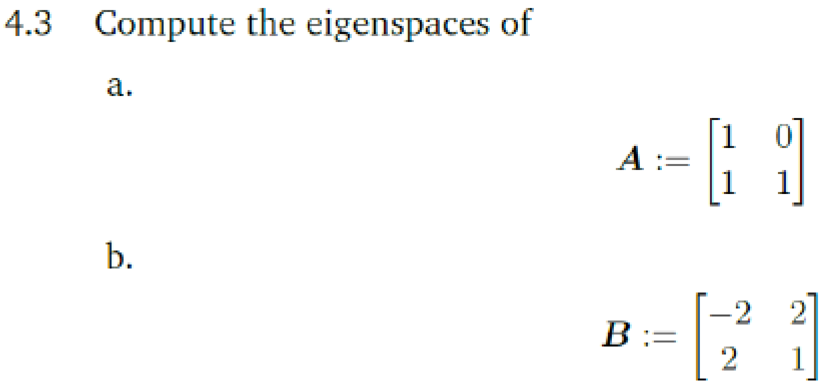
$$\boldsymbol{y} = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$$

as linear combination of

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

1. Solve $\sum_{i=1}^{3} \lambda_i x_i = \boldsymbol{y}$ using Gaussian Elimination

Homework 3



4.3 Compute the eigenspaces of

a.

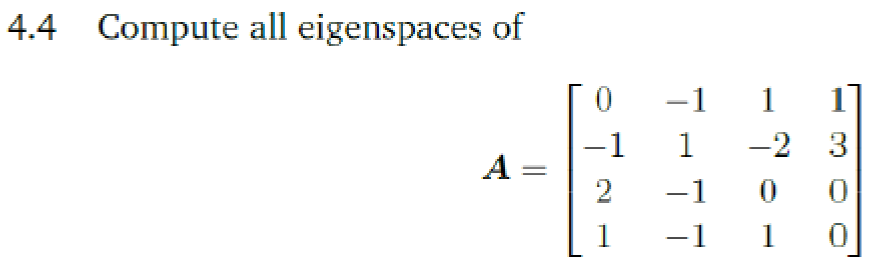
$$A := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

b.

$$B := \begin{bmatrix} -2 & 2 \\ 2 & 1 \end{bmatrix}$$

1. Find eigenvalues using $|A - \lambda I| = 0$
2. For each λ, calculate $(A - \lambda I)x = \mathbf{0}$

Homework 3

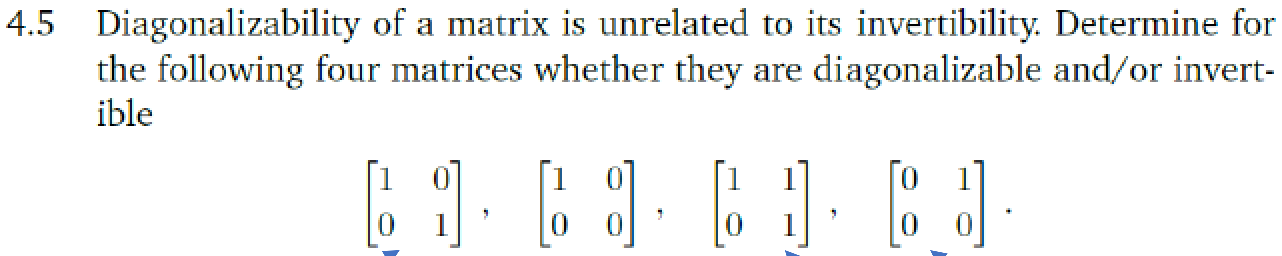


4.4 Compute all eigenspaces of

$$\boldsymbol{A} = \begin{bmatrix} 0 & -1 & 1 & 1 \\ -1 & 1 & -2 & 3 \\ 2 & -1 & 0 & 0 \\ 1 & -1 & 1 & 0 \end{bmatrix}$$

1. Find eigenvalues using $|A - \lambda I| = 0$
2. For each λ, calculate $(A - \lambda I)x = \mathbf{0}$

Homework 3



4.5 Diagonalizability of a matrix is unrelated to its invertibility. Determine for the following four matrices whether they are diagonalizable and/or invertible

$$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}, \quad \begin{bmatrix}1 & 1\\0 & 1\end{bmatrix}, \quad \begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}.$$

Diagonalizable and Invertible

Diagonalizable, Not Invertible

Not diagonalizable, Invertible

Not diagonalizable, Not invertible

Algebraic Mult. is the number of times the root is in the characteristic polynomial.

Geometric multiplicity is the # of linearly independent eigenvectors.

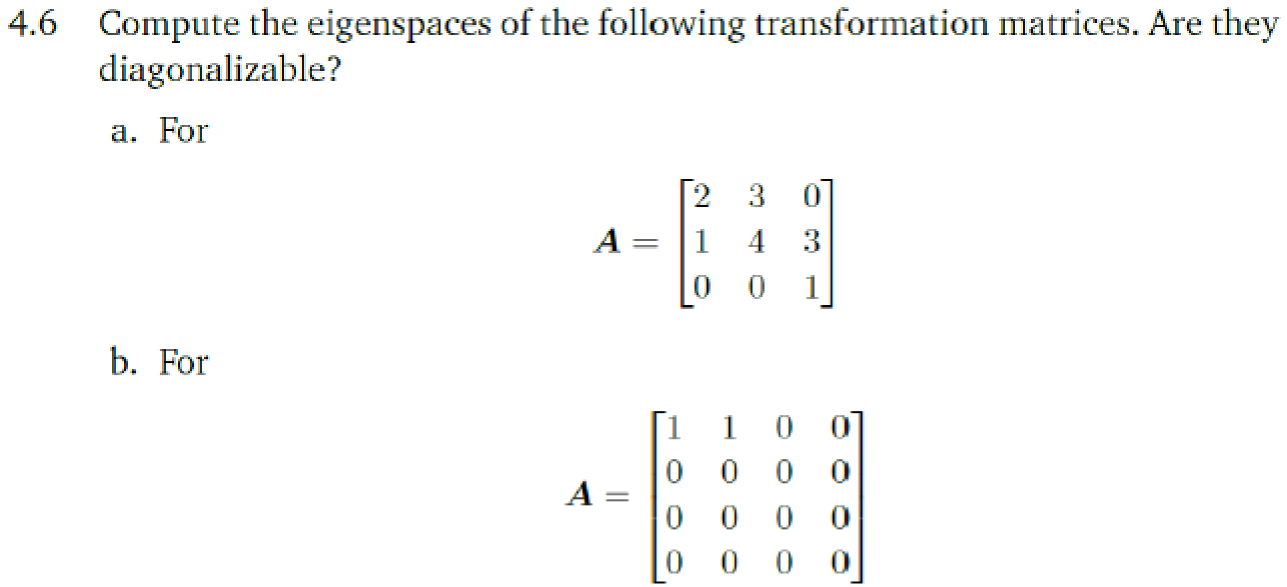
Diagonalizable Test for nxn matrix:

1. Find Eigenspaces
2. If sum of geometric multiplicities = n, then diagonalizable.
3. If sum of algebraic multiplicities = n and each eigenvalue has a geometric multiplicity = algebraic multiplicity.

Invertible Test:

1. Find determinate.
2. If the determinant of the matrix is zero then the matrix is not invertible, or else the matrix is invertible

Homework 3



4.6 Compute the eigenspaces of the following transformation matrices. Are they diagonalizable?

a. For

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

b. For

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1. Compute Eigenvalues
2. Compute Eigenvectors
3. Does sum of geometric multiplicities = n?

Geometric Multiplicity > 1

$$A = \begin{bmatrix} -2 & 2 & 2 \\ 2 & 2 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

λ_1=3

λ_2=−3

Significance of Optimization

- Minimizing cost function and improving accuracy
- Improving performance by optimizing hyperparameters
- Faster training and convergence using more efficient optimization techniques
- Finding solutions for problems involving highly complex non-linear functions
- Overcoming overfitting by optimization approaches such as regularization

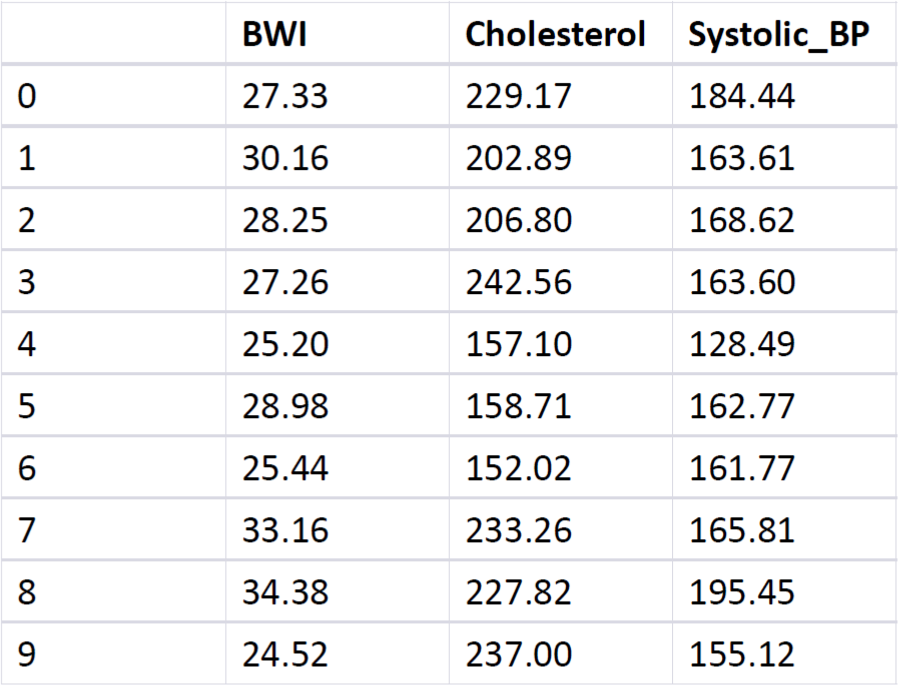
Examples of Usage

Linear Regression (Gradient Descent): In linear regression, gradient descent is a common optimization technique used to minimize the cost function. This iterative approach adjusts the coefficients of the model in order to find the best fit line that minimizes the difference between predicted and actual values.

- **Support Vector Machines (Quadratic Programming):** Support Vector Machines (SVMs) typically use quadratic programming for optimization. This method finds the optimal hyperplane that maximizes the margin between different classes in the dataset.
- **Logistic Regression (Newton's Method):** For logistic regression, optimization methods like Newton's Method, also known as the Newton-Raphson method, can be used. This approach iteratively adjusts the weights to find the best model that predicts categorical outcomes (like 0/1).
- **Neural Networks (Backpropagation and Stochastic Gradient Descent):** Neural networks often use backpropagation combined with optimization techniques like stochastic gradient descent (SGD). SGD is used to minimize the loss function by updating the weights of the network, often in conjunction with techniques like momentum or Adam optimizer to speed up convergence and improve performance.
- **Ensemble Methods (Boosting):** In ensemble methods like Gradient Boosting, optimization is used to combine multiple weak learners (like decision trees) into a strong learner. Boosting algorithms focus on sequentially improving the model by focusing on the instances that previous models misclassified.

Multilayer Perceptron Example

Given the following table, we would like to train a neural network to predict systolic blood pressure



	BWI	Cholesterol	Systolic_BP
0	27.33	229.17	184.44
1	30.16	202.89	163.61
2	28.25	206.80	168.62
3	27.26	242.56	163.60
4	25.20	157.10	128.49
5	28.98	158.71	162.77
6	25.44	152.02	161.77
7	33.16	233.26	165.81
8	34.38	227.82	195.45
9	24.52	237.00	155.12

Multilayer Perceptron Example

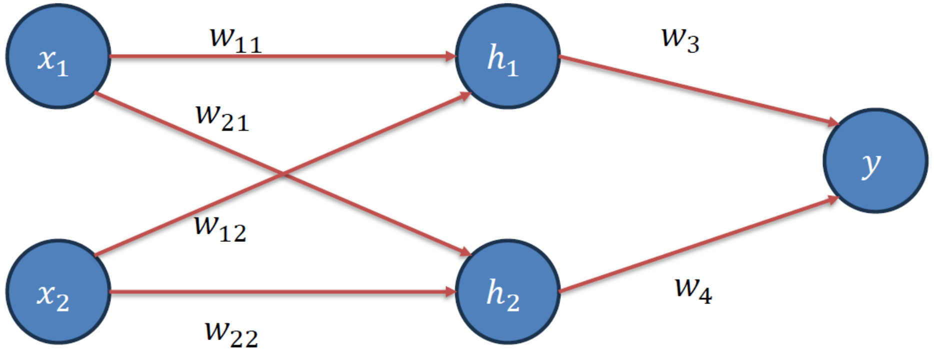
Algorithm: Initialize weights with random numbers: $w_{11}, w_{12}, w_{21}, w_{22}, w_3, w_4, b_1, b_2$

Calculate hidden layer neuron values: $h_1=\sigma(w_{11}x_1+w_{12}x_2+b_1)$ and $h2=\sigma(w_{21}x_1+w_{22}x_2+b_1)$

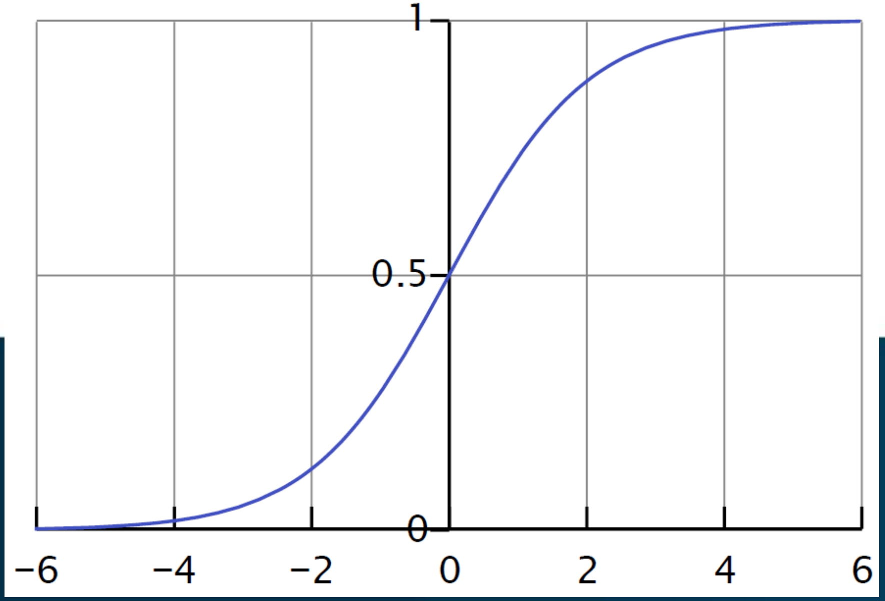
Calculate the output: $y=w_3h_1+w_4h_2+b_2$

where the sigmoid function is defined as:

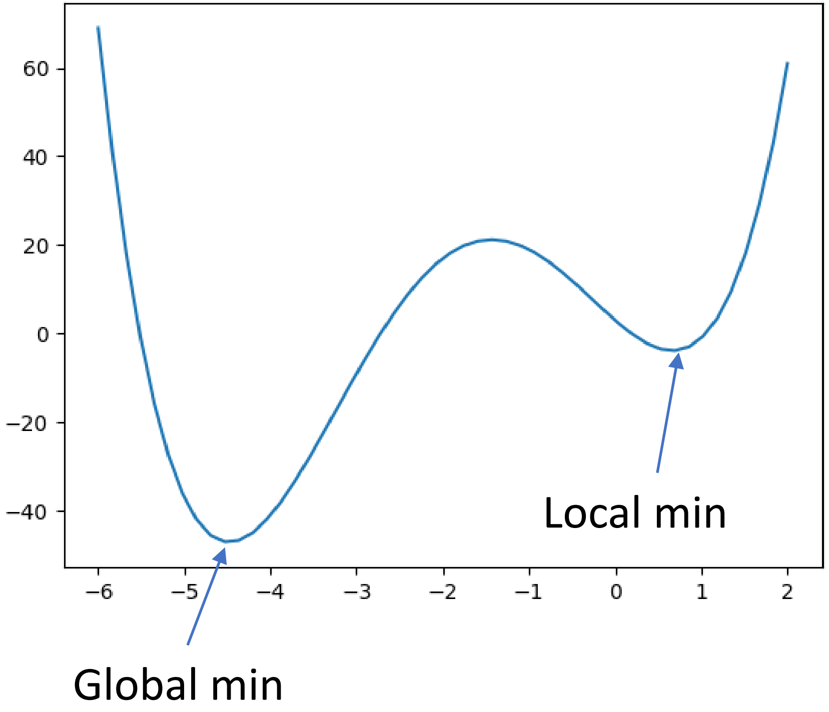
See Colab Notebook



$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Optimization

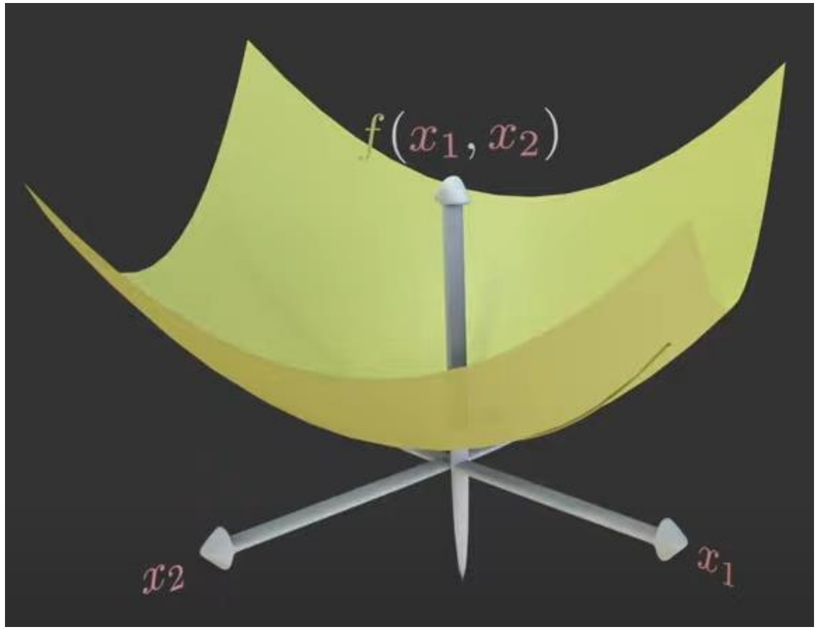


How many stationary points are there? (Slope = 0)
How do you solve for them?
How do you know if a stationary point is a min or max?

Hint: Negative makes it a max

If you can’t analytically solve for a min, what do you do?

Multidimensional Optimization



https://youtu.be/AM6BY4btj-M?si=1KSJyjIOIddXXWZi

Gradient Descent

$$min_x f(x)$$

$f(x)$ is differentiable, not feasibly solved analytically

1. Pick a point
2. Take steps proportional to the gradient. Gradient is defined as

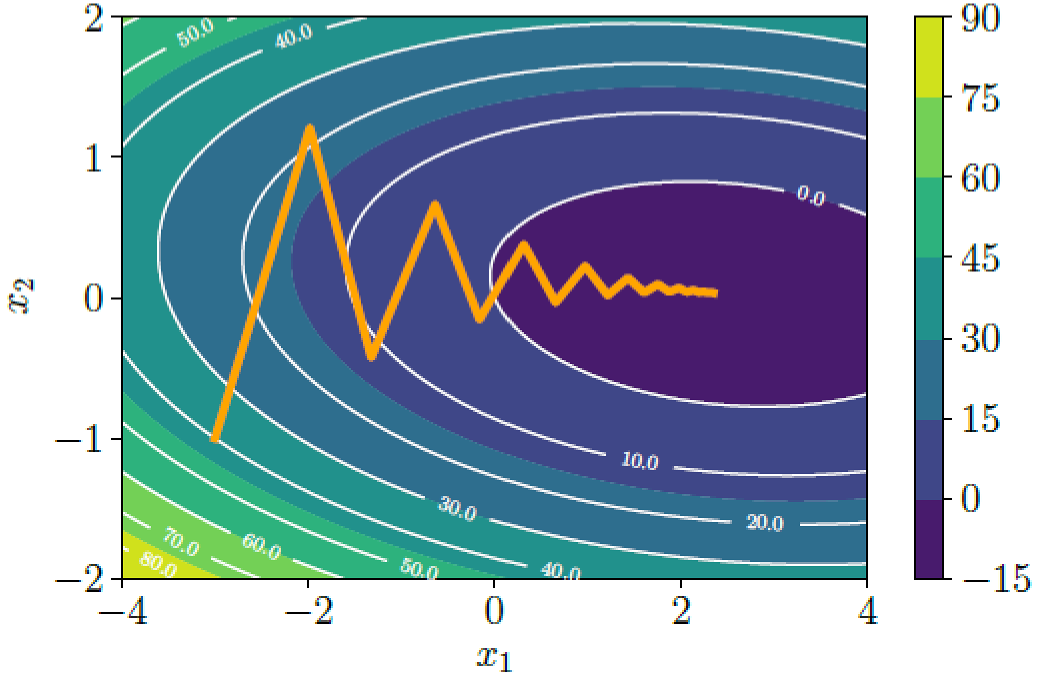
$$\nabla_{\boldsymbol{x}} f = \operatorname{grad} f = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{1\times n}$$

Gradient points orthogonally to contour lines

Gradient Descent

$$x_{i+1} = x_i - \gamma_i((\nabla f)(x_i))^\top$$

x_0 = initial guess
X_{0+1} = next step
$\gamma_i = step\ size$



Implications of Step Size

Gradient Descent with Momentum

$$x_{i+1} = x_i - \gamma_i((\nabla f)(x_i))^\top + \alpha \Delta x_i$$

$$\Delta x_i = x_i - x_{i-1} = \alpha \Delta x_{i-1} - \gamma_{i-1}((\nabla f)(x_{i-1}))^\top$$

Optimization with Constraints

Min/Max a function subject to another set of functions

$$\min f(x)$$
$$h_i(x) \ = \ 0, i \ = \ 1, \ldots$$
$$g_j(x) \ \leq \ 0, j \ = \ 1, \ldots$$

Examples?

Lagrange Multipliers

Max/min $f(x, y, z)$ subject to $g(x, y, z) = k$
Form $F(x, y, z, \lambda) = f(x, y, z) - \lambda(g(x, y, z) - k)$

$$\frac{dF}{dx} = 0, \frac{dF}{dy} = 0, \frac{dF}{dz} = 0, \frac{dF}{d\lambda} = 0$$

Sub solutions back into $f(x, y, z)$

Convex Optimization

$$\min f(x)$$
$$h_i(x) = 0, i = 1, \ldots$$
$$g_j(x) \leq 0, j = 1, \ldots$$

Feasible region is set of x that satisfies h,g

Convex Optimization

A set is Convex if a line between any two points are contained within the set.

Functions are convex if its region above the convex function is a convex set.

Optimization is convex is f(x) is convex, g(x) is convex, h(x) is linear

$$\min f(x)$$
$$h_i(x) = 0, i = 1, \ldots$$
$$g_j(x) \leq 0, j = 1, \ldots$$

Principle of Duality

Ability to view concept from primal and dual perspective.

Ex. convex sets can be determined by internal line test or by an infinite number of inequalities with hyperplanes. Convex functions can be represented by hyperplanes.

$f(x) = x^2 + 1$

$f'(x) = 0$ is minimum.

Linear Least Squares

$$\min f(x)$$
$$h_i(x) \ = \ 0, i \ = \ 1, \ldots$$
$$g_j(x) \ \leq \ 0, j \ = \ 1, \ldots$$

If all are linear, then linear program.

Least Squares

$\min ||Ax - b||^2 = \nabla ||Ax - b||^2 = 0 \qquad 2A^T(Ax - b) = 0$

$x = -(A^T A)^{-1} A^T b$

Test Topics

All topics in slides are valid, but focus on:

1. Vectors interpretations, properties, operations
2. Vector projection
3. Classification performance metrics
4. Solving systems of linear equations, specific and general solutions
5. Matrix properties, operations
6. Eigenvalues, eigenvectors, Eigenspaces
7. Linear dependence/independence
8. Matrix invertibility, diagonalizability
9. Principles of Gradient Descent, Lagrange Multipliers, Convex Optimization
10. Find a gradient using partial derivatives
11. Analytical solutions using Lagrange Multipliers