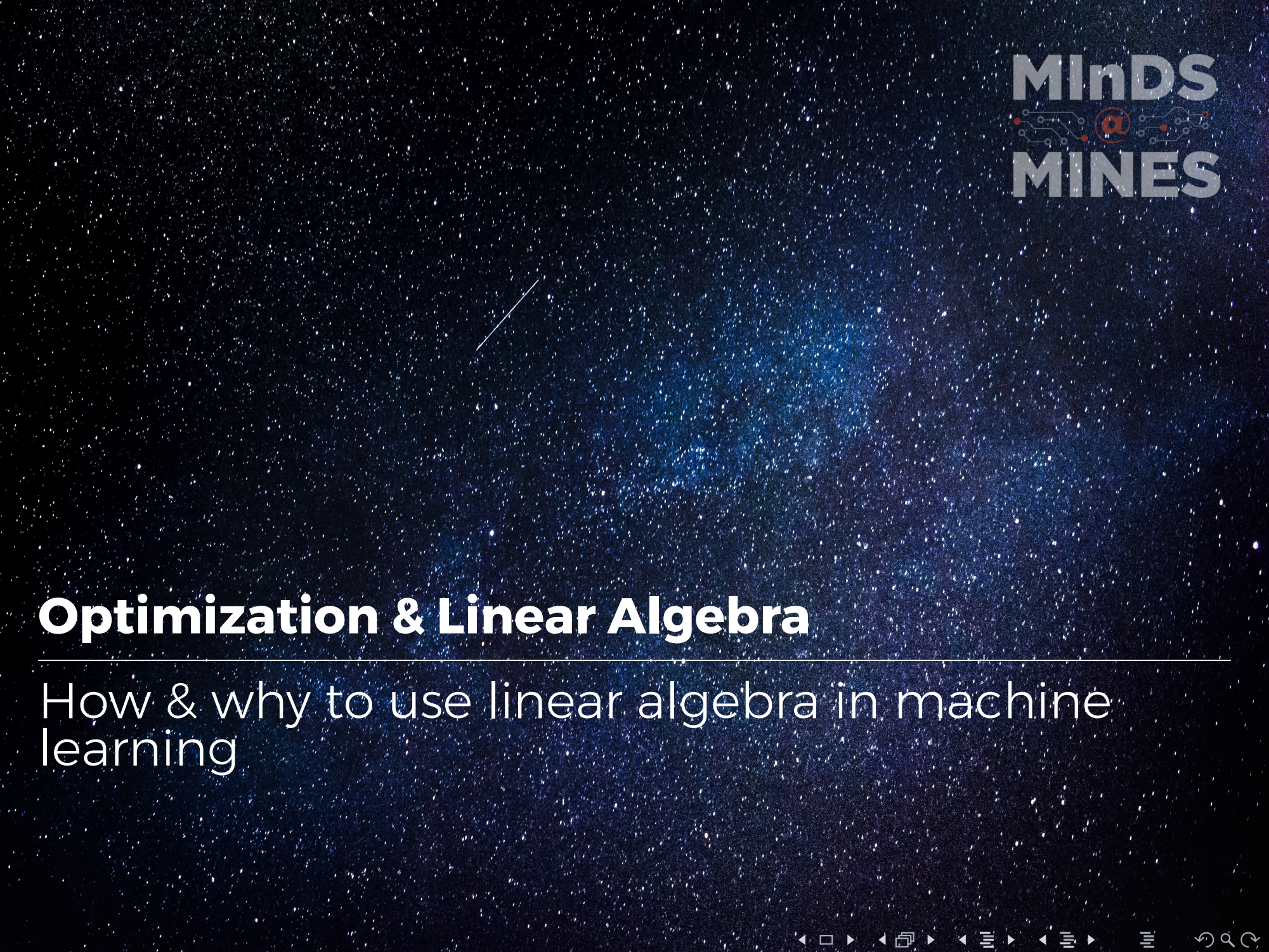


Optimization & Linear Algebra

How & why to use linear algebra in machine learning

Measurements

- **Scalars** are real numbers. $x \in \mathbb{R}$.
- **Vectors** generalize scalars in d dimensions. $\mathbf{x} \in \mathbb{R}^d$.
- **Matrices** generalize vectors in $m \times n$ dimensions. $\mathbf{X} \in \mathbb{R}^{m \times n}$.
- **Tensors** generalize matrices in any dimensions. $\mathcal{X} \in \mathbb{R}^{a \times b \times c \times \ldots}$.

Products

When taking the product of various measurements, we must **ensure that the inner dimensions align**.
A product of a measurement in $a \times b$ with a measurement in $b \times c$ results in a measurement in $a \times c$.

Products - Inner Product

$$\begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i$$

Products - Outer Product

$$\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & \dots & x_1 y_n \\ \vdots & \ddots & \vdots \\ x_m y_1 & \dots & x_m y_n \end{bmatrix}$$

Products - Matrices

$$\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 \times y_1 + x_2 \times y_4 & x_1 \times y_2 + x_2 \times y_5 & x_1 \times y_3 + x_2 \times y_6 \\ x_3 \times y_1 + x_4 \times y_4 & x_3 \times y_2 + x_4 \times y_5 & x_3 \times y_3 + x_4 \times y_6 \end{bmatrix}$$

Norms

A **norm** is a function $f(\mathbf{x})$ that assigns a strictly positive length or size to each vector in a vector space.

A norm

- is **non-negative**: $f(\mathbf{x}) \geq 0$
- is **definite**: $f(\mathbf{x}) = 0$ if and only if $\mathbf{x} = 0$
- is **homogenous**: $f(t\mathbf{x}) = |t|f(\mathbf{x})$
- follows the **triangle inequality**: $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$

ℓ_p norms

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

ℓ_1, ℓ_2 norms

$$||\mathbf{x}||_1 = \sum_i^n |x_i|$$

$$||\mathbf{x}||_2 = \sqrt{\sum_i^n x_i^2}$$

$\ell_{p,q}$ norms

$$||\mathbf{X}||_{p,q} = \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{m} |x_{ij}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} = \left(\sum_{i=1}^{n} ||\mathbf{x}^i||_p^q \right)^{\frac{1}{q}}$$

The $\ell_{p,q}$ norm of a matrix calculates the ℓ_p norm of the row vectors followed by the ℓ_q norm of the resulting vector.

Frobenius Norm

$$||\mathbf{X}||_{2,2} = ||\mathbf{X}||_F = \sqrt{\sum_{ij} x_{ij}^2} = \sqrt{\mathbf{tr}(\mathbf{X}\mathbf{X}^T)}$$

Matrix Properties & Operations

Important Matrices

$$\mathbf{I}_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$\mathbf{D}_{ij} = \begin{cases} d_i & i = j \\ 0 & i \neq j \end{cases}$$

Symmetric Matrices

A symmetric matrix $\mathbf{S}$ is one where $s_{ij} = s_{ji}$.

$$\mathbf{tr}(\mathbf{S}) = \sum_{i=1}^{n} s_{ii}$$

Rank

The rank of a matrix is the maximum number of linearly independent column vectors or linearly independent row vectors.

$$\textbf{rank}(\mathbf{X}) \leq \min(n, m) \text{where} \mathbf{X} \in \mathbb{R}^{n \times m}.$$

Transpose

If $\mathbf{Y} = \mathbf{X}^T$, then $y_{ij} = x_{ji}$.

Inverse

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Eigenvalues & Eigenvectors

$$\mathbf{X}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

Eigenvalues Explained Visually

Orthogonality

Orthogonal vectors, $\mathbf{x}^T\mathbf{y} = 0$.

Orthogonal matrix, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$

Singular Value Decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{R}^{n \times n}$ is the eigenvectors of $\mathbf{X}\mathbf{X}^T$, $\mathbf{V} \in \mathbb{R}^{m \times m}$ is the eigenvectors of $\mathbf{X}^T\mathbf{X}$, and $\mathbf{\Sigma} \in \mathbb{R}^{n \times m}$ is a diagonal of the $\sqrt{\text{eig}(\mathbf{X}\mathbf{X}^T)}$

Eigenvalue Decomposition

$$\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

where $\mathbf{V} \in \mathbb{R}^{n \times n}$ is the eigenvectors of $\mathbf{X}$, and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal of the eigenvalues of $\mathbf{X}$.

Eigenvalue Decomposition

$$\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

where $\mathbf{V} \in \mathbb{R}^{n \times n}$ is the eigenvectors of $\mathbf{X}$, and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal of the eigenvalues of $\mathbf{X}$.
For a symmetric positive matrix $\boldsymbol{S}$,

$$\text{eig}(\boldsymbol{S}^T\boldsymbol{S}) = \text{eig}(\boldsymbol{S}\boldsymbol{S}^T) = \text{eig}(\boldsymbol{S}) \circ \text{eig}(\boldsymbol{S}).$$

Eigenvalue Decomposition

$$\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

where $\mathbf{V} \in \mathbb{R}^{n \times n}$ is the eigenvectors of $\mathbf{X}$, and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal of the eigenvalues of $\mathbf{X}$.
For a symmetric positive matrix $\boldsymbol{S}$,

$$\text{eig}(\boldsymbol{S}^T\boldsymbol{S}) = \text{eig}(\boldsymbol{S}\boldsymbol{S}^T) = \text{eig}(\boldsymbol{S}) \circ \text{eig}(\boldsymbol{S}).$$

$$\sqrt{\text{eig}(\boldsymbol{S}\boldsymbol{S}^T)} = \text{eig}(\boldsymbol{S})$$

Matrix Calculus

Hessian Matrix

$$H(f) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\, \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\, \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2\, \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\, \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n\, \partial x_1} & \dfrac{\partial^2 f}{\partial x_n\, \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Convex Optimization

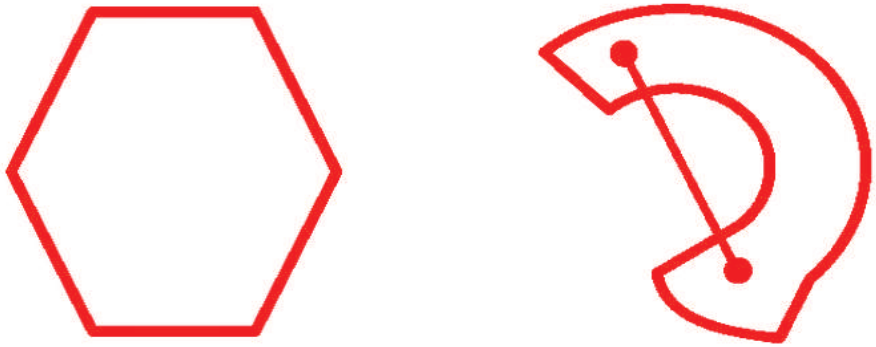
Optimization Formulation

$$\min_{\mathbf{w}} \; f(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{X}\mathbf{w} + \mathbf{c}^T\mathbf{w}$$

$$s.t. \quad \mathbf{A}\mathbf{w} \leq \mathbf{b} \quad \leftarrow \text{ inequality constraint}$$

$$\mathbf{E}\mathbf{w} = \mathbf{d} \quad \leftarrow \text{ equality constraint}$$

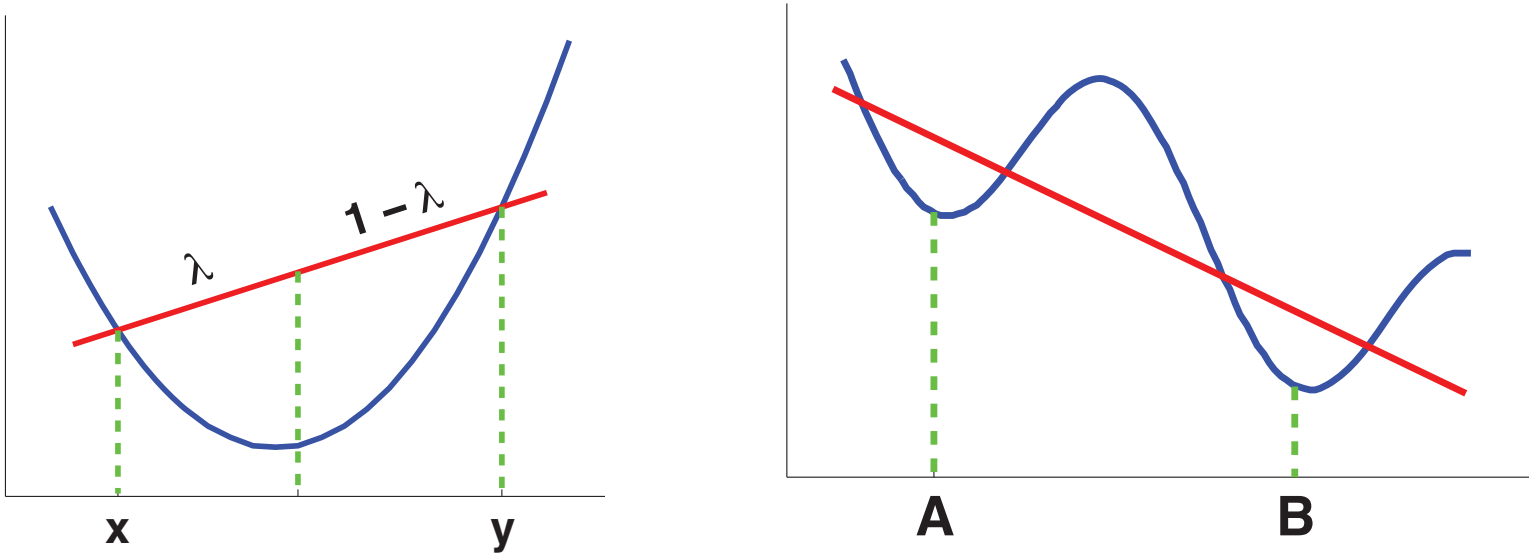
Convex Sets



(a) (b)

A set, $\mathcal{S}$, is convex, if for any $\theta, \theta' \in \mathcal{S}$, $\lambda\theta + (1-\lambda)\theta' \in \mathcal{S}, \forall\lambda \in [0,1]$.

Convex Functions



(a) (b)

A function, $f(\theta)$ is convex if it is defined on a convex set, and if any $\theta, \theta' \in \mathcal{S}$ and $\forall \lambda \in [0, 1]$, $f(\lambda\theta + (1 - \lambda)\theta') \leq f(\lambda\theta) + f((1 - \lambda)\theta')$.

Gradient Descent

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla f(\mathbf{x}_n).$$

Learning Rate

- Too large - overshoot past the optimal value
- Too small - take a very long time to reach optimal

Newton's Method

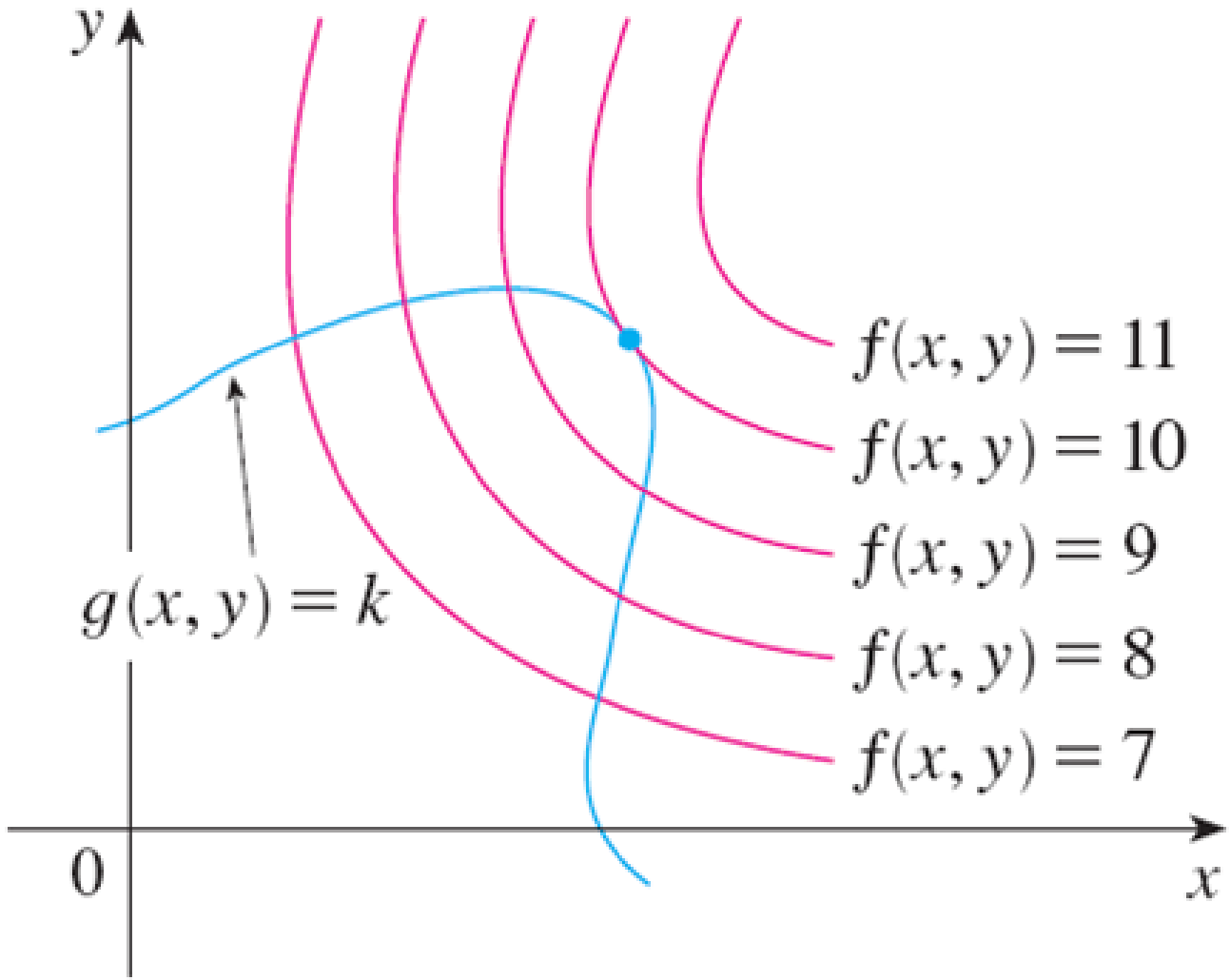
$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{f'(\mathbf{x}_n)}{f''(\mathbf{x}_n)}$$

Newton's Method

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [H(f(\mathbf{x}_n))]^{-1} \nabla f(\mathbf{x}_n),$$

Constrained Problems

Equality Constraints



Lagrangian Multipliers & Functions

Convert

$$\min_{\mathbf{x}} f(\mathbf{x}),$$
$$s.t. \quad g(\mathbf{x}) = 0,$$

to

$$\min_{\mathbf{x},\lambda} \; f(\mathbf{x}) + \lambda g(\mathbf{x}).$$

Lagrangian Function

$$\mathcal{J}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

$$\frac{\partial \mathcal{J}}{\partial \lambda} = 0 \rightarrow g(\mathbf{x}) = 0$$

Inequality Constraints

≥ 0 constraints can be converted to ≤ 0 by multiplying both sides by -1.
We can incorporate ≤ 0 constraints the same way we do with the Lagrangian function but they must abide by the KKT conditions.

Karush-Kuhn-Tucker conditions

- $g(\mathbf{x}) \leq 0$
- $\lambda \leq 0$
- $\lambda g(\mathbf{x}) \leq 0$

Questions