COMPSCI 514: Algorithms for Data Science

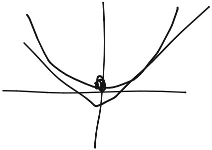
Cameron Musco

University of Massachusetts Amherst. Fall 2023.
Lecture 24 (Final Lecture!)

Logistics

- Problem Set 5 can be submitted up to 12/11 (Monday) at 11:59pm.
- Exam is next Thursday 12/14, from 10:30am-12:30pm in class.
- I am holding office hours Tuesday 12/12 1-3:30pm and Wednesday 12/13 2pm-3pm Both will be held in CS140.
- It would be really helpful if you could fill out SRTIs for this class.
- There is no quiz due this week.

Summary



Last Class:

- Analysis of gradient descent for convex and Lipschitz functions.

This Class:

- Extend gradient descent to constrained optimization via projected gradient descent.
- Course wrap up and review.

GD Analysis Proof

$\|\nabla f(x)\|_2 \leq G$

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$ $\quad \sim \leq \frac{\epsilon}{2}$

progress to θ_*

GD Analysis Proof

Theorem – GD on Convex Lipschitz Functions: For convex G-Lipschitz function f, GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon.$$

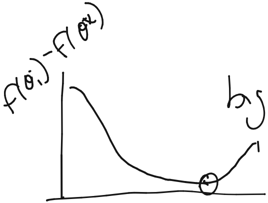
Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2} \implies$

Step 2: $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2}.$

telescoping sum

$\leq \varepsilon$ via algebra

$\frac{1}{t}\sum \theta_i$



biggest step is $G \cdot \eta$

$\hat{\theta} = \theta_t$

$\hat{\theta} = \operatorname{argmin}_{i=1,\ldots,t} f(\theta_i)$

Constrained Convex Optimization

Often want to perform convex optimization with convex constraints.

$$\vec{\theta}^* = \arg\min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}),$$

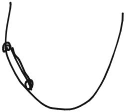
where $\mathcal{S}$ is a convex set.

Constrained Convex Optimization

Often want to perform convex optimization with convex constraints.

$S \subseteq \mathbb{R}^d$

$$\vec{\theta}^* = \arg\min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}),$$

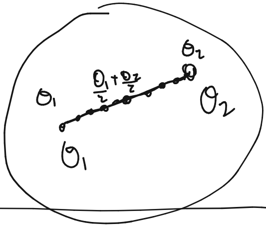


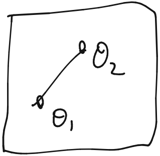
where $\mathcal{S}$ is a convex set.

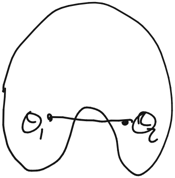
Definition – Convex Set: A set $\mathcal{S} \subseteq \mathbb{R}^d$ is convex if and only if, for any $\vec{\theta}_1, \vec{\theta}_2 \in \mathcal{S}$ and $\lambda \in [0, 1]$:

$$(1-\lambda)\vec{\theta}_1 + \lambda \cdot \vec{\theta}_2 \in \mathcal{S}$$

$\mathbb{R}^2$







Constrained Convex Optimization

Often want to perform convex optimization with convex constraints.

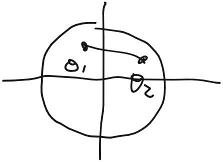
$$\vec{\theta}^* = \arg\min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}),$$

where $\mathcal{S}$ is a convex set.

Definition – Convex Set: A set $\mathcal{S} \subseteq \mathbb{R}^d$ is convex if and only if, for any $\vec{\theta}_1, \vec{\theta}_2 \in \mathcal{S}$ and $\lambda \in [0, 1]$:

$$(1-\lambda)\vec{\theta}_1 + \lambda \cdot \vec{\theta}_2 \in \mathcal{S}$$

E.g. $\mathcal{S} = \{\underline{\vec{\theta} \in \mathbb{R}^d} : \|\underline{\vec{\theta}}\|_2 \leq 1\}$. $\rightarrow$ triangle inequality



$\theta_1, \theta_2 \in S$

$\lambda\theta_1 + (1-\lambda)\theta_2 \in S$

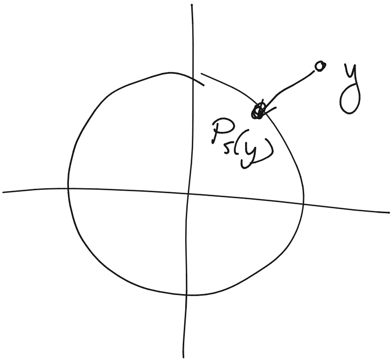
$\|\lambda\theta_1 + (1-\lambda)\theta_2\|_2 \leq 1$

$\leq \|\lambda\theta_1\| + \|(1-\lambda)\theta_2\|_2 = \lambda\|\theta_1\|_2 + (1-\lambda)\|\theta_2\|_2 \leq 1$

Projected Gradient Descent

For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto $\mathcal{S}$.

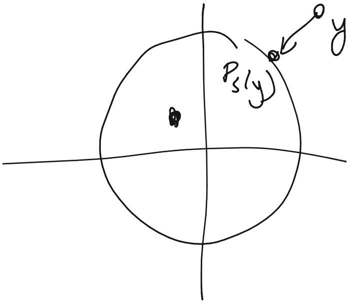
- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.



Projected Gradient Descent

For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto $\mathcal{S}$.

- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta}\in\mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- For $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_{\mathcal{S}}(\vec{y})$?



$P_S(y) =$ if $\|y\|_2 > 1$ then $\frac{y}{\|y\|_2}$

if $\|y\|_2 < 1$ y

Projected Gradient Descent

For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto $\mathcal{S}$.

- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta}\in\mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- For $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_{\mathcal{S}}(\vec{y})$?
- For $\mathcal{S}$ being a k dimensional subspace of $\mathbb{R}^d$, what is $P_{\mathcal{S}}(\vec{y})$?

Let V be an orthonormal basis for S

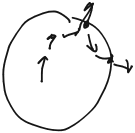
$P_S(y) = VV^T y$

Projected Gradient Descent

For any convex set let $P_{\mathcal{S}}(\cdot)$ denote the projection function onto $\mathcal{S}$.

- $P_{\mathcal{S}}(\vec{y}) = \arg\min_{\vec{\theta} \in \mathcal{S}} \|\vec{\theta} - \vec{y}\|_2$.
- For $\mathcal{S} = \{\vec{\theta} \in \mathbb{R}^d : \|\vec{\theta}\|_2 \leq 1\}$ what is $P_{\mathcal{S}}(\vec{y})$?
- For $\mathcal{S}$ being a k dimensional subspace of $\mathbb{R}^d$, what is $P_{\mathcal{S}}(\vec{y})$?

Projected Gradient Descent



- Choose some initialization $\vec{\theta}_1$ and set $\eta = \frac{R}{G\sqrt{t}}$.
- For $i = 1, \ldots, t-1$
  - $\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta \cdot \vec{\nabla} f(\vec{\theta}_i)$
  - $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.
- Return $\hat{\theta} = \arg\min_{\vec{\theta}_i} f(\vec{\theta}_i)$.

Convex Projections

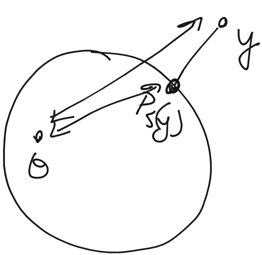
Projected gradient descent can be analyzed identically to gradient descent!

Convex Projections

Projected gradient descent can be analyzed identically to gradient descent!

Theorem – Projection to a convex set: For any convex set $\mathcal{S} \subseteq \mathbb{R}^d$, $\vec{y} \in \mathbb{R}^d$, and $\vec{\theta} \in \mathcal{S}$,

$$\|P_{\mathcal{S}}(\vec{y}) - \vec{\theta}\|_2 \leq \|\vec{y} - \vec{\theta}\|_2.$$



S subspace

$b \in S$

$\|b - W^T x\| \leq \|b - x\|$

$\downarrow$

$\|b - W^T x\| + \|(I - W^T) x\|$

Projected Gradient Descent Analysis

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

Projected Gradient Descent Analysis

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

Recall: $\vec{\theta}^{(out)}_{i+1} = \vec{\theta}_i - \eta \cdot \vec{\nabla} f(\vec{\theta}_i)$ and $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}^{(out)}_{i+1})$.

Projected Gradient Descent Analysis

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

Recall: $\underline{\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta} \cdot \vec{\nabla} f(\vec{\theta}_i)$ and $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \theta_*\|_2^2 - \|\vec{\theta}_{i+1}^{(out)} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

Projected Gradient Descent Analysis

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

Recall: $\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta \cdot \vec{\nabla} f(\vec{\theta}_i)$ and $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.

$\| \theta_{i+1}^{out} - \theta_* \|_2^2 \geq \| \theta_{i+1} - \theta_* \|_2$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \theta_*\|_2^2 - \|\vec{\theta}_{i+1}^{(out)} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

Step 1.a: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

Theorem – Projected GD: For convex G-Lipschitz function f, and convex set $\mathcal{S}$, Projected GD run with $t \geq \frac{R^2 G^2}{\epsilon^2}$ iterations, $\eta = \frac{R}{G\sqrt{t}}$, and starting point within radius R of $\vec{\theta}_*$, outputs $\hat{\theta}$ satisfying:

$$f(\hat{\theta}) \leq f(\vec{\theta}_*) + \epsilon = \min_{\vec{\theta} \in \mathcal{S}} f(\vec{\theta}) + \epsilon$$

$\nabla f(x) = c$

Recall: $\vec{\theta}_{i+1}^{(out)} = \vec{\theta}_i - \eta \cdot \vec{\nabla} f(\vec{\theta}_i)$ and $\vec{\theta}_{i+1} = P_{\mathcal{S}}(\vec{\theta}_{i+1}^{(out)})$.

$f(x) = x^T c$

Step 1: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \theta_*\|_2^2 - \|\vec{\theta}_{i+1}^{(out)} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

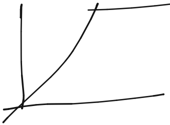
$\min_{\|x\|_2 \leq 1} x^T c$

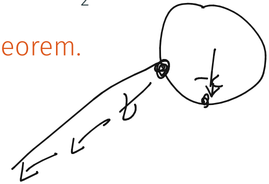
Step 1.a: For all i, $f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{\|\vec{\theta}_i - \vec{\theta}_*\|_2^2 - \|\vec{\theta}_{i+1} - \vec{\theta}_*\|_2^2}{2\eta} + \frac{\eta G^2}{2}$.

$\frac{-c}{\|c\|_2}$

Step 2: $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}_i) - f(\vec{\theta}_*) \leq \frac{R^2}{2\eta \cdot t} + \frac{\eta G^2}{2}$ $\implies$ Theorem.

$\leq \epsilon$





$S : \{x : \|x\|_2 \le 1\}$

$$f(x) = x^T c$$

$$\forall\, \theta_1, \theta_2, \lambda \quad f((1-\lambda)\theta_1 + \lambda\theta_2) \le (1-\lambda) f(\theta_1) + \lambda f(\theta_2)$$

$$\Downarrow$$

$$[(1-\lambda)\theta_1 + \lambda\,\theta_2]^T c$$

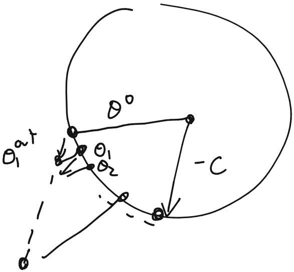
$$= (1-\lambda)\theta_1^T c + \lambda\,\theta_2^T c = (1-\lambda) f(\theta_1) + \lambda f(\theta_2)$$

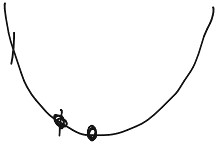
Course Review

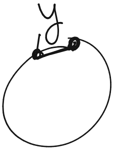
$$\theta_0 - \underline{t} \cdot \eta c$$

$$\approx -t\eta c$$

$$P_S(-t\eta c) = -c$$







Randomized Methods

Randomization as a computational resource for massive datasets.

Randomized Methods

Randomization as a computational resource for massive datasets.

- Focus on problems that are easy on small datasets but hard at massive scale – set size estimation, load balancing, distinct elements counting (MinHash), checking set membership (Bloom Filters), frequent items counting (Count-min sketch), near neighbor search (locality sensitive hashing).

Randomized Methods

Randomization as a computational resource for massive datasets.

- Focus on problems that are easy on small datasets but hard at massive scale – set size estimation, load balancing, distinct elements counting (MinHash), checking set membership (Bloom Filters), frequent items counting (Count-min sketch), near neighbor search (locality sensitive hashing).
- Just the tip of the iceberg on randomized streaming/sketching/hashing algorithms. Check out 614 if you want to learn more.

Randomized Methods

Randomization as a computational resource for massive datasets.

- Focus on problems that are easy on small datasets but hard at massive scale – set size estimation, load balancing, distinct elements counting (MinHash), checking set membership (Bloom Filters), frequent items counting (Count-min sketch), near neighbor search (locality sensitive hashing).
- Just the tip of the iceberg on randomized streaming/sketching/hashing algorithms. Check out 614 if you want to learn more.
- In the process covered probability/statistics tools that are very useful beyond algorithm design: concentration inequalities, higher moment bounds, law of large numbers, central limit theorem, linearity of expectation and variance, union bound, median as a robust estimator.

Dimensionality Reduction

Methods for working with (compressing) high-dimensional data

Dimensionality Reduction

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.

Dimensionality Reduction

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.

Dimensionality Reduction

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.
- Low-rank approximation of similarity matrices and entity embeddings (e.g., LSA, word2vec, DeepWalk).

Dimensionality Reduction

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.
- Low-rank approximation of similarity matrices and entity embeddings (e.g., LSA, word2vec, DeepWalk).
- Spectral graph theory – nonlinear dimension reduction and spectral clustering for community detection.

Dimensionality Reduction

Methods for working with (compressing) high-dimensional data

- Started with randomized dimensionality reduction and the JL lemma: compression from *any* d-dimensions to $O(\log n/\epsilon^2)$ dimensions while preserving pairwise distances.
- Dimensionality reduction via low-rank approximation and optimal solution with PCA/eigendecomposition/SVD.
- Low-rank approximation of similarity matrices and entity embeddings (e.g., LSA, word2vec, DeepWalk).
- Spectral graph theory – nonlinear dimension reduction and spectral clustering for community detection.
- In the process covered linear algebraic tools that are very broadly useful in ML and data science: eigendecomposition, singular value decomposition, projection, norm transformations.

Continuous Optimization

Foundations of continuous optimization and gradient descent.

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.

Continuous Optimization

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.
- How to analyze gradient descent in a simple setting (convex Lipschitz functions).

Continuous Optimization

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.
- How to analyze gradient descent in a simple setting (convex Lipschitz functions).
- Simple extension to projected gradient descent for optimization over a convex constraint set.

Continuous Optimization

Foundations of continuous optimization and gradient descent.

- Foundational concepts like convexity, convex sets, Lipschitzness, directional derivative/gradient.
- How to analyze gradient descent in a simple setting (convex Lipschitz functions).
- Simple extension to projected gradient descent for optimization over a convex constraint set.
- Lots that we didn't cover: online and stochastic gradient descent, accelerated methods, adaptive methods, second order methods (quasi-Newton methods), practical considerations. Gave mathematical tools to understand these methods.

Thanks for a great semester!

Final Exam Questions/Review

$\lambda_1 \approx \lambda_2$ $\qquad x = c_1 v_1 \dots c_d v_d$

$$Ax = \lambda_1 c_1 v_1 \dots \lambda_d c_d v_d$$

- don't fully understand error analysis at end

$\|\hat{v} - v_1\| \le \epsilon$ $\qquad O\left(\frac{\log(d/\epsilon)}{\gamma}\right)$

$$A(A(Ax)) = \underline{\underline{A^t}} x = \lambda_1^t c_1 v_1 \dots \lambda_d^t c_d v_d$$

$x \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$ $\qquad \boxed{3d^2}$ $\qquad A^3x \quad AAx = A^2x \quad Ax$

$(\underline{A^3 - 2A^2} + A)x \qquad d^2 \quad d^2 \quad d^2$

Πx $\quad$ $\underline{V^T x}$

JL $\quad$ PCA

$f(x) = x^T c \qquad \nabla f(x) = c$ $\quad \mathbb{R}^d$

$f(x) = \sum_{i=1}^{d} x(i)\, c(i)$

$\frac{\partial f(x)}{\partial x_i} = c(i)$

$$\begin{bmatrix} \frac{\partial f x}{\partial x_1} \\ \vdots \\ \frac{\partial f x}{\partial x_n} \end{bmatrix} = \begin{bmatrix} c(1) \\ \vdots \\ c(n) \end{bmatrix} = \vec{c}$$

Final Exam Questions/Review

$f(x) = x^T A x$

$\langle x, Ax \rangle$

$x \cdot A \cdot x$

$x^2 A$

$2xA$

$\nabla f(x) = 2Ax$

$f(x) = \sum_{i=1}^{d} x(i)[Ax](i)$

$\sum_{i=1}^{d} x(i) \sum_{j=1}^{d} A_{ij} x(j)$

$\sum_{j \neq i} x(i)x(j)A_{ij} + x(j)x(i)A_{ji}$

$+ x(i)^2 A_{ii}$

$\sum_{j \neq i} 2x_j A_{ij} + 2x(i) A_{ii}$

$= 2[Ax](i)$

$\left[\Sigma \quad A \quad J \right] \left[x \right] = \left[0 \right]$

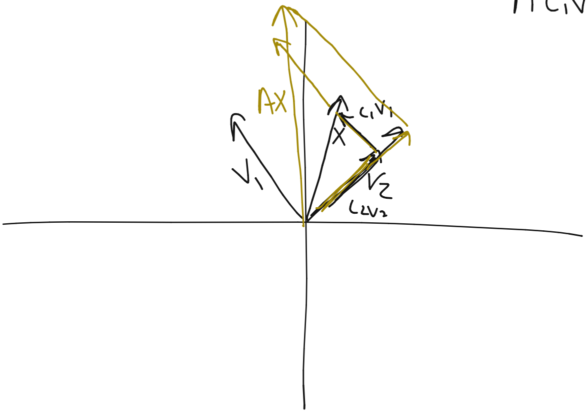
$\dfrac{\partial f(x)}{\partial x(i)}$

Final Exam Questions/Review

$$\begin{array}{cc} & 2 \\ 2 & \begin{bmatrix} \ A \ \end{bmatrix} \end{array} \begin{bmatrix} x \end{bmatrix}$$

$$Ax = A(c_1v_1 + c_2v_2)$$

$$Ac_1v_1 + Ac_2v_2$$



Final Exam Questions/Review