

C477: First Order Methods

Ruth Misener

r.misener@imperial.ac.uk

Panos Parpas

p.parpas@imperial.ac.uk

Computational Optimisation Group
Department of Computing

Imperial College
London



26 Oct 2018

Outline

- **Topics**

- ▶ Overview of Gradient-Based Methods
- ▶ Level Sets
- ▶ Descent Directions
- ▶ Steepest Descent Algorithm
- ▶ Implementing First-Order methods
- ▶ Convergence Theory

- **Example**

- ▶ Recognising hand-written digits in the MNIST dataset

- **Reading**

- ▶ Chapter 8 (Gradient Methods), Chong & Zak, Third Edition, Chapter 4 (The Gradient Method), Beck.

- **Acknowledgements**

- ▶ Parts of these slides were originally developed by Benoit Chachuat and Panos Parpas. L^AT_EX design by Miten Mistry. Mistakes by Ruth Misener.

Introduction

C477 uses these terms interchangeably

- First-Order Methods
- Gradient-Based Methods
- Descent Algorithms



First-Order Methods in C477 vs The First Order in *Star Wars*

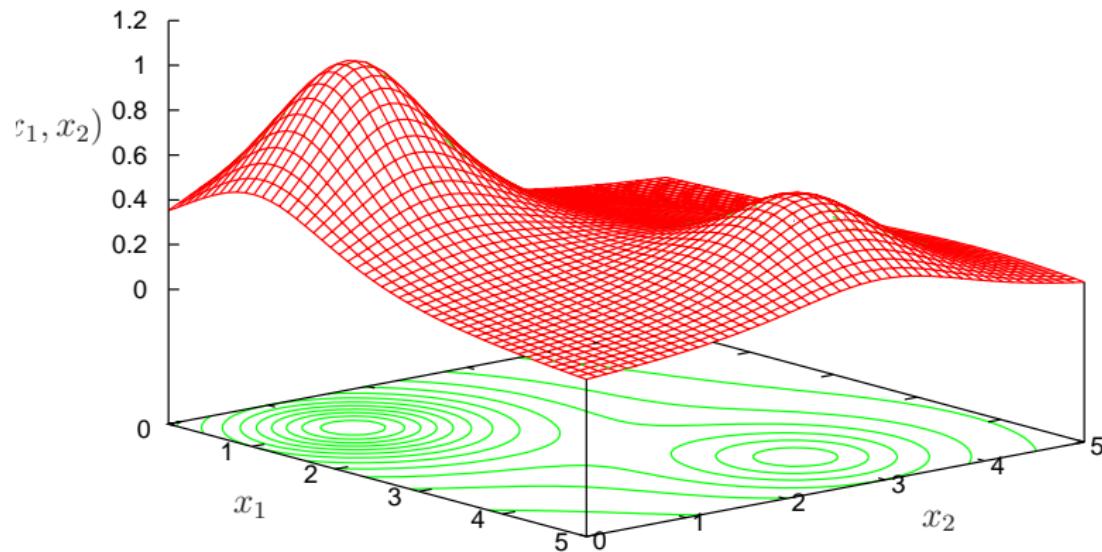
In the movie *The Force Awakens*, the First Order is a fictional military power. We argue that the power of the First Order may be related to using strong optimisation methods.

http://starwars.wikia.com/wiki/First_Order

Preview of First-Order, Gradient-Based Methods

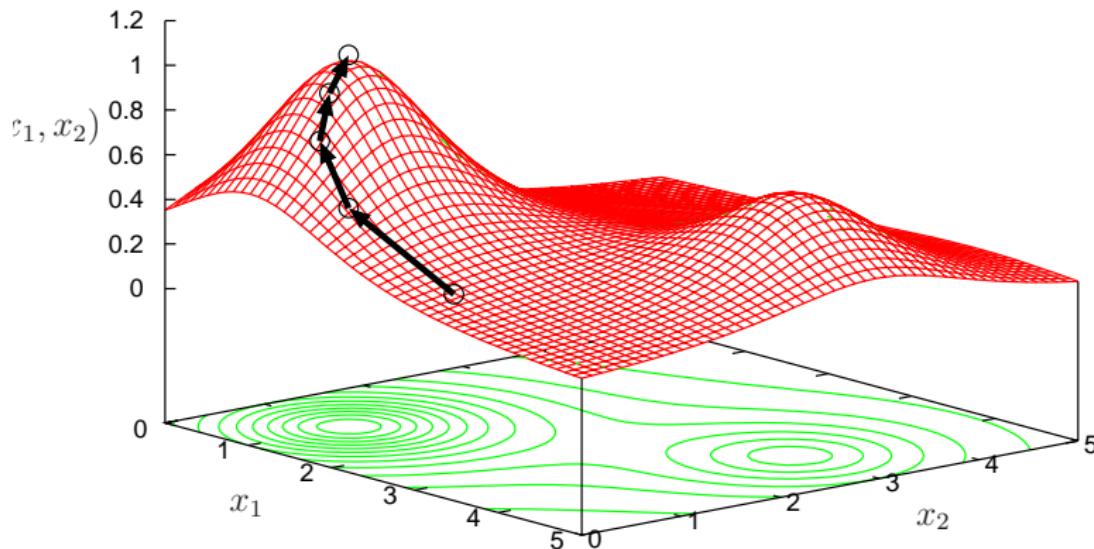
Consider the maximisation problem:

$$\max_{0 \leq x_1, x_2 \leq 5} f(x_1, x_2) \triangleq \frac{1}{1 + (x_1 - 1)^2 + (x_2 - 1)^2} + \frac{0.5}{1 + (x_1 - 4)^2 + (x_2 - 3)^2}$$



Preview of First-Order, Gradient-Based Methods

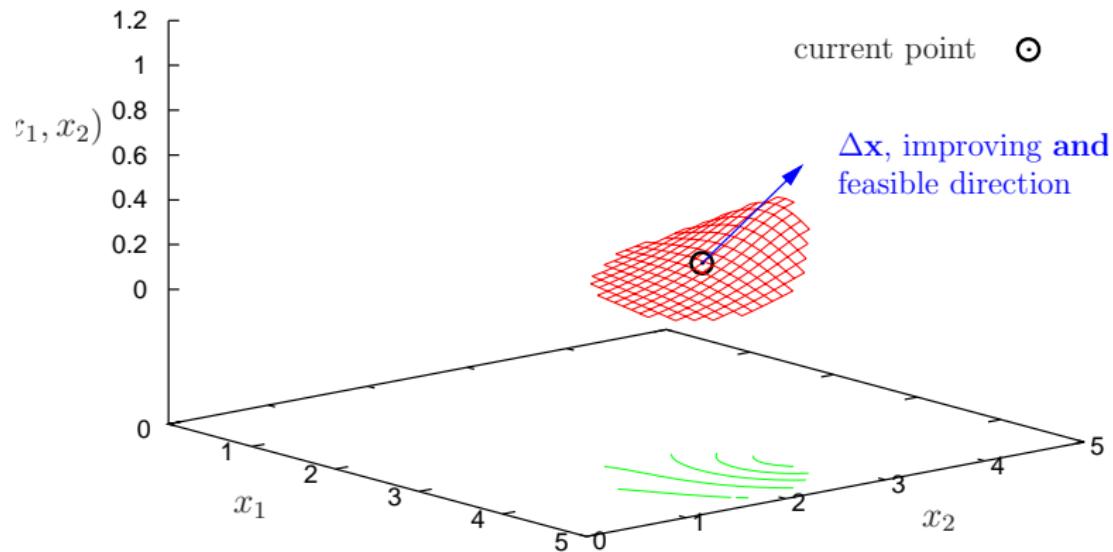
First-order methods are numerical algorithms starting at a **feasible** solution to an optimisation model. Using function and gradient evaluations, they advance along a search path of feasible points with **improving** function values.



Preview of First-Order, Gradient-Based Methods

The Dilemma: Typically, only some **local information** is known about the objective function typically at a current point $x^k = (x_1^k, x_2^k)$!

→ May lead to a **local optimum** only!



Overview of First-Order, Gradient-Based Methods

Unconstrained problem,

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}).$$

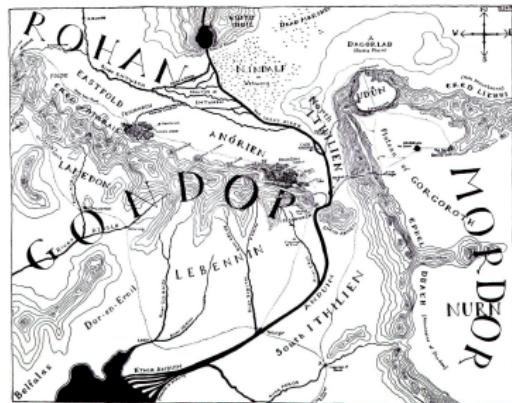
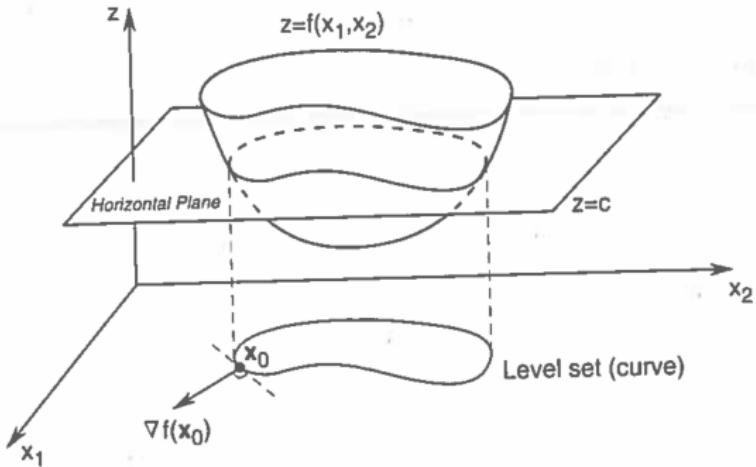
Idea: Generate a sequence of points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}, \dots$ such that:

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$$

We seek convergence in the sense:

- $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^*$;
- $\lim_{k \rightarrow \infty} f(\mathbf{x}^{(k)}) = f^*$;
- \mathbf{x}^* is a local minimum, e.g., satisfies optimality conditions;
- Sometimes more precise statements can be made.

Reminder: Level Set



Level Set

A level set of a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the set of points x satisfying $f(x) = c$ for some constant $c \in \mathbb{R}$.

Reminder: Directional Derivative

Definition (Directional Derivative)

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a real valued function, and let $\mathbf{d} \in \mathbb{R}^n \setminus \mathbf{0}$. The directional derivative of f in the direction \mathbf{d} is defined:

$$\frac{\partial f}{\partial \mathbf{d}}(\mathbf{x}) = \lim_{\alpha \rightarrow 0} \frac{f(\mathbf{x} + \alpha \mathbf{d}) - f(\mathbf{x})}{\alpha} = \mathbf{d}^\top \nabla f(\mathbf{x})$$

Reminder: The gradient of f is denoted:

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^\top.$$

The Jacobian of f is denoted by Df and $\nabla f(\mathbf{x}) = Df^\top$. The rate of change in f at \mathbf{x} in the i^{th} direction is $\partial f / \partial x_i(\mathbf{x})$.

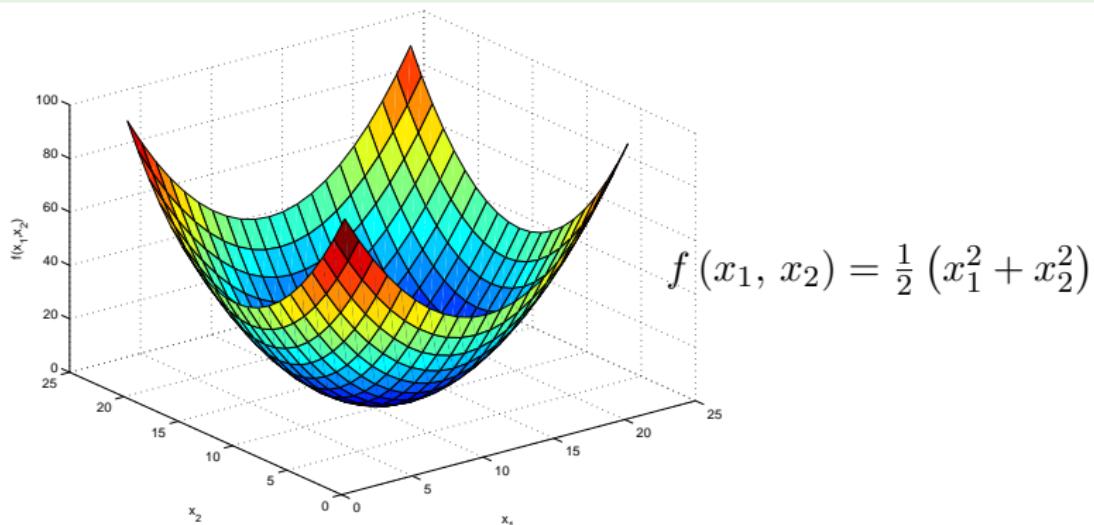
- Directional derivatives allow all the dimensions to vary at the same time and by different amounts. For consistency: $\|\mathbf{d}\|_2 = 1$

Example

Find the directional derivative of f :

$$f(\mathbf{x}) = \frac{1}{2} (x_1^2 + x_2^2),$$

at the point $\mathbf{x} = [-4, 5]^\top$ in the direction $\mathbf{d} = \left[\frac{-2}{\sqrt{13}}, \frac{3}{\sqrt{13}} \right]^\top$

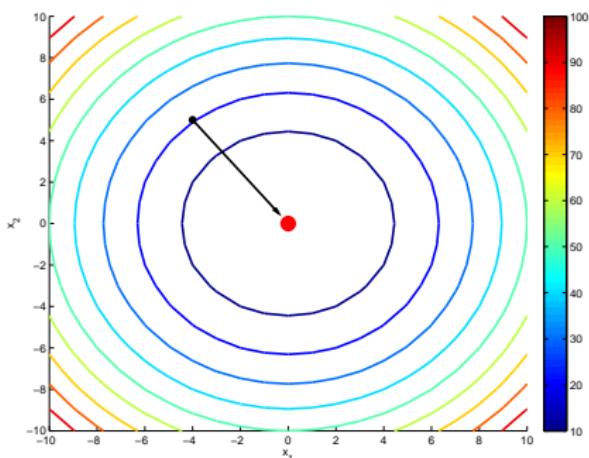


Example

Find the directional derivative of f

$$f(\mathbf{x}) = \frac{1}{2}(x_1^2 + x_2^2)$$

at the point $\mathbf{x} = [-4, 5]^\top$ in the direction $\mathbf{d} = \left[\frac{4}{\sqrt{41}}, \frac{-5}{\sqrt{41}} \right]^\top$.

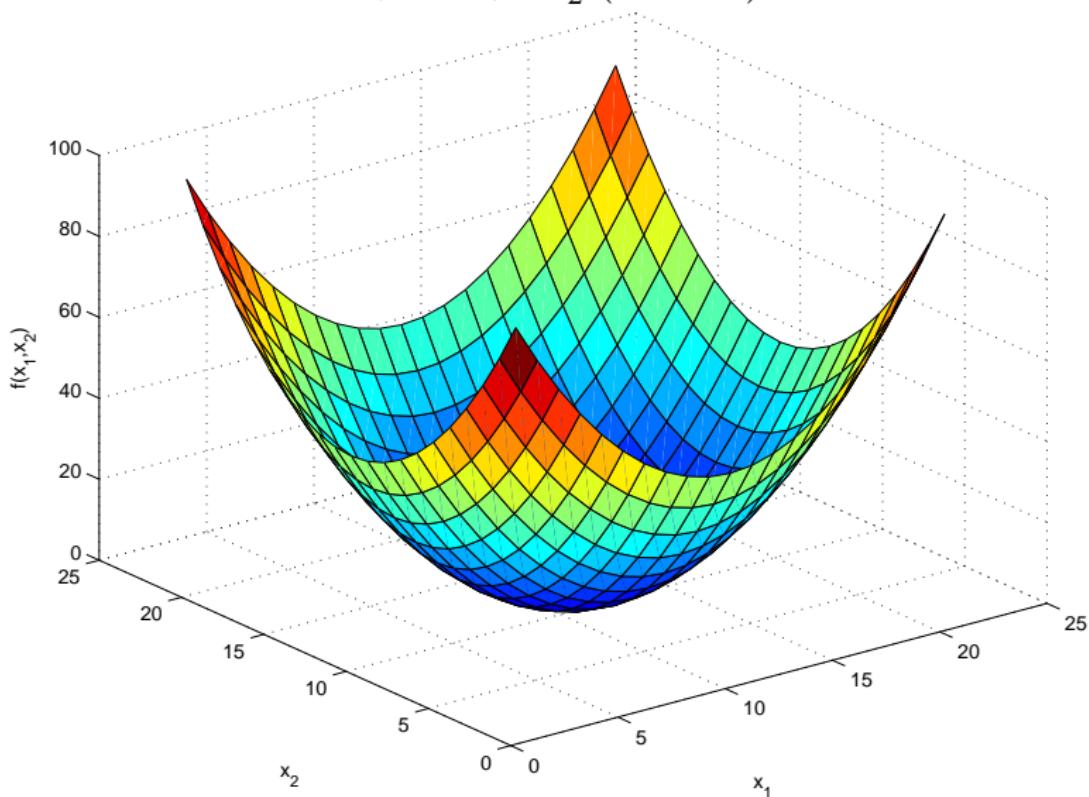


Sanity Check

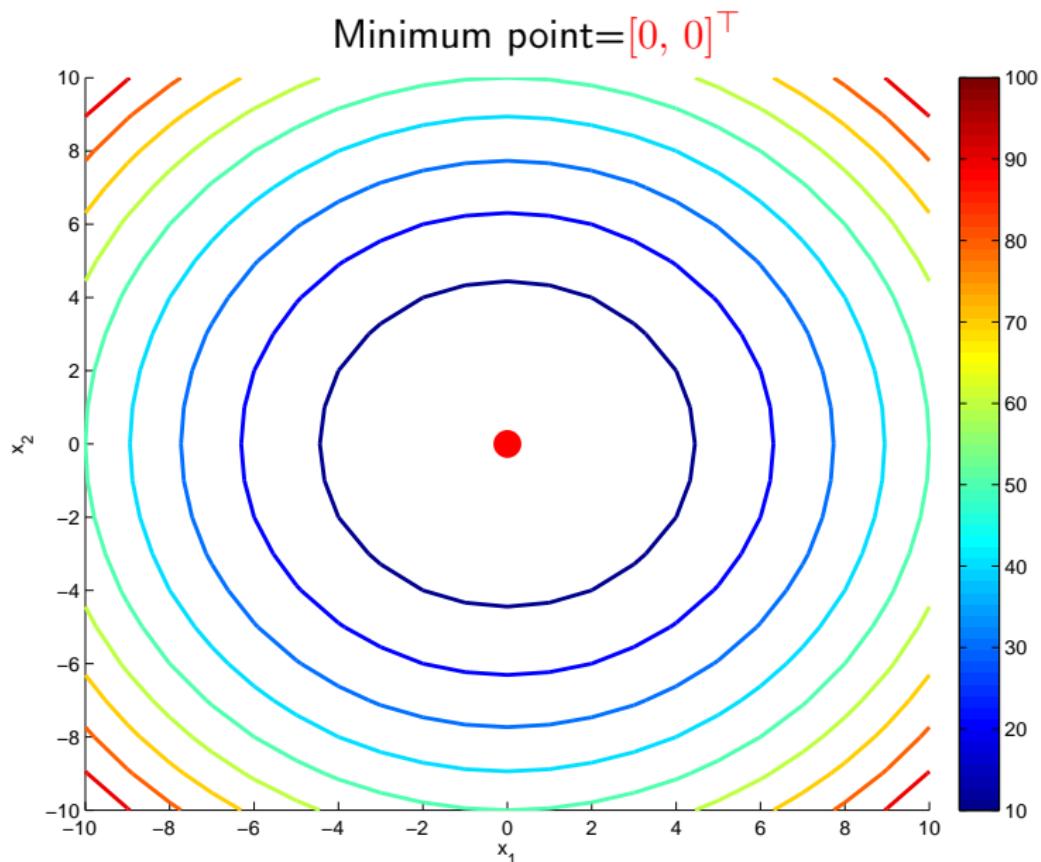
Does the function increase or decrease at the point \mathbf{x} in the direction \mathbf{d} ?

Example

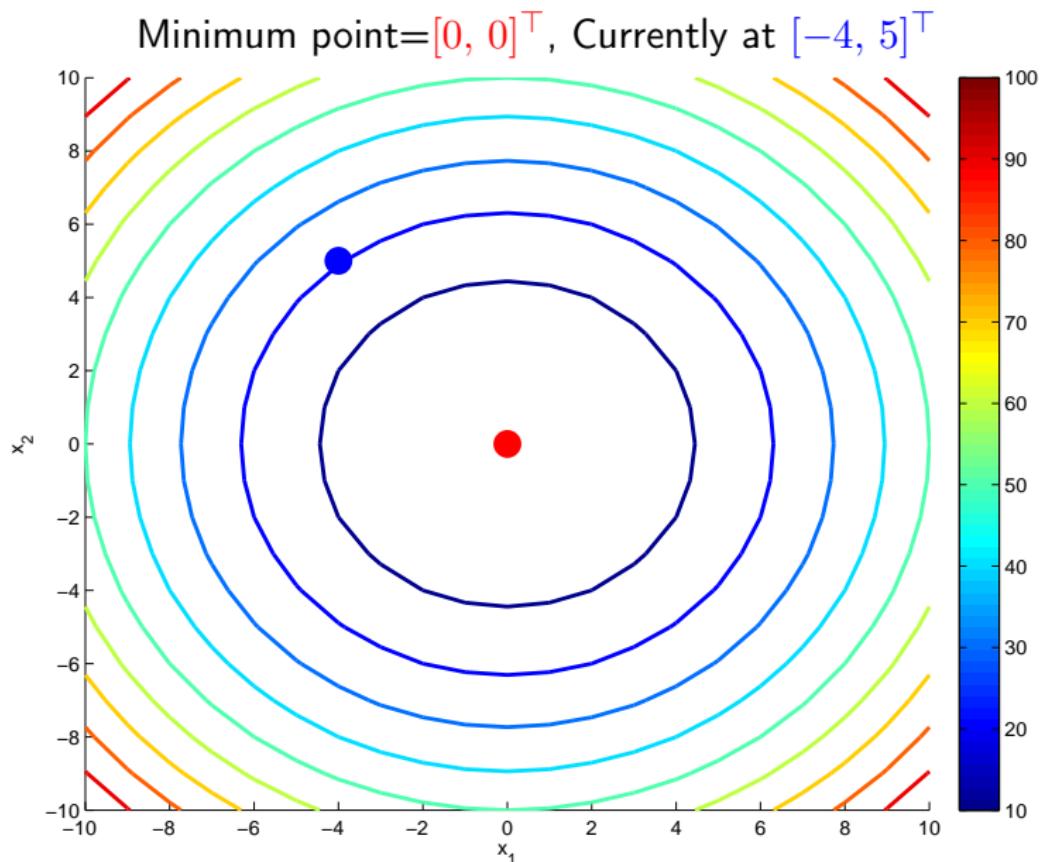
$$f(x_1, x_2) = \frac{1}{2} (x_1^2 + x_2^2)$$



Example

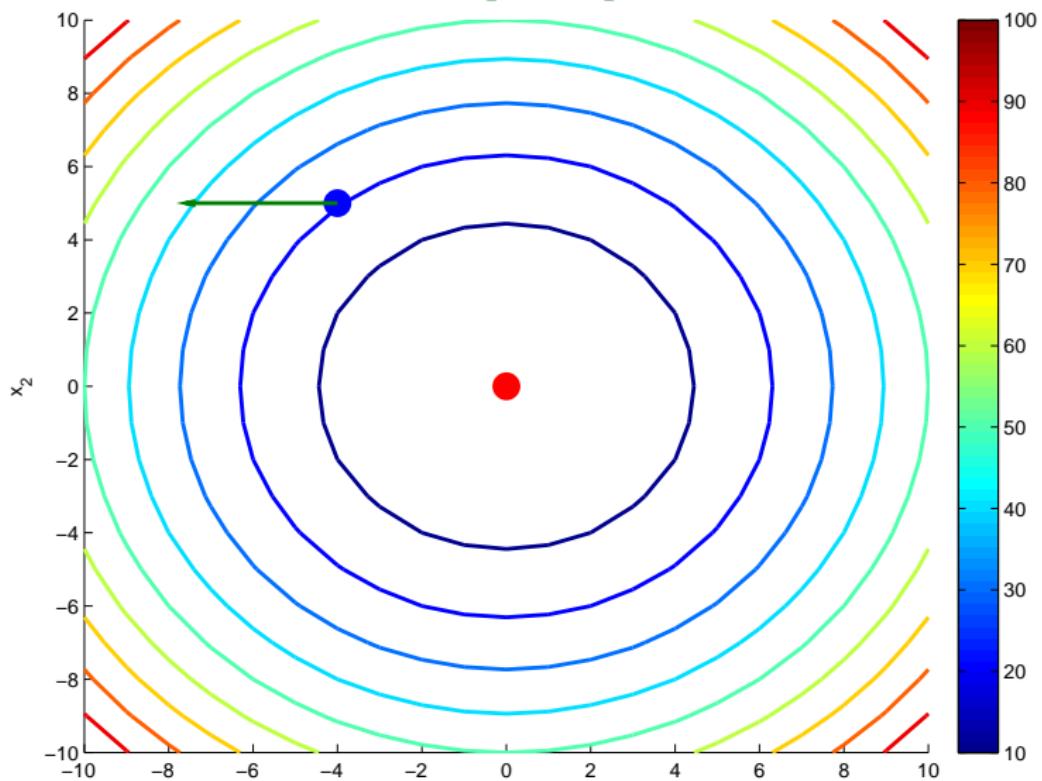


Example



Example

$$\langle \nabla f(\mathbf{x}), \mathbf{1}_1 \rangle = \left[\frac{\partial f}{\partial x_1}, 0 \right]^\top \text{ at } [-4, 5]^\top$$



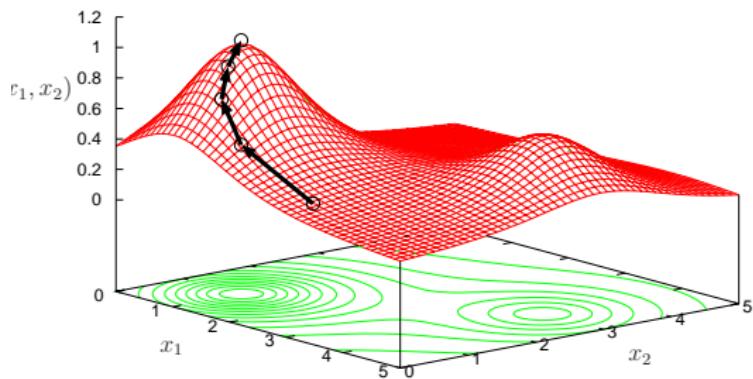
Basic Ingredients of Descent Algorithms

- ➊ Given a point $\mathbf{x}^{(k)}$.
- ➋ Derive a descent direction $\mathbf{d}^{(k)} \in \mathbb{R}^n$, i.e.,

$$\nabla f\left(\mathbf{x}^{(k)}\right)^\top \mathbf{d}^{(k)} < 0.$$

- ➌ Decide on a step-size α_k .
- ➍ Transition to the next point,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}.$$



Descent Directions

Definition: Descent Direction (\mathbf{d})

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a continuously differentiable function over \mathbb{R}^n . A vector $\mathbf{d} \neq \mathbf{0}$ is called a **descent direction** of f at \mathbf{x} if the directional derivative of f at \mathbf{x} in the direction \mathbf{d} is negative:

$$\nabla f(\mathbf{x})^\top \mathbf{d} < 0.$$

Sanity Check

What does the descent condition mean?

Step Size: $\alpha_k > 0$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

Sanity Check

Why do we need the step-size computation?

Steepest Ascent

For a small displacement, the function f increases more in the direction of the gradient than in any other direction. By definition:

$$\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle = \nabla f(\mathbf{x})^\top \mathbf{d} \text{ with } \|\mathbf{d}\|_2 = 1$$

is the rate of increase in f along the direction \mathbf{d} . Using the Cauchy-Schwarz inequality,

$$\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle \leq \|\nabla f(\mathbf{x})\|_2 \|\mathbf{d}\|_2 = \|\nabla f(\mathbf{x})\|_2$$

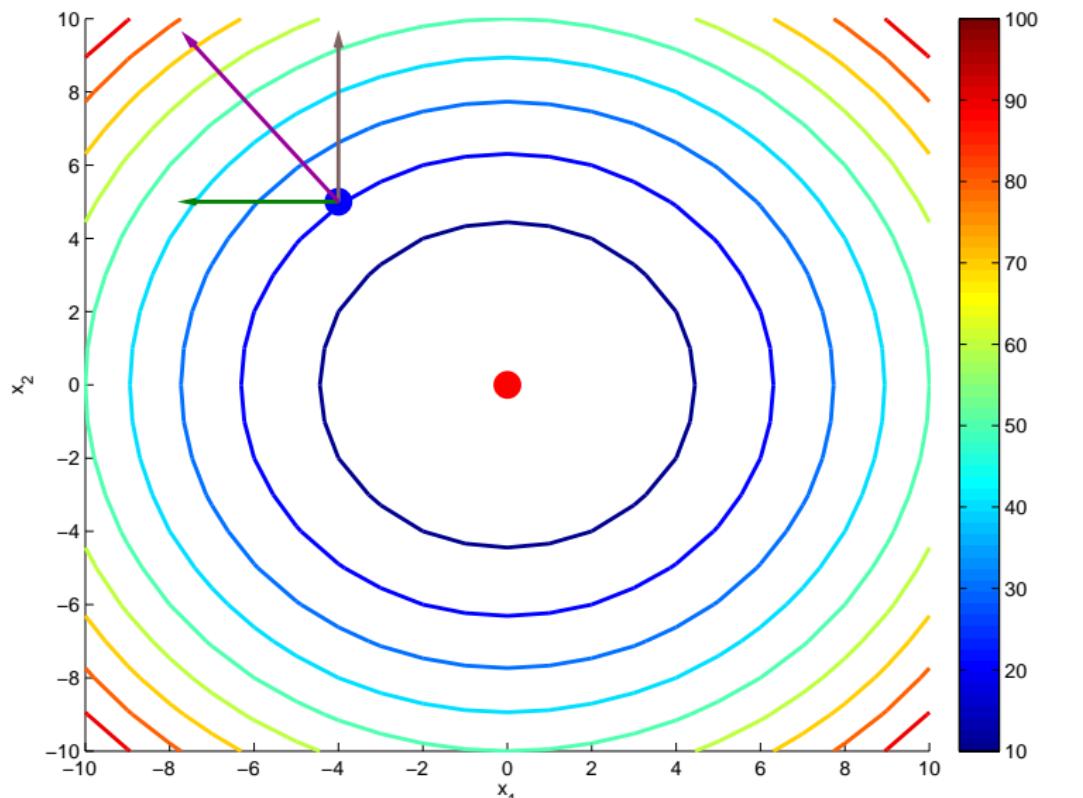
Therefore the rate of increase cannot be greater than $\|\nabla f(\mathbf{x})\|_2$. Then taking $\mathbf{d} = \nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\|_2$:

$$\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle = \left\langle \nabla f(\mathbf{x}), \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|_2} \right\rangle = \frac{\|\nabla f(\mathbf{x})\|_2^2}{\|\nabla f(\mathbf{x})\|_2} = \|\nabla f(\mathbf{x})\|_2$$

and $\mathbf{d} = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|_2}$ achieves the upper bound.

Example

$$\langle \nabla f(\mathbf{x}), \mathbb{1}_1 \rangle = \left[\frac{\partial f}{\partial x_1}, 0 \right]^\top, \quad \langle \nabla f(\mathbf{x}), \mathbb{1}_2 \rangle = \left[0, \frac{\partial f}{\partial x_2} \right]^\top \quad \nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right]^\top$$



General Justification of Descent Condition

- ① Suppose that $\mathbf{d}^{(k)}$ is a descent direction $\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{d}^{(k)} \rangle < 0$;
- ② Compute a point such that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}$;
- ③ Then if α is small enough we have:

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}).$$

Using Taylor's theorem we obtain,

$$f(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k)}) + \alpha \nabla f(\mathbf{x}^{(k)})^\top \mathbf{d}^{(k)} + r(\alpha \mathbf{d}^{(k)}).$$

Note that $\nabla f(\mathbf{x}^{(k)}) \neq 0$, \mathbf{d} is a descent direction so if α is small enough we obtain,

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}).$$

Descent Directions – Summary

- ① Given a point $\mathbf{x}^{(k)}$.
- ② Derive a **descent** direction $\mathbf{d}^{(k)} \in \mathbb{R}^n$, i.e.,

$$\nabla f(\mathbf{x}^{(k)})^\top \mathbf{d}^{(k)} < 0.$$

- ③ Decide on a step-size α_k .
- ④ Transition to the next point,

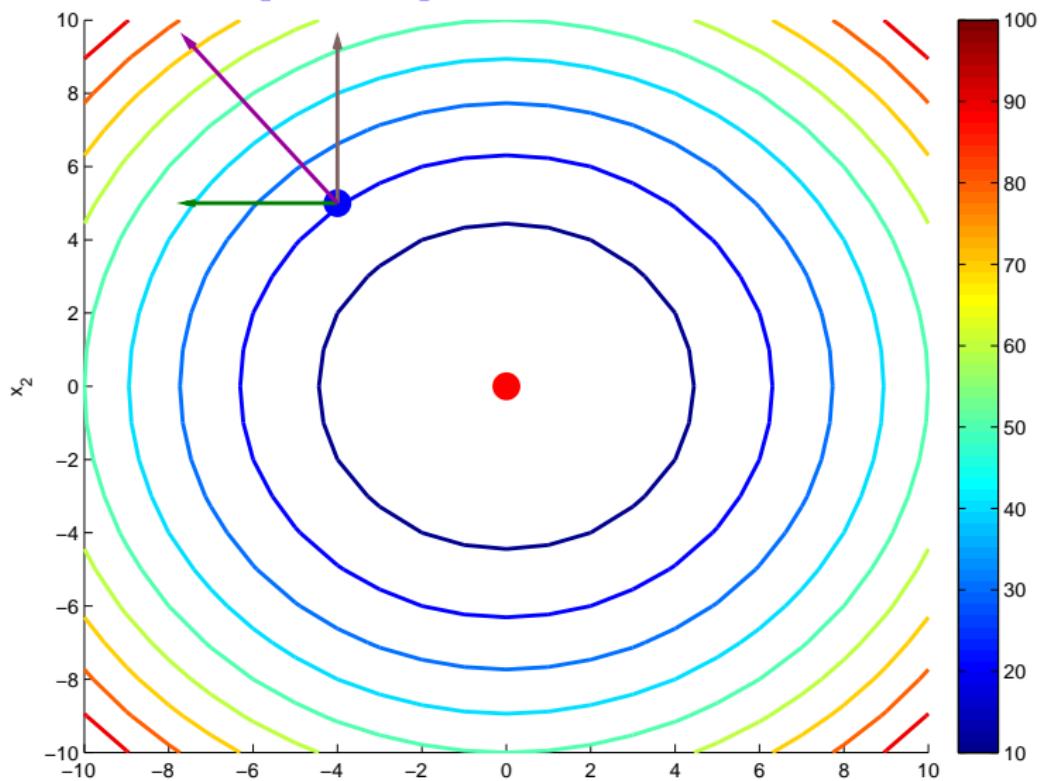
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}.$$

Sanity Check

What does $\nabla f(\mathbf{x}^{(k)})^\top \mathbf{d}^{(k)} < 0$ mean? How do we chose a good α ?

How to select the descent direction?

$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right]^\top$: Direction of greatest increase



How to select the step-size

If $\mathbf{d}^{(k)}$ is a descent direction, e.g., $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) / \|\nabla f(\mathbf{x}^{(k)})\|_2$, and:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

We know that if α_k is small enough then, $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$, i.e., the algorithm **improves** the current point $\mathbf{x}^{(k)}$.

Exact Step Size Strategy

Select the step size with the maximum improvement, i.e.,

$$\alpha_k \in \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$

Sanity Check

How many dimensions in the optimisation problem calculating the exact step size? If f is convex, will it have any special properties?

Steepest Descent Algorithm

General Descent Algorithm

- ① Given a point $\mathbf{x}^{(k)}$.
- ② Derive a **descent** direction $\mathbf{d}^{(k)} \in \mathbb{R}^n$, i.e.,

$$\nabla f\left(\mathbf{x}^{(k)}\right)^{\top} \mathbf{d}^{(k)} < 0.$$

- ③ Decide on a step-size α_k .
- ④ Transition to the next point,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

Steepest Descent Algorithm

- ① Given a point $\mathbf{x}^{(k)}$.
- ② Compute the gradient at $\mathbf{x}^{(k)}$

$$\mathbf{d}^{(k)} = -\nabla f\left(\mathbf{x}^{(k)}\right)$$

($\mathbf{d}^{(k)}$ is a descent direction)

- ③ Decide on a step-size α_k ,

$$\alpha_k \in \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$

- ④ Transition to the next point,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

Implementation Issues [1/2]

Convergence Criteria

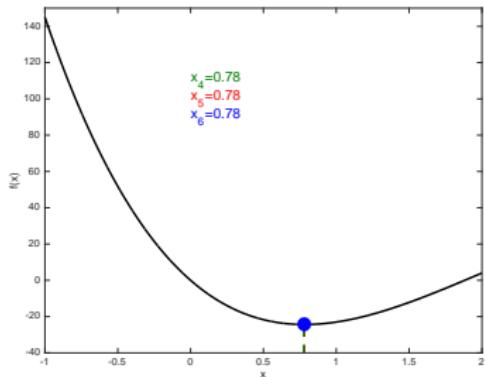
- ① Stop when FONC is satisfied:

$$\nabla f(\mathbf{x}^{(k)}) = \mathbf{0}.$$

- ② In practice

- $\|\nabla f(\mathbf{x}^{(k)})\|_2 < \epsilon_1;$
- $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2 < \epsilon_2;$
- $|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})| < \epsilon_3;$

where $\epsilon_1, \epsilon_2, \epsilon_3 > 0$ are tolerance parameters.



Implementation Issues [2/2]

Computing Derivatives

- ① Use exact derivatives (symbolic/manual);
- ② Finite Difference;
- ③ Automatic Differentiation.

Step-size Strategy

- ① Exact Line Search; minimise with a one dimensional method:

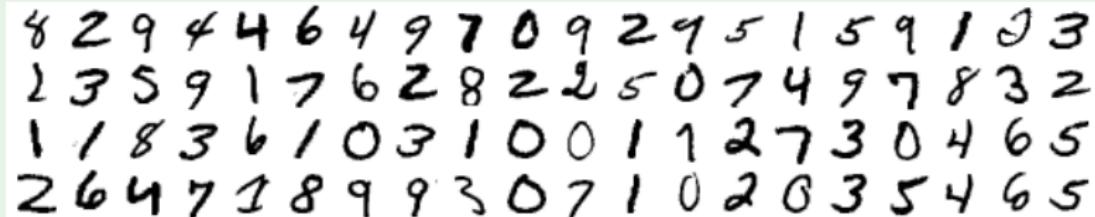
$$\alpha_k \in \arg \min_{\alpha} f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$

- ② Inexact line search
 - ▶ Percentage Test, Armijo's Rule, Curve Fitting, Backtracking, ...

Example: Sparse Logistic Regression

Challenge: How can we classify hand-written digits from images?

$$\min_{\beta_1, \dots, \beta_{10}} \sum_{i=1}^m \left(\log \sum_{k=1}^{10} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_k) - \mathbf{x}_i^\top \boldsymbol{\beta}_{y_i} \right) + \lambda \sum_{k=1}^{10} \|\boldsymbol{\beta}_k\|_1$$



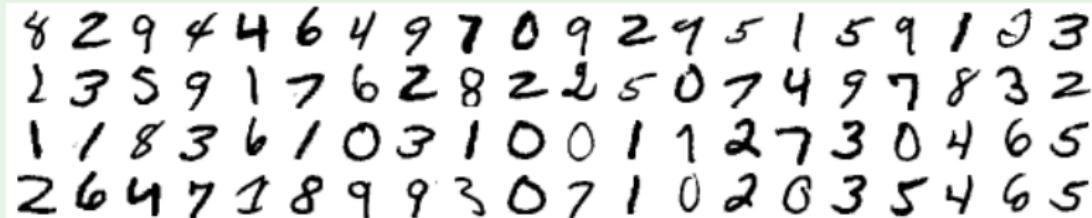
Where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{0, 1, \dots, 9\}$ are the input features and output label, respectively, for each training example, $i \in \{1, 2, \dots, m\}$. There are also feature weights, $\boldsymbol{\beta}_k \in \mathbb{R}^n$, for $k = \{0, 1, \dots, 9\}$

Typically use **first-order**, **sub**gradient-based methods, to solve this.

Example: Sparse Logistic Regression

Changing the Problem: Classify hand-written digits from images?

$$\min_{\beta_1, \dots, \beta_{10}} \sum_{i=1}^m \left(\log \sum_{k=1}^{10} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_k) - \mathbf{x}_i^\top \boldsymbol{\beta}_{y_i} \right) + \lambda \sum_{k=1}^{10} \|\boldsymbol{\beta}_k\|_2^2$$



Where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{0, 1, \dots, 9\}$ are the input features and output label, respectively, for each training example, $i \in \{1, 2, \dots, m\}$. There are also feature weights, $\boldsymbol{\beta}_k \in \mathbb{R}^n$, for $k = \{0, 1, \dots, 9\}$

Now we can use **first-order**, gradient-based methods, to solve this.

Inexact Line Search

Examples of Inexact Line Search

- ① **Percentage Test:** Similar to exact line search but stops within a fixed accuracy, e.g., $|\alpha_k - \alpha^*| < c\alpha^*$, with $0 < c < 1$, $c = 0.1$ a typical value;
- ② **Curve Fitting:** Fit a function to $f(x^{(k)} + \alpha_k d^{(k)})$ and minimise that, e.g., Newton's method with a quadratic fit;
- ③ **Armijo's Rule:** A rule to ensure that α is not too large and not too small;
- ④ **Backtracking:** A simple variation of Armijo's rule.

Backtracking Line Search

Backtracking Line Search

One way to adaptively choose the step size is to initialise parameters $0 < c_\alpha < 1$ and $0 < c_\beta < 1$. At each iteration k , start with $\alpha^{(k)} = 1$, and while:

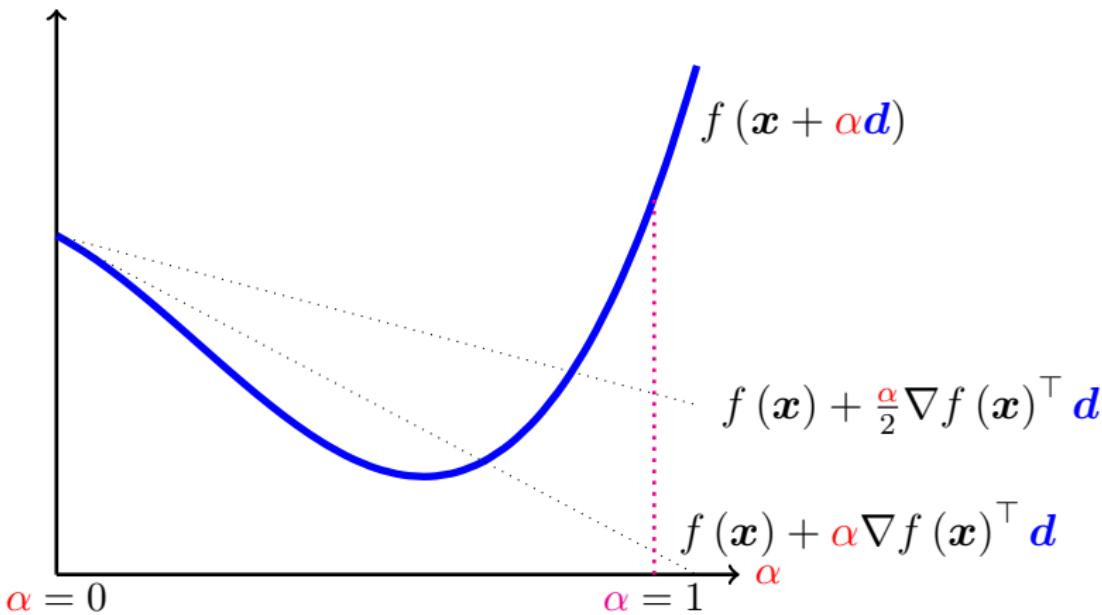
$$f\left(\mathbf{x}^{(k)} - \alpha^{(k)} \nabla f\left(\mathbf{x}^{(k)}\right)\right) > f(\mathbf{x}^{(k)}) - c_\alpha \alpha^{(k)} \|\nabla f\left(\mathbf{x}^{(k)}\right)\|_2^2$$

shrink $\alpha^{(k)} = c_\beta \alpha^{(k)}$. Else perform gradient descent update:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f\left(\mathbf{x}^{(k)}\right)$$

Simple and tends to work well in practice. A common choice is $c_\alpha = 1/2$.

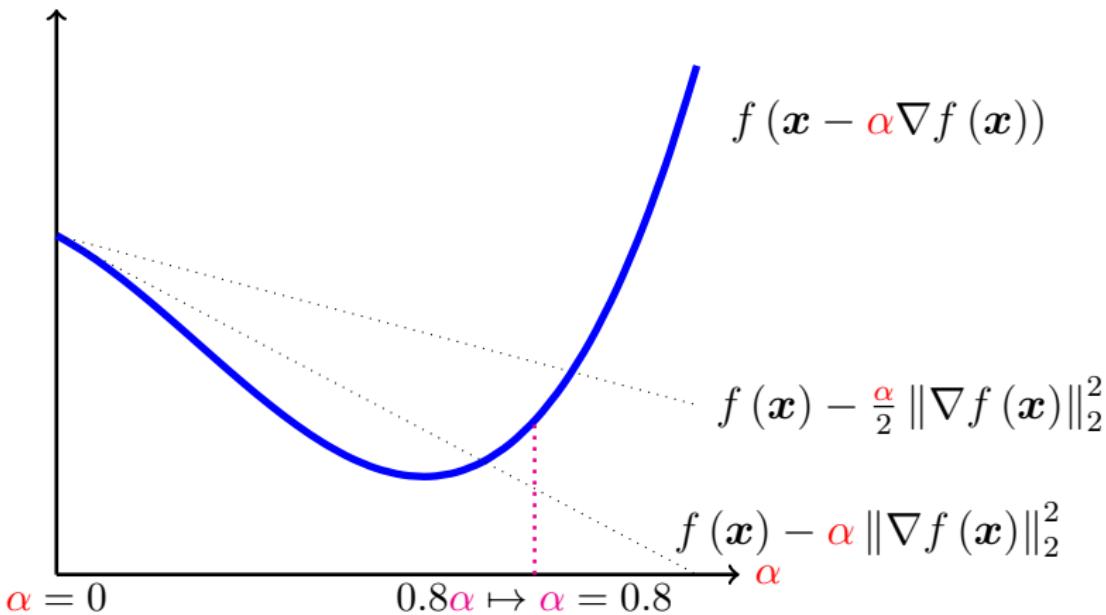
Backtracking interpretation - Steepest Descent



Sanity Check

Backtracking line search initialises parameters $0 < c_\alpha < 1$ and $0 < c_\beta < 1$, e.g. $c_\alpha = 1/2$ and $c_\beta = 0.8$. Would it be reasonable to choose $c_\alpha = 1$ for a twice continuously differentiable function?

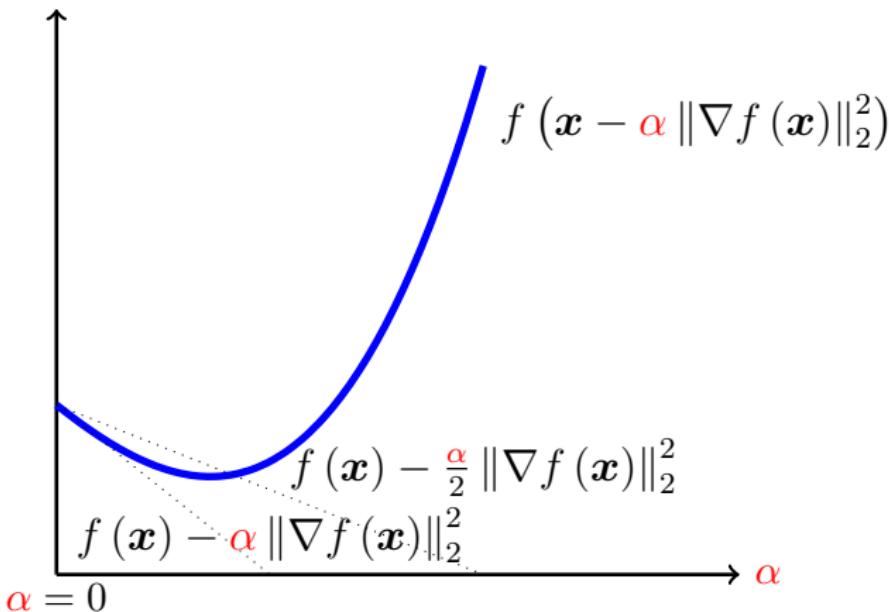
Backtracking interpretation - Steepest Descent



Sanity Check

Backtracking line search initialises parameters $0 < c_\alpha < 1$ and $0 < c_\beta < 1$, e.g. $c_\alpha = 1/2$ and $c_\beta = 0.8$. Would it be reasonable to choose $c_\alpha = 1$ for a twice continuously differentiable function?

Backtracking interpretation - Steepest Descent



Sanity Check

Backtracking line search initialises parameters $0 < c_\alpha < 1$ and $0 < c_\beta < 1$, e.g. $c_\alpha = 1/2$ and $c_\beta = 0.8$. Would it be reasonable to choose $c_\alpha = 1$ for a twice continuously differentiable function?

Convergence Theory

Some General Convergence Results

If function $f \in \mathcal{C}^2$ is strictly convex and we use an exact line search algorithm, then:

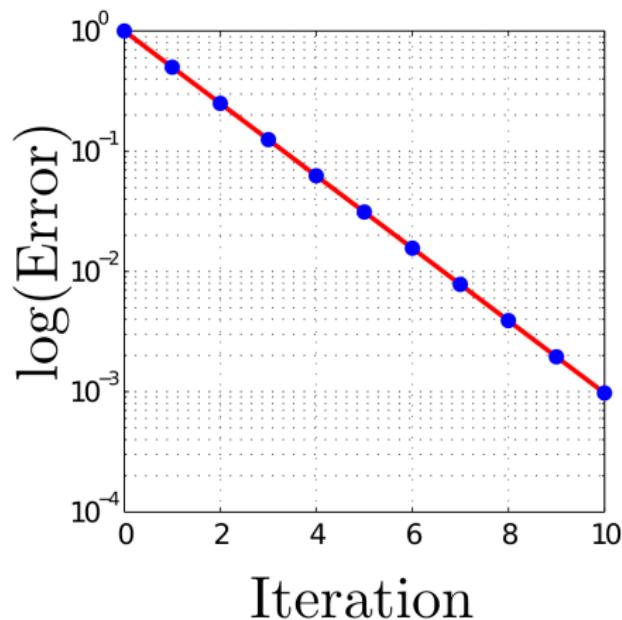
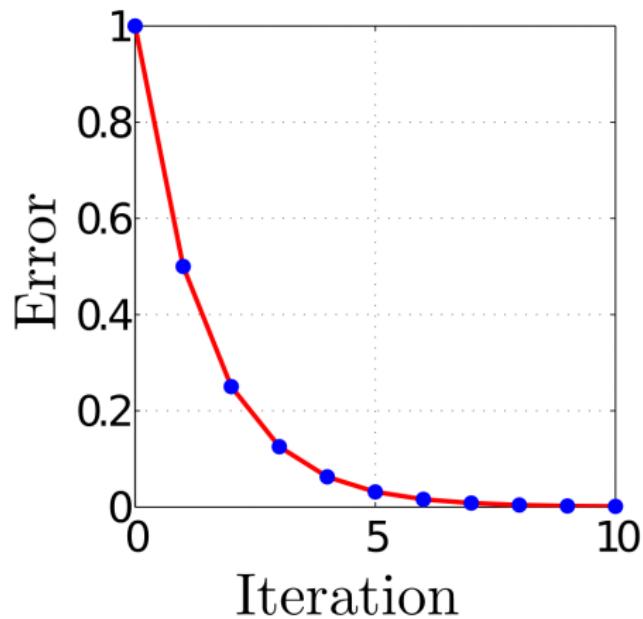
- **Convergence:** The sequence generated by the algorithm $f(\mathbf{x}^{(k)})$ converges to $f(\mathbf{x}^*)$.
- **Convergence Rate:** The algorithm converges *linearly* to \mathbf{x}^* with a rate given by,

$$f(\mathbf{x}^{(k)}) - f^* \leq \left(1 - \frac{m}{M}\right)^k \left(f(\mathbf{x}^{(0)}) - f^*\right),$$

where x_0 is the initial starting point. We define the smallest and greatest eigenvalue of the Hessian of f at \mathbf{x}^* as $m = \lambda_{\min}(\nabla^2 f(\mathbf{x}^*))$ and $M = \lambda_{\max}(\nabla^2 f(\mathbf{x}^*))$.

Convergence Rate

Linear convergence rate means converges linearly in log scale



$$f(\pmb{x}) = \tfrac{1}{2}(x_1^2 + \gamma x_2^2) \quad \gamma = 1,\; x^{(0)} = [10,\,1]^\top$$

$$f(\pmb{x}) = \tfrac{1}{2}(x_1^2 + \gamma x_2^2) \quad \gamma = 5,\; x^{(0)} = [10,\, 1]^\top$$

$$f(\pmb{x}) = \tfrac{1}{2}(x_1^2 + \gamma x_2^2) \quad \gamma = 10,\; \pmb{x}^{(0)} = [10,\, 1]^\top$$

For First-Order Algorithms



Pro:

- + Easy to implement;
- + Only requires first-order, gradient-based information;
- + Convergence results exist if an appropriate step size strategy is used;
- + Only choice for large scale problems;
- + Variations for non-differentiable problems exist; these are called sub-gradient methods and are not covered in this course.

Against First-Order Algorithms

Con:

- Convergence is slow (there are algorithms with quadratic rate, see lecture on 2nd order methods);
- Sensitive to the scaling of the problem.



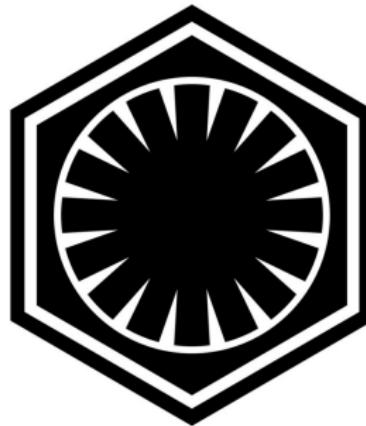
First Order Insignia

Condition Number

Definition of Condition Number

The condition number of an $n \times n$ positive definite matrix \mathbf{Q} is the ratio between the maximal and minimal eigenvalues of \mathbf{Q} :

$$\chi(\mathbf{Q}) = \frac{M}{m} = \frac{\lambda_{\max}(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})}.$$



Condition Number Example [1/3]

Quadratic Minimisation Example [Beck, 2014]

Consider the quadratic minimisation problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) \equiv \mathbf{x}^\top \mathbf{Q} \mathbf{x} \right\}$$

where $\mathbf{Q} \succ 0$. The optimal solution is obviously $\mathbf{x}^* = \mathbf{0}$. The gradient method with exact line search takes the form:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)},$$

where $\mathbf{d}^{(k)} = -2\mathbf{Q}\mathbf{x}^{(k)}$ is the negative gradient of f at $\mathbf{x}^{(k)}$ and the stepsize α_k chosen by the exact minimisation rule is (why?):

$$\alpha_k = \frac{\mathbf{d}^{(k)\top} \mathbf{d}^{(k)}}{2\mathbf{d}^{(k)\top} \mathbf{Q} \mathbf{d}^{(k)}}.$$

Condition Number Example [2/3]

Quadratic Minimisation Example (continued)

Therefore:

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &= \mathbf{x}^{(k+1)\top} \mathbf{Q} \mathbf{x}^{(k+1)} \\ &= (\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)})^\top \mathbf{Q} (\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}) \\ &= \mathbf{x}^{(k)\top} \mathbf{Q} \mathbf{x}^{(k)} + 2\alpha_k \mathbf{d}^{(k)\top} \mathbf{Q} \mathbf{x}^{(k)} + \alpha_k^2 \mathbf{d}^{(k)\top} \mathbf{Q} \mathbf{d}^{(k)} \\ &= \mathbf{x}^{(k)\top} \mathbf{Q} \mathbf{x}^{(k)} - \alpha_k \mathbf{d}^{(k)\top} \mathbf{d}^{(k)} + \alpha_k^2 \mathbf{d}^{(k)\top} \mathbf{Q} \mathbf{d}^{(k)} \end{aligned}$$

plugging in the expression for α_k given on the previous slide and rearranging (ugly maths):

$$f(\mathbf{x}^{(k+1)}) = \left(1 - \frac{(\mathbf{d}^{(k)\top} \mathbf{d}^{(k)})^2}{(\mathbf{d}^{(k)\top} \mathbf{Q} \mathbf{d}^{(k)}) (\mathbf{d}^{(k)\top} \mathbf{Q}^{-1} \mathbf{d}^{(k)})} \right) f(\mathbf{x}^{(k)})$$

Condition Number Example [3/3]

Kantorovich Inequality (stated without proof)

Let \mathbf{Q} be a positive definite $n \times n$ matrix. Then for any $\mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n$, the following inequality holds, where $m = \lambda_{\min}(\mathbf{Q})$ and $M = \lambda_{\max}(\mathbf{Q})$:

$$\frac{(\mathbf{x}^T \mathbf{x})^2}{(\mathbf{x}^T \mathbf{Q} \mathbf{x})(\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x})} \geq \frac{4\lambda_{\max}(\mathbf{Q})\lambda_{\min}(\mathbf{Q})}{(\lambda_{\max}(\mathbf{Q}) + \lambda_{\min}(\mathbf{Q}))^2} = \frac{4Mm}{(M+m)^2}.$$

Quadratic Minimisation Example (continued)

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &= \left(1 - \frac{(\mathbf{d}^{(k)\top} \mathbf{d}^{(k)})^2}{(\mathbf{d}^{(k)\top} \mathbf{Q} \mathbf{d}^{(k)}) (\mathbf{d}^{(k)\top} \mathbf{Q}^{-1} \mathbf{d}^{(k)})}\right) f(\mathbf{x}^{(k)}) \\ &\leq \left(1 - \frac{4Mm}{(M+m)^2}\right) f(\mathbf{x}^{(k)}) = \left(\frac{M-m}{M+m}\right)^2 f(\mathbf{x}^{(k)}) \\ &= \left(\frac{\chi-1}{\chi+1}\right)^2 f(\mathbf{x}^{(k)}) \leq \left(\frac{\chi-1}{\chi+1}\right)^{2(k+1)} f(\mathbf{x}^{(0)}), \end{aligned}$$

and the speed of convergence depends on the condition number χ .

How to deal with poor condition numbers? [1/3]

Motivation: Ill-conditioned optimisation problems are difficult

One way to circumvent this issue is to *condition* the problem by making appropriate linear transformation of the decision variables.

How to condition an unconstrained optimisation problem [Beck, 2014]

Consider:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \{f(\mathbf{x})\}.$$

For a given nonsingular matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, we make the linear transformation $\mathbf{x} = \mathbf{S}\mathbf{y}$ and obtain the equivalent problem:

$$\min_{\mathbf{y} \in \mathbb{R}^n} \{g(\mathbf{y}) \equiv f(\mathbf{S}\mathbf{y})\}.$$

Since $\nabla g(\mathbf{y}) = \mathbf{S}^\top \nabla f(\mathbf{S}\mathbf{y}) = \mathbf{S}^\top \nabla f(\mathbf{x})$, it follows that the gradient method applied to the transformed problem takes the form:

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \alpha_k \mathbf{S}^\top \nabla f(\mathbf{S}\mathbf{y}^{(k)}).$$

How to deal with poor condition numbers? [2/3]

How to condition an unconstrained optimisation problem (cont.)

Recall the gradient method of the transformed problem:

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \alpha_k \mathbf{S}^\top \nabla f(\mathbf{S}\mathbf{y}^{(k)}).$$

Multiplying this equality by \mathbf{S} on the left, using the notation $\mathbf{x}_k = \mathbf{S}\mathbf{y}_k$, and defining $\mathbf{D} = \mathbf{S}\mathbf{S}^\top$:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{S}\mathbf{S}^\top \nabla f(\mathbf{x}^{(k)}) = \mathbf{x}^{(k)} - \alpha_k \mathbf{D} \nabla f(\mathbf{x}^{(k)}),$$

we have the *scaled gradient method* with scaling matrix \mathbf{D} . By definition, \mathbf{D} is positive definite. The direction $-\mathbf{D} \nabla f(\mathbf{x}^{(k)})$ is a descent direction of f at $\mathbf{x}^{(k)}$ when $\nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$ because $-\nabla f(\mathbf{x}^{(k)})^\top \mathbf{D} \nabla f(\mathbf{x}^{(k)}) < 0$.

How to deal with poor condition numbers? [3/3]

How to condition an unconstrained optimisation problem (cont.)

The scaled gradient method with scaling matrix \mathbf{D} is equivalent to the gradient method on the function $g(\mathbf{y}) = f(\mathbf{D}^{1/2}\mathbf{x})$. The gradient and Hessian of g where $\mathbf{x} = \mathbf{D}^{1/2}\mathbf{y}$:

$$\nabla g(\mathbf{y}) = \mathbf{D}^{1/2} \nabla f(\mathbf{D}^{1/2}\mathbf{y}) = \mathbf{D}^{1/2} \nabla f(\mathbf{x}),$$

$$\nabla^2 g(\mathbf{y}) = \mathbf{D}^{1/2} \nabla^2 f(\mathbf{D}^{1/2}\mathbf{y}) \mathbf{D}^{1/2} = \mathbf{D}^{1/2} \nabla^2 f(\mathbf{x}) \mathbf{D}^{1/2}.$$

Scaled Gradient Method

Input: $\epsilon \Rightarrow$ tolerance Parameter. **Initialisation:** Pick $\mathbf{x}^{(0)} \in \mathbb{R}^n$ arbitrarily.

General step: For any $k = 0, 1, 2, \dots$ execute the following steps:

- Pick a scaling matrix $\mathbf{D}_k \succ \mathbf{0}$.
- Pick a stepsize α_k by line search: $h(\alpha) = f(\mathbf{x}^{(k)} - \alpha \mathbf{D}_k \nabla f(\mathbf{x}^{(k)}))$.
- Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{D}_k \nabla f(\mathbf{x}^{(k)})$. If $\|\nabla f(\mathbf{x}^{(k)})\|_2 \leq \epsilon$, STOP.

How to choose the scaling matrix?

Want the scaled Hessian as close as possible to the identity matrix

To accelerate the rate of convergence, which depends on the scaled Hessian $\mathbf{D}_k^{1/2} \nabla^2 f(\mathbf{x}^{(k)}) \mathbf{D}_k^{1/2}$, the scaling matrix is often chosen to make the scaled Hessian as close as possible to the identity matrix.

Sanity Check

If $\nabla^2 f(\mathbf{x}^{(k)}) \succ \mathbf{0}$, what is a good choice for \mathbf{D}_k ?

Chose \mathbf{D}_k to get Newton's method!

We can chose $\mathbf{D}_k = (\nabla^2 f(\mathbf{x}^{(k)}))^{-1}$ and the scaled Hessian becomes the identity matrix! The result is Newton's method:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \left(\nabla^2 f(\mathbf{x}^{(k)}) \right)^{-1} \nabla f(\mathbf{x}^{(k)})$$

Could be computationally expensive. Consider a diagonal scaling matrix!

Summary

