Introduction to C477 & Applications-Based Introduction to Optimisation

Ruth Misener
r.misener@imperial.ac.uk

Panos Parpas
p.parpas@imperial.ac.uk

Computational Optimisation Group
Department of Computing

13 January 2020

Course Basics: Who we are & how to contact us

- **Course Leaders**

Ruth Misener	Huxley 379	`r.misener@imperial.ac.uk`
Panos Parpas	Huxley 357	`p.parpas@imperial.ac.uk`

- **Tutorial Helpers**

Johannes Wiebe	`j.wiebe17@imperial.ac.uk`
Daniel Lengyel	`d.lengyel19@imperial.ac.uk`

- **How to access us**
  - *Office hours:* By appointment
  - *Piazza:* `http://piazza.com/imperial.ac.uk/spring2020/co477/home`

Course Basics: Assessment

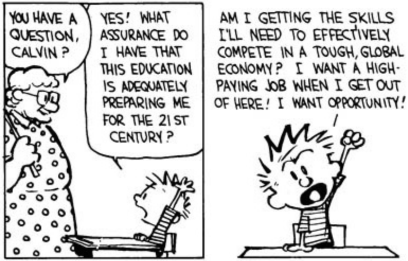
- Lecture slides, additional notes & tutorials available on **CATE**
- Tutorial answers available **one week** after tutorial sheet is given
- **Two** assessed courseworks
- **Active participation** expected/encouraged
  - Participation on Piazza counts as active participation

Course Basics: Piazza

piazza.com/imperial.ac.uk/spring2020/co477

- onjsq4he3u
- All non-personal C477-related traffic will go to Piazza. We will repost (suitably anonymised) email questions on Piazza;
- We aim to address administrative messages within 1 working day;
- All other posts will be left for at least 1 working day before an instructor response; this is to encourage student-led discussions.

Course Basics: SOLE



We need your feedback!

- We are continually changing C477 based on feedback given during class, on Piazza, and on SOLE;
- Please help us continually improve C477 by giving feedback, both positive and negative! We need to know: What should change? What should stay the same?

What is optimisation?

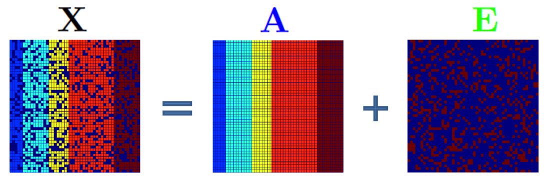
$$\min f(\boldsymbol{x})$$
$$g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, m.$$

$\boldsymbol{x} = [x_1, \ldots, x_n]^\top$	Optimisation, decision, design variables
$f : \mathbb{R}^n \mapsto R$	Objective function
$g_i(\boldsymbol{x}) \leq 0$	Constraints

- A vector $\boldsymbol{x}^*$ is **optimal** if it is **feasible**, $g_i(\boldsymbol{x}^*) \leq 0$, $i = 1, \ldots, m$ and has **lower** objective function value $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all other feasible points x;
- **Modelling**: Formulating real world problems into optimisation models
- **Optimisation Algorithms**: Computational methods solving optimisation models

Example: Robust Principal Component Analysis (PCA)

Challenge: Given $X = \mathbf{A} + \mathbf{E}$, recover $\mathbf{A}$ & $\mathbf{E}$



Optimisation problem: What is the lowest rank matrix that agrees with the data up to some sparse error?

$$\min_{\mathbf{A},\,\mathbf{E}} \operatorname{rank}(\mathbf{A}) + \lambda \left\|\mathbf{E}\right\|_0$$
$$X = \mathbf{A} + \mathbf{E}$$

This is an **NP-hard** problem, but using **convexity** allows us to recover almost any matrix of rank $\mathcal{O}(m/log^2 n)$ from errors corrupting $\mathcal{O}(m\ n)$ of the observations [At Imperial: Pantic, Zafeiriou].

Applications of Global Optimisation

In engineering applications, accurate mathematical modelling may require **discrete decisions** and **nonlinear relationships**

- Engineering design & manufacturing
  - Product & process design
  - Production planning/scheduling/logistics
- Computational chemistry
  - Chemical & phase equilibria
  - Molecular design
- Biochemistry & biochemical sciences:
  - Molecular (protein) structure prediction
  - Diagnosis, e.g., cancer

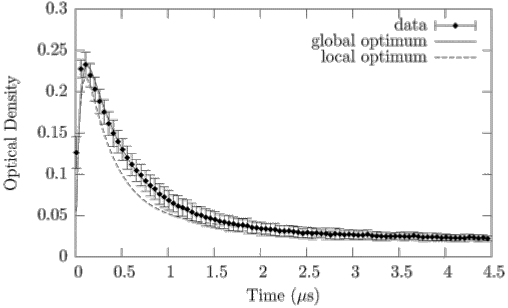
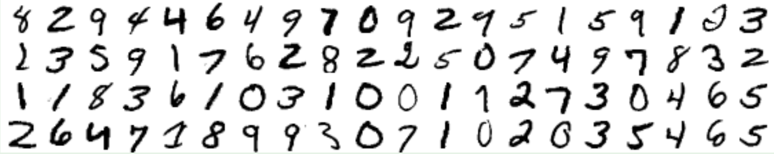


Figure 1: Singer et al., 2006

Example: Sparse Logistic Regression

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

$$\min_{\boldsymbol{\beta_1}, \ldots, \boldsymbol{\beta_{10}}} \sum_{i=1}^{m} \left(\log \sum_{k=1}^{10} \exp(\boldsymbol{x_i}^\top \boldsymbol{\beta_k}) - \boldsymbol{x_i}^\top \boldsymbol{\beta_{y_i}} \right) + \lambda \sum_{k=1}^{10} \|\boldsymbol{\beta_k}\|_1$$



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

Use **first-order**, gradient-based methods, to solve this example.

Prerequisites

Required

- **C145** Mathematical Methods (for DoC MEng / BEng students) or similar for MSc students;
- *C145 develops familiarity with* linear algebra, functions of several variables, multivariable calculus.
- You can read/write simple MATLAB programs (10-20 lines)

Highly Recommended

- **C233:** Computational Techniques (or similar)
- **C343:** Operations Research (or similar)
- **C496:** Mathematics for Inference and Machine Learning (or similar)

How to prepare for C477

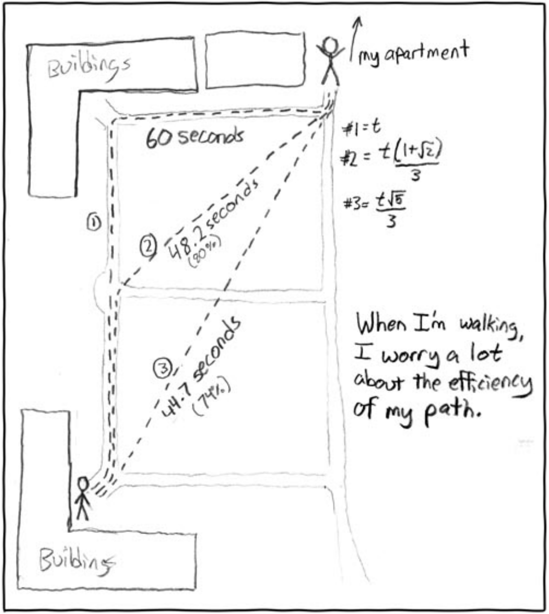
Self-assess & Get feedback

- Self-assess your level-of-preparation with **Mathematics Background** handout and quiz;
- Post to Piazza
- Quiz 0 posted on CATE does not affect your mark. The answers to the quiz will be posted on the morning of 15 October.

Remember that this is an optimisation course

We'll cover several applications of optimisation including engineering, machine learning, and finance. But C477 prioritises modelling and solving the optimisation problems over understanding the application.

Any questions on the review sheet or quiz?



Sanity Check

Does one of the paths on the left represent $\|\cdot\|_1$? Or $\|\cdot\|_2$? Or $\|\cdot\|_\infty$?

Recall Definitions

1-norm

$\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \ldots + |x_n|$

2-norm

$\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$

∞-norm

$\|\boldsymbol{x}\|_\infty = \max_i |x_i|$

`http://xkcd.com/85/`

Optimisation in C477

- Functions with n dimensions, e.g., many assets, many flights, ...
- Constraints, e.g., budget constraints, maximum wait time, ...
- No closed form solutions, we will study **numerical algorithms**
- Applications
  - Computer Science, e.g., Bayesian Optimisation, Neural Networks, Support Vector Machines, Genetic Programming;
  - Engineering, e.g., Capacity Expansion, Uncertainty & Energy;
  - Finance, e.g., Portfolio Optimisation.

Aims of C477

Goals

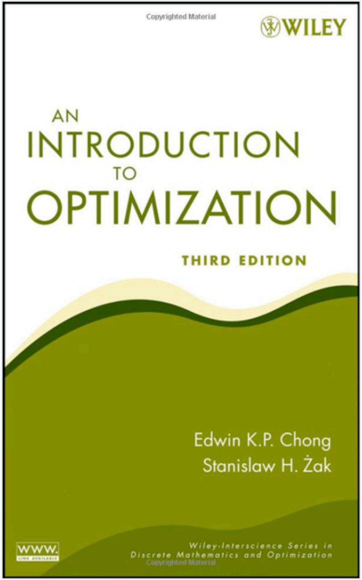
1. Understand the basic concepts of **mathematical optimisation**:
   a) Formulate model given a description
   b) Conditions under which a candidate decision is optimal
   c) Know the class of optimisation models that are tractable
2. **Computational optimisation algorithms:** Know the main classes of algorithms and which algorithm to apply to which problem
3. **Develop small code** (10-50 lines) for modelling & algorithm development

Planned Course Outline

1. Classifying types of optimisation problems
2. Convexity
   - **Applications:** Energy efficiency, solver software, Robust principal component analysis
3. Optimisation & Optimality Conditions
4. One Dimensional Search Algorithms
5. Unconstrained Optimisation
   - **Classes of methods:** Zeroth, First, & Second Order
   - **Algorithms:** Steepest Descent, Newton-Raphson, ...
   - **Applications:** Bayesian Optimisation, Sparse Logistic Regression, Neural Networks, ...
6. Constrained Optimisation
   - **Theory:** Optimality Conditions
   - **Applications:** Portfolio selection in finance
   - **Algorithms:** Lagrangian & Penalty methods

Recommended Book

- Most material covered in C477 is also covered by Chong & Żak;
- Mathematical levels of C477 & Chong & Żak are similar;
- Matlab code is very useful;
- Exercises there are useful.



Second Recommended Book

- Covers more material than C477;
- More mathematically rigorous than C477;
- Fewer applications in this book.