Optimization Lecture

Manickam S.
[manickam.somasundaram@proplant-mvi.com]

CAx- and AI-Solutions, MVI Proplant Nord GmbH

November 9, 2023

Introduction

Understand $\leftrightarrow$ **Search** $\leftrightarrow$ Solve

- Efficient algorithms exist for many problems.
- Implementations of these algorithms are also available.
- The remaining difficulty is often having enough prior knowledge to recognize a problem.

Figure: Johann Bernoulli, thanks for Brachistochrone curve

Transferring the problem to code

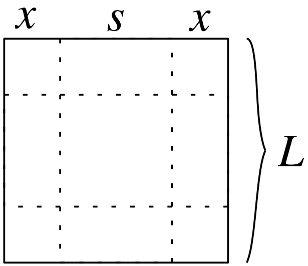
(1) The paper box problem

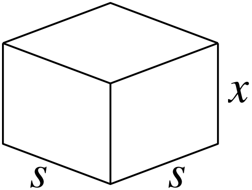
Introduction

Given a square piece of paper, cut the paper at a location x to maximize the volume of the resulting box.

This problem can be formulated as

$$
\begin{aligned}
\text{minimize} \quad & -V(x,s) = -s^2 x \\
\text{subject to} \quad & 2x + s = L \\
& s, x \geq 0.
\end{aligned}
$$

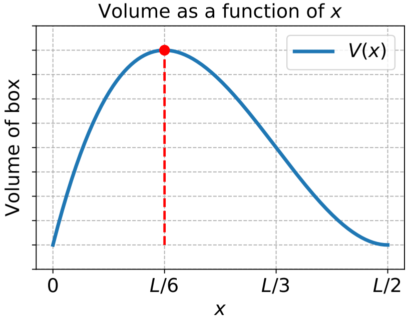




(1) The paper box problem

Problem instance

$$\begin{aligned} \text{minimize} \quad & -V(x) = -(L-2x)^2 x \\ \text{subject to} \quad & x \geq 0 \\ & x \leq L/2 \end{aligned}$$



Problem generalization

Given a smooth function $f : \mathbb{R} \to \mathbb{R}$.

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \geq a \\ & x \leq b \end{aligned}$$

Solved using differentiation.

(2) The advertisement problem

Introduction

We want equal exposure to 4 segments. Given 2 advertisement channels and their associated reach in units of views/dollar, allocate the money.

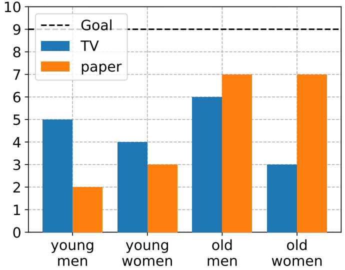
$$\begin{pmatrix}5\\4\\6\\3\end{pmatrix} x_1 + \begin{pmatrix}2\\3\\7\\7\end{pmatrix} x_2 = \begin{pmatrix}9\\9\\9\\9\end{pmatrix} + \begin{pmatrix}e_1\\e_2\\e_3\\e_4\end{pmatrix}$$

This problem can be formulated as

$$\text{minimize} \quad \sum_{i=1}^{4} e_i^2 = \mathbf{e}^T\mathbf{e} = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 .$$

Views per unit of money

$$A = \begin{array}{c} \\ \text{y} ♀ \\ \text{y} ♂ \\ \text{o} ♀ \\ \text{o} ♂ \end{array} \begin{array}{c} \begin{array}{cc} \text{tv} & \text{paper} \end{array} \\ \begin{pmatrix} 5 & 2 \\ 4 & 3 \\ 6 & 7 \\ 3 & 7 \end{pmatrix} \end{array}$$



(2) The advertisement problem

Problem instance

minimize $\quad \mathbf{e}^T\mathbf{e} = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$

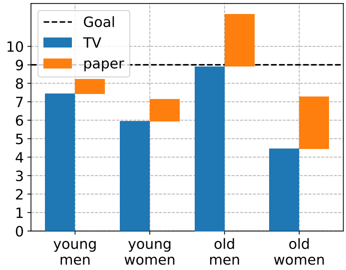
Problem generalization

Minimizing $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ is a *least squares problem*, solved analytically by the equation

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}.$$

Solution

$$\begin{pmatrix}5\\4\\6\\3\end{pmatrix} 1.5 + \begin{pmatrix}2\\3\\7\\7\end{pmatrix} 0.4 = \begin{pmatrix}9\\9\\9\\9\end{pmatrix} + \begin{pmatrix}-0.7\\-1.8\\2.8\\-1.7\end{pmatrix}$$



(3) The *constrained* advertisement problem

Introduction

Same as before, but constrained by a budget of 1 unit of money.

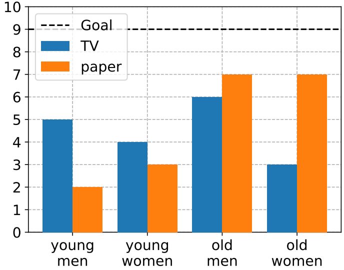
$$\begin{pmatrix}5\\4\\6\\3\end{pmatrix} x_1 + \begin{pmatrix}2\\3\\7\\7\end{pmatrix} x_2 = \begin{pmatrix}9\\9\\9\\9\end{pmatrix} + \begin{pmatrix}e_1\\e_2\\e_3\\e_4\end{pmatrix}$$

This problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{4} e_i^2 = \mathbf{e}^T\mathbf{e} = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{subject to} \quad & x_1 + x_2 = 1 \end{aligned}$$

Views per unit of money

$$A = \begin{array}{c} \\ \text{y}\,♀ \\ \text{y}\,♂ \\ \text{o}\,♀ \\ \text{o}\,♂ \end{array} \begin{array}{c} \begin{array}{cc} \text{tv} & \text{paper} \end{array} \\ \begin{pmatrix} 5 & 2 \\ 4 & 3 \\ 6 & 7 \\ 3 & 7 \end{pmatrix} \end{array}$$



The *constrained* advertisement problem

Problem instance

$$\begin{array}{rl} \text{minimize} & \mathbf{e}^T\mathbf{e} = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{subject to} & \mathbf{x}^T\mathbf{1} = 1 \end{array}$$

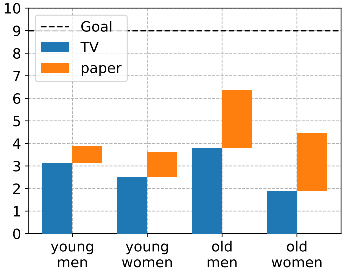
Problem generalization

Constrained least squares problem, solved by Lagrange multipliers and linear algebra.

$$\begin{pmatrix} 2\mathbf{A}^T\mathbf{A} & \mathbf{1} \\ \mathbf{1} & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}} \end{pmatrix} = \begin{pmatrix} 2\mathbf{A}^T\mathbf{b} \\ \mathbf{d} \end{pmatrix}$$

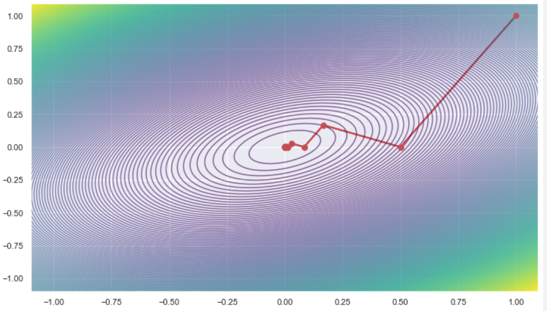
Solution

$$\begin{pmatrix} 5 \\ 4 \\ 6 \\ 3 \end{pmatrix} \mathbf{0.6} + \begin{pmatrix} 2 \\ 3 \\ 7 \\ 7 \end{pmatrix} \mathbf{0.4} = \begin{pmatrix} 9 \\ 9 \\ 9 \\ 9 \end{pmatrix} + \begin{pmatrix} -5.1 \\ -5.4 \\ -2.6 \\ -4.5 \end{pmatrix}$$



(4) Gradient Descent

- Gradient Descent is a fundamental optimization technique used in machine learning and mathematical optimization.
- It is the backbone of training algorithms for various models, such as linear regression and neural networks.
- Gradient Descent aims to find the minimum of a given function by iteratively adjusting model parameters.



Key Concepts

- **Objective Function**: The function to be minimized or maximized, typically represented as $J(\theta)$.
- **Gradient**: The vector of partial derivatives of the objective function with respect to the model parameters, $\nabla J(\theta)$.
- **Learning Rate (α)**: A hyperparameter that controls the step size during each iteration.
- **Iteration**: Each cycle of updating model parameters based on the gradient.

Basic Steps of Gradient Descent

1. Start with an initial guess for model parameters, θ.
2. Compute the gradient of the objective function with respect to θ.
3. Update θ by subtracting the gradient scaled by the learning rate: $\theta \leftarrow \theta - \alpha \nabla J(\theta)$.
4. Repeat steps 2 and 3 until convergence or a predefined number of iterations.

(5) Simulated Annealing

- Simulated Annealing is a stochastic optimization technique inspired by the annealing process in metallurgy.
- It is used to find approximate solutions to optimization and search problems.
- Simulated Annealing is particularly useful for problems with rugged landscapes and multiple local optima.

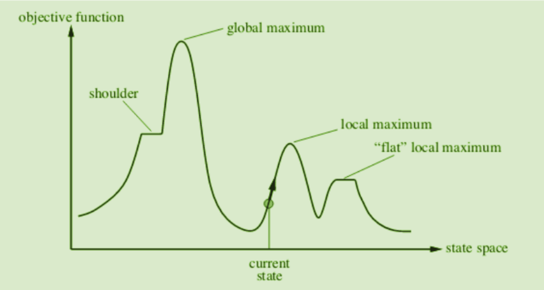


Figure: Convergence of Gradient Descent

Key Concepts

- **Objective Function**: The function to be minimized or maximized.
- **State Space**: The set of all possible solutions.
- **Energy**: A measure of the objective function's value.
- **Temperature**: A control parameter that regulates exploration vs. exploitation.
- **Cooling Rate**: The rate at which temperature decreases during the annealing process.

Basic Steps of Simulated Annealing

1. Start with an initial solution.
2. Initialize the temperature and cooling rate.
3. Iterate through a predefined number of steps or until convergence.
4. Perturb the current solution to obtain a neighboring solution.
5. Calculate the change in energy between the current and neighboring solutions.
6. Accept or reject the neighboring solution based on energy change and temperature.
7. Reduce the temperature according to the cooling rate.

Acceptance Probability

The probability of accepting a worse solution:

$$P(\text{accept}) = \exp\left(-\frac{\Delta E}{T}\right)$$

where ΔE is the change in energy, and T is the current temperature.

Key Considerations

- Simulated Annealing is a probabilistic method.
- It can escape local optima by accepting worse solutions early in the optimization.
- The algorithm's performance depends on the choice of temperature and cooling rate.
- The convergence to the global optimum is not guaranteed.

Applications

Simulated Annealing is used in various fields, including:

- Combinatorial optimization (e.g., Traveling Salesman Problem).
- Machine learning (e.g., training neural networks).
- VLSI design and chip manufacturing.
- Structural optimization in engineering.

Conclusion

- Simulated Annealing is a powerful optimization technique for tackling complex problems.
- It offers a balance between exploration and exploitation.
- Proper parameter tuning and careful design of the objective function are crucial for success.

(6) The worker-assignment problem

Introduction

Assign 4 workers to 4 tasks, given a matrix C specifying to which degree workers enjoy each task.

This amounts to specifying X with entries in $X_{ij} \in \{0, 1\}$, i.e.

$$X = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

The above yields a satisfaction of

$$6 + 4 + 8 + 1 = 19.$$

Problem data

$$C = \begin{array}{l} \\ \text{ole} \\ \text{åse} \\ \text{dag} \\ \text{lise} \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 5 & 6 & 1 & 6 \\ 4 & 5 & 0 & 1 \\ 1 & 2 & 6 & 8 \\ 7 & 2 & 1 & 1 \end{pmatrix} \end{array}$$

Solution space growth

n	digits in $n!$
1	1
10	7
25	26
50	65

The worker-assignment problem

Problem instance

$$\begin{array}{rl} \text{minimize} & -\sum_i \sum_j C_{ij} X_{ij} \\ \text{subject to} & \sum_i X_{ij} = 1 \text{ for every } j \\ & \sum_j X_{ij} = 1 \text{ for every } i \end{array}$$

Problem generalization

This is the *assignment problem*, solved in $\mathcal{O}(n^3)$ time, not $\mathcal{O}(n!)$.

Solution

$$C_{ij}\widehat{X}_{ij} = \begin{array}{l} \\ \text{ole} \\ \text{åse} \\ \text{dag} \\ \text{lise} \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 5 & 6 & 1 & \mathbf{6} \\ 4 & \mathbf{5} & 0 & 1 \\ 1 & 2 & \mathbf{6} & 8 \\ \mathbf{7} & 2 & 1 & 1 \end{pmatrix} \end{array}$$

$$-\sum_i \sum_j C_{ij}\widehat{X}_{ij} = 6 + 4 + 6 + 7 = 23$$

(7) The hotel problem

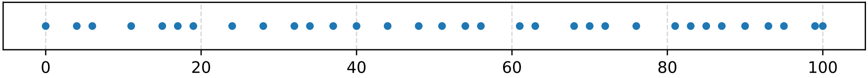
Introduction

We wish to travel 100 units of distance. There are many hotels along the way. Pick hotels to travel ~10 units per day.

Problem instance

Set $M = 10$. Find a sequence $h_1, h_2, \ldots, h_n$ to

$$\text{minimize} \quad \sum_j (M - (h_j - h_{j-1}))^2 .$$



Examples

Traveling from $x = 0$ to $x = 6$ incurs a penalty of $(10 - (6 - 0))^2 = 4^2$.
Traveling from $x = 0$ to $x = 11$ incurs a penalty of $(10 - (11 - 0))^2 = 1^2$.
There are 31 hotels above, and $2^{31} = 2\,147\,483\,648$ possibilities.

The hotel problem

Problem

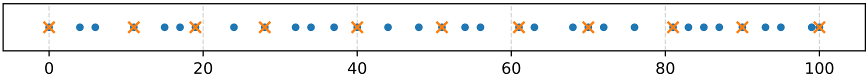
Let $P(j)$ be the minimal penalty at stop j. Realize that

$$P(j) = \min_{0 \leq i < j} \left(P(i) + (M - (h_j - h_i))^2 \right).$$

Solved in $\mathcal{O}(n^2)$ time, not $\mathcal{O}(2^n)$.

Problem generalization

The solution technique is called *dynamic programming* (DP). To use DP, we must (1) identify a recursive relationship, (2) define initial conditions and (3) solve problems in correct order.



Introduction

We are given 6 magnets. Choose $x_i \in \{-1, 1\}$ to minimize the total energy

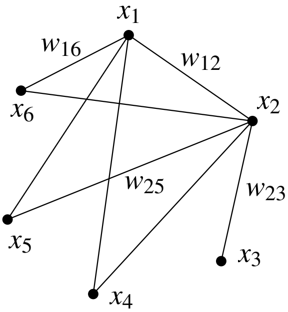
$$E(\mathbf{x}) = w_{12}x_1x_2 + w_{13}x_1x_3 + \cdots + w_{56}x_5x_6.$$

The problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & E(\mathbf{x}) = \mathbf{x}^T \mathbf{W} \mathbf{x} \\ \text{subject to} \quad & x_i \in \{-1, 1\}. \end{aligned}$$

There are 2^{6-1} states, and $E(\mathbf{x})$ is not differentiable. A difficult problem.

Problem

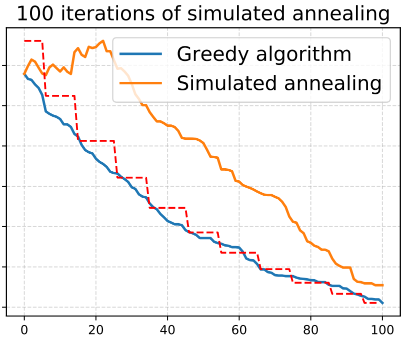


Problem instance

$$\begin{aligned} \text{minimize} \quad & E(\mathbf{x}) = \mathbf{x}^T \mathbf{W} \mathbf{x} \\ \text{subject to} \quad & x_i \in \{-1, 1\}. \end{aligned}$$

Problem generalization

Simulated annealing balances exploitation and exploration.
Widely applicable meta-heuristic.



(8) The egg boiling problem

Introduction

Let b be the boiling time of an egg, c be the cooling time, and s be the amount of salt used. Let $f(b,c,s) : \mathbb{R}^3 \to \mathbb{R}$ be the quality of a boiled egg.

This problem can be formulated as

$$\begin{array}{rl} \text{minimize} & -f(b,c,s) \\ \text{subject to} & b,c,s \geq 0 \end{array}$$

Evaluating $f(b,c,s)$ is expensive.

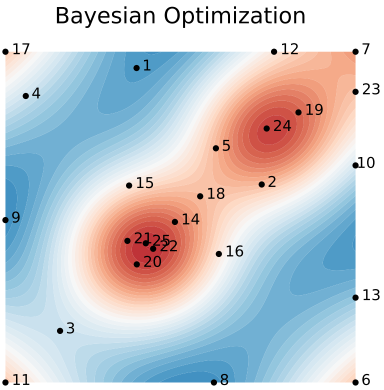
The egg boiling problem

Problem generalization

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ which is expensive to evaluate.

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{a} \leq \mathbf{x} \leq \mathbf{b} \end{aligned}$$

Clever sampling via *bayesian optimization*, which builds a probability distribution over functions. Exploration vs. exploitation.



References

- Strang, Gilbert. *Introduction to Applied Mathematics*. Wellesley, Mass: Wellesley-Cambridge Press, 1986.
- Boyd, Stephen, and Lieven Vandenberghe. *Introduction to Applied Linear Algebra*. Cambridge University Press, 2018.
- Dasgupta, Sanjoy, Christos H. Papadimitriou, and Umesh Virkumar. Vazirani. *Algorithms*. Boston, Mass: McGraw Hill, 2008.
- Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern Classification*. 2 edition. New York: Wiley-Interscience, 2000.
- Jasper Snoek, Hugo Larochelle, Ryan P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms*. arXiv.org, 2012.
- ChatGPT

Thanks

Thank you for your attention.

For LaTeX source, Python code and questions mail at:

`manickam.somasundaram@proplant-mvi.com`