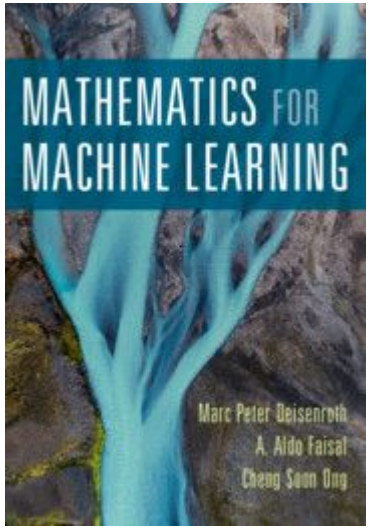
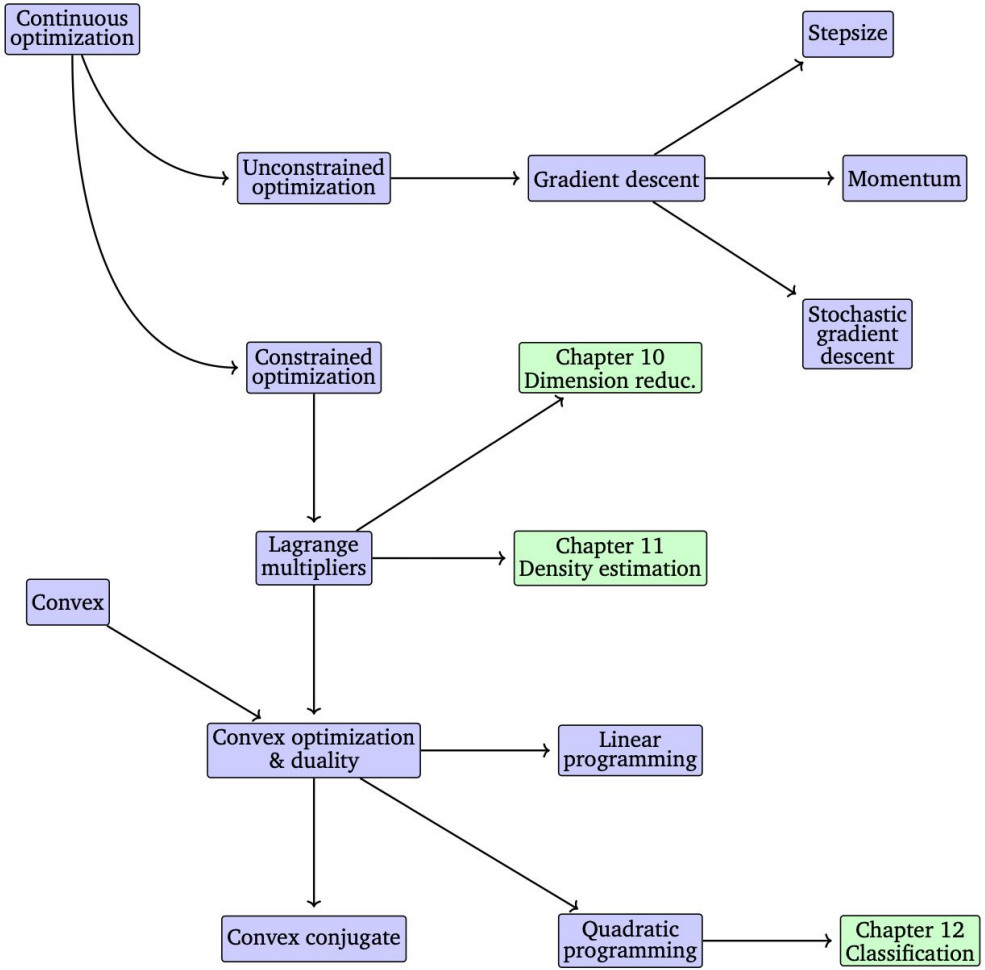
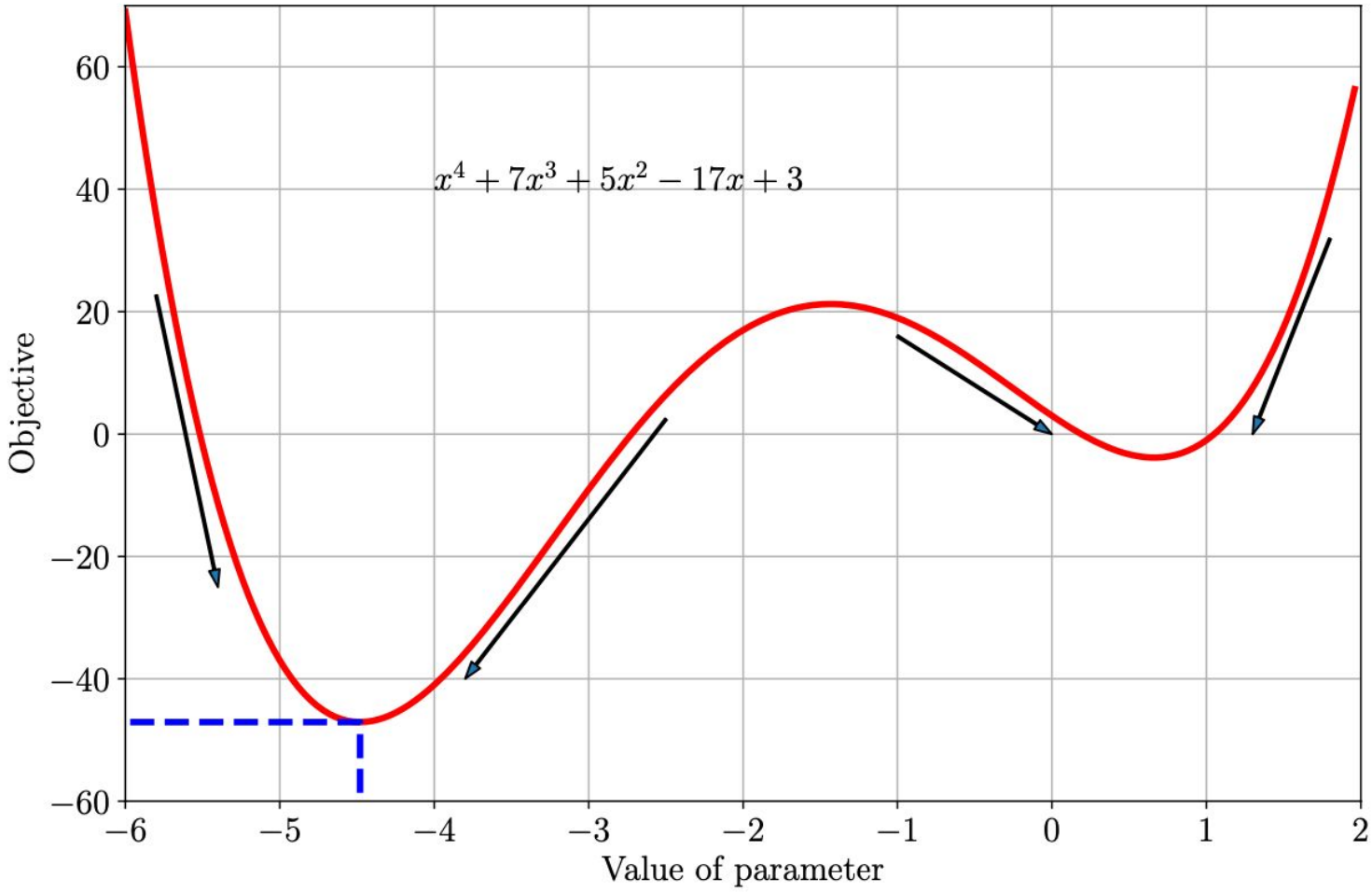


Chapter 7. Continuous Optimization
San Diego Machine Learning
Ryan Chesler

Continuous Optimization

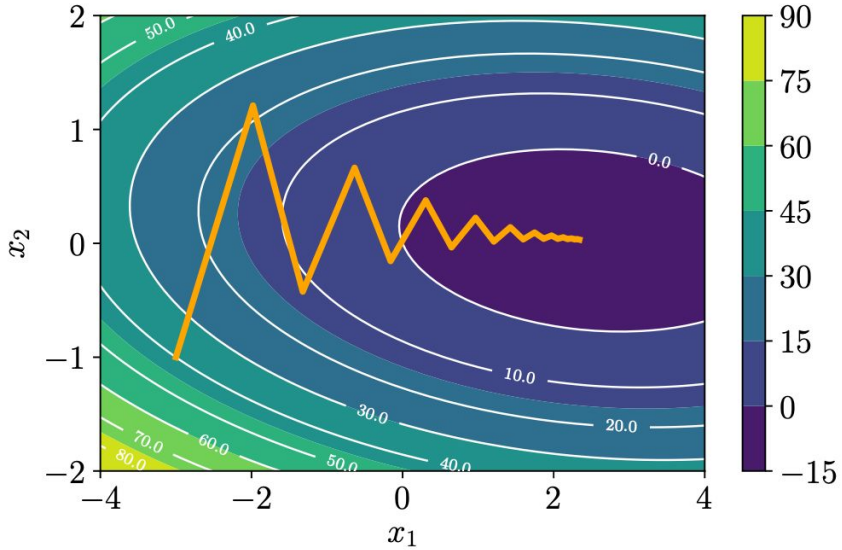
- In its simplest form, we have an objective function/loss
- We want to change the parameters of a system to make that number go down
- In neural networks we typically assume this is differentiable
  - Our techniques for optimizing differentiable systems are much more developed
- Ideally we'd like to find a global minimum, but in practice this is hard in complex systems


Continuous optimization
Unconstrained optimization
Gradient descent
Stepsize
Momentum
Stochastic gradient descent
Constrained optimization
Chapter 10 Dimension reduc.
Lagrange multipliers
Chapter 11 Density estimation
Convex
Convex optimization & duality
Linear programming
Convex conjugate
Quadratic programming
Chapter 12 Classification


$x^4 + 7x^3 + 5x^2 - 17x + 3$
Objective
Value of parameter
60
40
20
0
−20
−40
−60
−6
−5
−4
−3
−2
−1
0
1
2

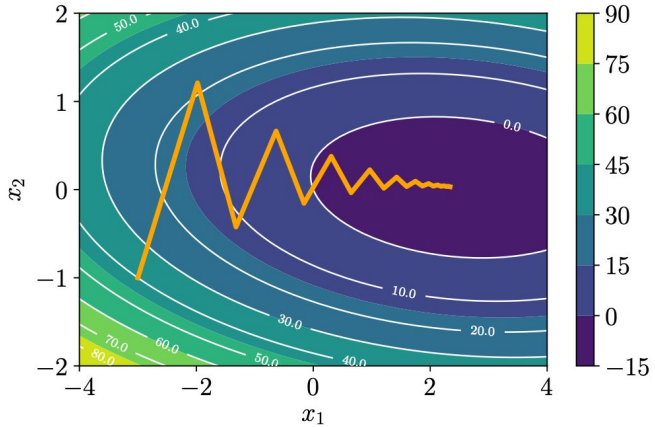
7.1 Optimization Using Gradient Descent

- Assuming our function is differentiable, we can calculate the gradient and move a step in the negative to move towards a minimum of the loss function
- We do not take a full step, we take a mini step scaled by some learning rate



7.1 Optimization Using Gradient Descent

- The speed of convergence is heavily dependent on the condition number
  - The ratio of maximum to minimum singular value
  - Curved direction vs least curved direction: long, thin valleys
- Can use a preconditioner
  - Idea is that there is some P^-1 that could shape the space to have a better condition number while being cheap to transform back and forth



7.1.2 Gradient Descent with Momentum

https://distill.pub/2017/momentum/

7.1.3 Stochastic Gradient Descent

- What we have discussed up to this point is assuming you calculate the gradient for the full dataset
- This is very expensive when you have a lot of data
- Does not converge as quickly
- SGD, takes individual points and takes steps based on them
- It is ok if it is noisy, as long as it is an unbiased estimate of the true gradient
- In practice we typically do mini-batches

https://arxiv.org/abs/1812.06162

7.2 Constrained Optimization and Lagrange Multipliers

- There might be inequality constraints in a system we are trying to optimize
  - Must keep values within a certain bound
- We could modify the space so that anything outside of range gets a huge loss, but this doesnt make it any easier to optimize
- Can be solved with Lagrange multipliers
  - https://www.youtube.com/watch?v=yuqB-d5MjZA
  - https://www.youtube.com/watch?v=aep6lwPqm6I

$$\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x})$$
$$= f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{g}(\boldsymbol{x}),$$

7.2 Constrained Optimization and Lagrange Multipliers

- Primal variables - Original set of variables x
- Dual variables - transformed set of variables λ

Definition 7.1. The problem in (7.17)

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leqslant 0 \quad \text{for all} \quad i = 1, \ldots, m \end{aligned} \tag{7.21}$$

$$\begin{aligned} \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \quad & \mathfrak{D}(\boldsymbol{\lambda}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geqslant \mathbf{0}\,, \end{aligned}$$

where $\boldsymbol{\lambda}$ are the dual variables and $\mathfrak{D}(\boldsymbol{\lambda}) = \min_{\boldsymbol{x} \in \mathbb{R}^d} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$.

7.3 Convex Optimization

- Convex set + convex functions = strong duality
  - Optimal solution for dual = primal
- Can check convexity with gradients and hessian
  - a straight line connecting any two elements of the set lie inside the set
- Jensen’s inequality - nonnegative weighted sums of convex functions

Figure 7.5 Example of a convex set.

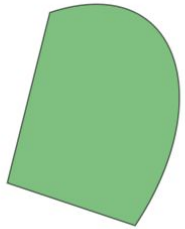
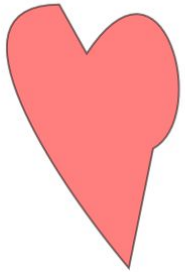


Figure 7.6 Example of a nonconvex set.



convex function

concave function

7.3.1 Linear Programming

Consider the special case when all the preceding functions are linear, i.e.,

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}\,, \end{aligned} \tag{7.39}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{R}^m$. This is known as a *linear program*. It has d variables and m linear constraints. The Lagrangian is given by

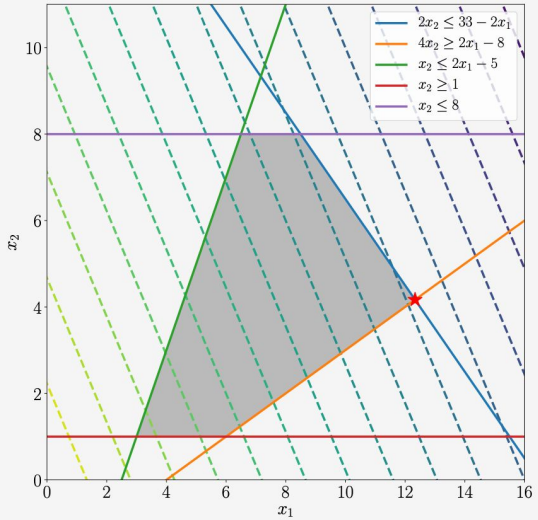
$$\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\,, \tag{7.40}$$

We can solve for the primal or dual program based on whether m or d is larger

Example 7.5 (Linear Program)
Consider the linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^2} \quad & -\begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix}2 & 2\\2 & -4\\-2 & 1\\0 & -1\\0 & 1\end{bmatrix} \begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}33\\8\\5\\-1\\8\end{bmatrix} \end{aligned} \tag{7.44}$$

with two variables. This program is also shown in Figure 7.9. The objective function is linear, resulting in linear contour lines. The constraint set in standard form is translated into the legend. The optimal value must lie in the shaded (feasible) region, and is indicated by the star.



7.3.2 Quadratic Programming

Consider the case of a convex quadratic objective function, where the constraints are affine, i.e.,

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A} \boldsymbol{x} \leqslant \boldsymbol{b}\,, \end{aligned} \tag{7.45}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times d}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{c} \in \mathbb{R}^d$. The square symmetric matrix $\boldsymbol{Q} \in \mathbb{R}^{d \times d}$ is positive definite, and therefore the objective function is convex. This is known as a *quadratic program*. Observe that it has d variables and m linear constraints.

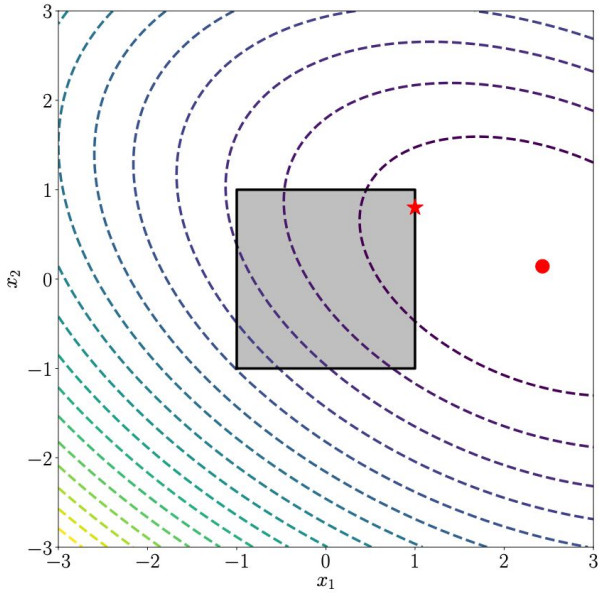
7.3.2 Quadratic Programming

Example 7.6 (Quadratic Program)
Consider the quadratic program

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \quad \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^\top \begin{bmatrix}2 & 1\\1 & 4\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix} \tag{7.46}$$

$$\text{subject to} \quad \begin{bmatrix}1 & 0\\-1 & 0\\0 & 1\\0 & -1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}1\\1\\1\\1\end{bmatrix} \tag{7.47}$$

of two variables. The program is also illustrated in Figure 7.4. The objective function is quadratic with a positive semidefinite matrix $\boldsymbol{Q}$, resulting in elliptical contour lines. The optimal value must lie in the shaded (feasible) region, and is indicated by the star.



7.3.3 Legendre-Fenchel Transform and Convex Conjugate

- We can fully describe a convex set by its supporting hyperplanes
  - Convex functions can be described by a function of their gradient
- A transformation that maintains all information
-