Optimization and Data Science

Lecture 20: Constrained Optimization Problems: Projection and Penalty Methods

Prof. Dr. Thomas Slawig

Kiel University - CAU Kiel
Dep. of Computer Science

Summer 2020

Contents

1. Constrained Optimization Problems: Projection and Penalty Methods
   - Adaption of Step-size
   - Projection Methods
   - Penalty Methods

Constrained optimization problems

- General form:

$$\min_{x \in X_{ad}} f(x)$$

  where the admissible or feasible set $X_{ad} \subset X$ is now a real subset of X.
- We consider $X = \mathbb{R}^n$.
- Often X_{ad} is defined by functions $g : \mathbb{R}^n \to \mathbb{R}^m, h : \mathbb{R}^n \to \mathbb{R}^p$:

$$X_{ad} := \{x \in \mathbb{R}^n : g(x) \leq 0, h(x) = 0\}.$$

$\rightsquigarrow$ We write

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} g(x) \leq 0 & \text{(inequality constraints)} \\ h(x) = 0 & \text{(equality constraints).} \end{cases}$$

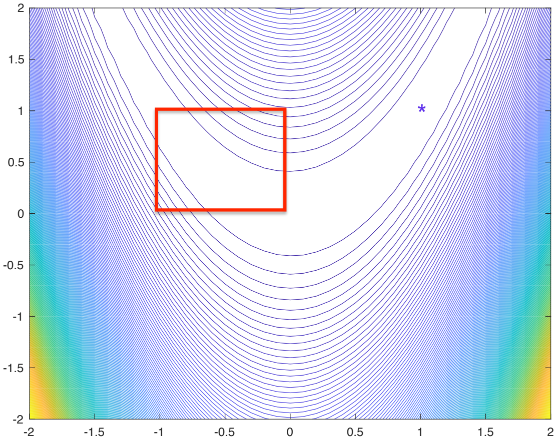
- An inequality constraint g_i is called **active** in x if $g_i(x) = 0$, and **inactive** if $g_i(x) < 0$.

Constrained optimization problem: Example

- Rosenbrock function

$$f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

- Unconstrained problem: mimimizer in $\mathbb{R}^2 : x^* = (1, 1)$.
- Usually different in constrained case on a subset X_{ad}.



Contents

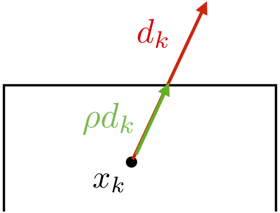
1 Constrained Optimization Problems: Projection and Penalty Methods
- Adaption of Step-size
- Projection Methods
- Penalty Methods

Adaption of step-size in descent methods

Algorithm (General descent method with line search):

1. Choose initial guess $x_0 \in \mathbb{R}^n$.
2. For $k = 0, 1, \dots$:
   1. Choose a descent direction $d_k \in \mathbb{R}^n$.
   2. Choose an efficient step-size $\rho_k > 0$ (e.g., with Armijo rule).
   3. Set $x_{k+1} = x_k + \rho_k d_k$.

   until a stopping criterion is satisfied.



- Problem if d_k points outwards of X_{ad}.

$\rightsquigarrow$ Reduce step-size ρ in the line search.

Adaption of step-size in descent methods

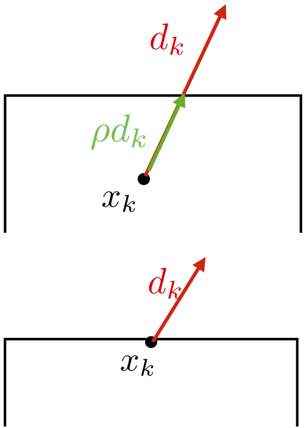
Algorithm (General descent method with line search):

1. Choose initial guess $x_0 \in \mathbb{R}^n$.
2. For $k = 0, 1, \ldots$:
    1. Choose a descent direction $d_k \in \mathbb{R}^n$.
    2. Choose an efficient step-size $\rho_k > 0$.

    $\rightarrow$ New: additionally ensure that X_{ad} is not left.

    3. Set $x_{k+1} = x_k + \rho_k d_k$.

   until a stopping criterion is satisfied.



- Not working if x_k is already boundary point and d_k points outwards of X_{ad}.

Contents

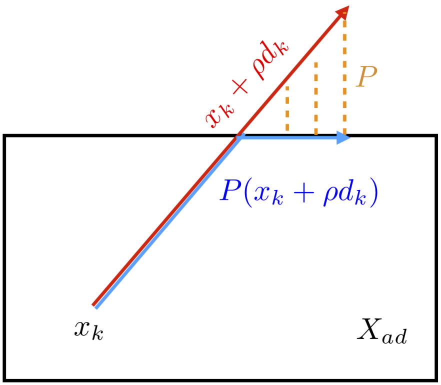
1 Constrained Optimization Problems: Projection and Penalty Methods
- Adaption of Step-size
- Projection Methods
- Penalty Methods

Projection Methods

- If the search direction points out of the admissible set X_{ad}, ...
- ... and the line search gives a point which is outside of X_{ad},
- ... we may project the resulting point onto the admissible set.
- Here, the mapping

$$P : \mathbb{R}^n \to X_{ad}$$

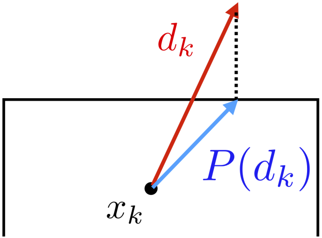
is called the **projection onto** X_{ad}.



- A projection satisfies $P^2 := P \circ P = P$ (i.e., projecting twice does not change the result).
- Applying this, e.g., for the gradient method, gives the **gradient projection method**.

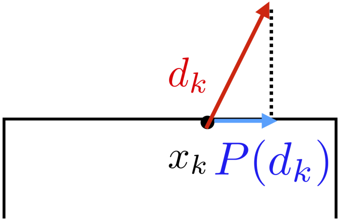
Projection Methods

- Another idea is to directly project the search direction onto the admissible set X_{ad} ...
- ... and use the projected direction as search direction.
- Applying this, e.g., for the gradient method, gives the **projected gradient method**.



- Both methods now also work if the current iterate x_k is on the boundary.
- A problem may occur if x_k is at the corner of X_{ad}.

$\rightsquigarrow$ stop.



Projection Methods: Idea

- For projection methods, we have to compute the projection.
- This is easy in the case of linear constraints, ...
- ... where the admissible set is given as:

$$X_{ad} := \{x \in \mathbb{R}^n : g(x) \le 0, h(x) = 0\}$$

with

$$\begin{aligned} h(x) &:= Ax - b = 0 \quad (\Leftrightarrow Ax = b) \\ g(x) &:= Cx - d \le 0 \quad (\Leftrightarrow Cx \le d) \end{aligned}$$

where $A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p, C \in \mathbb{R}^{m \times n}, d \in \mathbb{R}^m$.

- Linear constraints always define a **convex** admissible set X_{ad}.

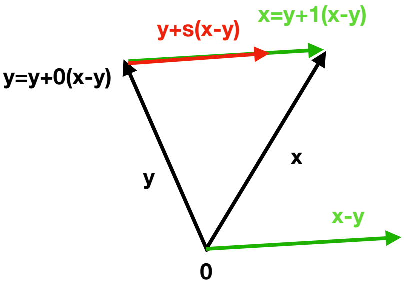
Convex Sets

Definition

A set $M \subset \mathbb{R}^n$ is called **convex**, if

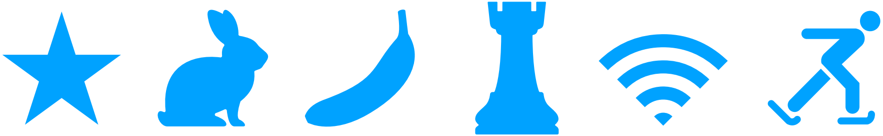
$$x, y \in M, s \in [0,1] \Longrightarrow y + s(x-y) = sx + (1-s)y \in M.$$

- For two points in the set, also the complete connecting line of the two points belongs to the set.



Convex Sets

convex



not convex

Linear constraints: Admissible set is convex

Lemma

For linear constraints, the admissible set is closed and convex.

Proof.

- X_{ad} is closed since g, h are continuous and we have " $\leq$ " for the inequality constraints.
- Convexity: $x, y \in X_{ad}$ satisfy $\quad Ax = b, Ay = b,\ Cx \leq d, Cy \leq d.$
- We have to show that for arbitrary $s \in [0,1] : sx + (1-s)y \in X_{ad}$, i.e.,

$$A\left(sx + (1-s)y\right) = b, \quad C\left(sx + (1-s)y\right) \leq d.$$

- Since the constraints are linear, we get:

$$\begin{aligned} A\left(sx + (1-s)y\right) &= sAx + (1-s)Ay = sb + (1-s)b = b, \\ C\left(sx + (1-s)y\right) &= sCx + (1-s)Cy \leq sd + (1-s)d = d. \end{aligned}$$

- $\Rightarrow sx + (1-s)y \in X_{ad}$ for all $s \in [0,1] \Rightarrow X_{ad}$ is convex. □

Projection onto the admissible set (linear constraints)

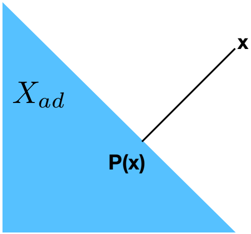
Lemma

If $X_{ad} \subset \mathbb{R}^n$ is defined by linear constraints and not empty, then the **orthogonal projection** *onto X_{ad} is well-defined and linear. It is represented by a matrix $P \in \mathbb{R}^{n\times n}$ that satisfies*

$$Px := \arg\min_{y \in X_{ad}} \|x - y\|_2, \quad x \in \mathbb{R}^n,$$

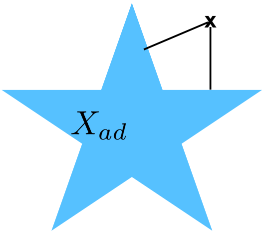
and

$$y - Py \perp x - Py, \text{ i.e., } (y - Py)^\top (x - Py) = 0 \text{ for all } y \in \mathbb{R}^n, x \in X_{ad}.$$



← convex, projection well-defined

not convex, projection not defined →



Projection for box constraints/simple bounds

- In the easiest case, we have just **box constraints**:

$$a \le x \le b \quad \text{with given } a, b \in \mathbb{R}^n$$

- They can be written as

$$\left.\begin{array}{r} g_i(x) = a_i - x_i \\ g_{n+i}(x) = x_i - b_i \end{array}\right\} \le 0, \quad i = 1, \dots, n.$$

- Then we get for the projection

$$y = Px \;:\Leftrightarrow\; y_i = \left\{\begin{array}{ll} a_i, & \text{if } x_i < a_i \\ b_i, & \text{if } x_i > b_i \\ x_i, & \text{elsewhere} \end{array}\right\}, \quad i = 1, \dots, n.$$

Contents

1 Constrained Optimization Problems: Projection and Penalty Methods
- Adaption of Step-size
- Projection Methods
- Penalty Methods

Penalty Method

- Idea: Transform the constrained problem into an unconstrained one.
- Penalize violation of constraints by addition of a penalty term to the cost function:
$$\min_{x\in\mathbb{R}^n} f(x) + c_k P(x), \quad c_k > 0.$$
- Then solve the unconstrained problem, increase c_k, iterate.

Definition

A continuous function $P : \mathbb{R}^n \to \mathbb{R}$ satisfying

$$P(x) = 0 \text{ for all } x \in X_{ad},$$
$$P(x) > 0 \text{ for all } x \in \mathbb{R}^n \setminus X_{ad},$$

or equivalently

$$P(x) \geq 0 \text{ for all } x \in \mathbb{R}^n,$$
$$P(x) = 0 \Leftrightarrow x \in X_{ad},$$

is called **penalty function**.

Penalty Functions: Examples

- Equality constraints $h(x) = 0$:

$$P(x) = \|h(x)\|_2^2 = \sum_{i=1}^{m} h_i(x)^2.$$

- Inequality constraints $g(x) \leq 0$:

$$P(x) = \sum_{i=1}^{p} (\max\{0, g_i(x)\})^2$$

Here, the square is used since the function is now differentiable at the point where $g_i(x) = 0$.

Penalty Method: Algorithm

Algorithm (Penalty Method)

1. Choose initial guess $x_0 \in \mathbb{R}^n$, $c_0 > 0$ and accuracy $\epsilon \geq 0$.
2. For $k = 1, 2, \dots$:
   (a) Starting with initial guess x_{k-1}, compute an approximative solution of

   $$\min_{x \in \mathbb{R}^n} f(x) + c_k P(x)$$

   with the given accuracy ϵ, i.e., find x_k such that

   $$f(x_k) + c_k P(x_k) \leq \min_{x \in \mathbb{R}^n} f(x) + c_k P(x) + \epsilon. \tag{1}$$

   (b) Increase penalty parameter: choose $c_{k+1} > c_k$.

   until a stopping criterion is satisfied.

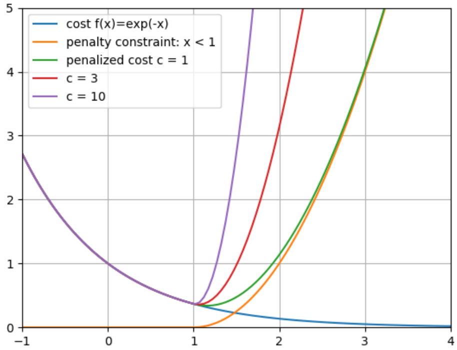
Properties of penalty methods

Lemma

If we solve the penalized (inner) problems exactly ($\epsilon = 0$), the iterates of the penalty method satisfy

$$
\begin{aligned}
P(x_k) &\geq P(x_{k+1}),\\
f(x_k) + c_k P(x_k) &\leq f(x_{k+1}) + c_{k+1} P(x_{k+1}),\\
f(x_k) &\leq f(x_{k+1}).
\end{aligned}
$$

- We prove the general result for $\epsilon \geq 0$.



Properties of penalty methods

Lemma

The iterates of the penalty method satisfy

$$P(x_k) + \delta_k \geq P(x_{k+1}),$$
$$f(x_k) + c_k P(x_k) \leq f(x_{k+1}) + c_{k+1} P(x_{k+1}) + \epsilon,$$
$$f(x_k) \leq f(x_{k+1}) + \gamma_k$$

with $\delta_k, \gamma_k \geq 0$ *depending on* ϵ. *If* $\epsilon = 0$, *then* $\delta_k = \gamma_k = 0$.

Proof.

Step 2a of the algorithm,

$$f(x_k) + c_k P(x_k) \leq \min_{x \in \mathbb{R}^n} f(x) + c_k P(x) + \epsilon,$$

gives

$$\begin{aligned} \text{applied for } k: \quad & f(x_k) + c_k P(x_k) \leq f(x_{k+1}) + c_k P(x_{k+1}) + \epsilon, \\ \text{applied for } k+1: \quad & f(x_{k+1}) + c_{k+1} P(x_{k+1}) \leq f(x_k) + c_{k+1} P(x_k) + \epsilon. \end{aligned}$$

Properties of penalty methods

Adding both inequalities and substracting the terms with f gives

$$\begin{array}{rcl} f(x_{k+1}) + c_k P(x_{k+1}) + \epsilon & \geq & f(x_k) + c_k P(x_k), \\ f(x_k) + c_{k+1} P(x_k) + \epsilon & \geq & f(x_{k+1}) + c_{k+1} P(x_{k+1}) \\ \hline (c_{k+1} - c_k) P(x_k) + 2\epsilon & \geq & (c_{k+1} - c_k) P(x_{k+1}). \end{array} \tag{2}$$

Dividing by $(c_{k+1} - c_k) > 0$, this is the first inequality in the Lemma with $\delta_k = \dfrac{2\epsilon}{c_{k+1} - c_k}$.

Because of $c_{k+1} > c_k$ and (2) we get

$$f(x_{k+1}) + c_{k+1} P(x_{k+1}) + \epsilon \geq f(x_{k+1}) + c_k P(x_{k+1}) + \epsilon \geq f(x_k) + c_k P(x_k).$$

This is the second inequality of the Lemma.

Inequality (2) and the first inequality of the Lemma give

$$f(x_{k+1}) + c_k P(x_{k+1}) + \epsilon \geq f(x_k) + c_k P(x_k) \geq f(x_k) + c_k P(x_{k+1}) - c_k \delta_k.$$

This is the third inequality of the Lemma with $\gamma_k = \epsilon + c_k \delta_k$. □

Properties of penalty methods

We can now give an upper bound for the function values in the penalty method:

Lemma

Let $x^* \in X_{ad}$ be a local minimizer of the original constrained problem. For the sequence of the iterates $(x_k)_{k\in\mathbb{N}}$ of the penalty algorithm, we then have

$$f(x^*) + \epsilon \geq f(x_k) + c_k P(x_k) \geq f(x_k) \quad \textit{for all } k \in \mathbb{N}.$$

Proof.

Let k be arbitrary. Since $x^* \in X_{ad}$, we have $P(x^*) = 0$ by definition of penalty functions. Because x_k is the result of step 2a in the algorithm, it satisfies

$$f(x_k) + c_k P(x_k) \leq \min_{x\in\mathbb{R}^n} f(x) + c_k P(x) + \epsilon \leq f(x^*) + c_k P(x^*) + \epsilon.$$

This gives

$$f(x^*) + \epsilon = f(x^*) + c_k P(x^*) + \epsilon \geq f(x_k) + c_k \underbrace{P(x_k)}_{\geq 0} \geq f(x_k). \quad \square$$

Convergence of penalty methods

Theorem

Let f be continuous and let $x^* \in X_{ad}$ be a local solution of the constrained problem. Then every accumulation point $\bar{x}$ of the sequence $(x_k)_k$ of iterates of the penalty method lies in X_{ad} and satisfies

$$f(\bar{x}) \leq f(x^*) + \epsilon. \tag{3}$$

Thus, for $\epsilon = 0$ the penalty method yields a local minimizer of the constrained problem.

Proof.

By the Lemma on page 24 we have

$$f(x_k) \leq f(x_k) + c_k P(x_k) \leq f(x^*) + \epsilon \quad \text{for all } k \in \mathbb{N}. \tag{4}$$

Let $\bar{x}$ be an accumulation point with $x_k \to \bar{x}$ for $k \to \infty, k \in \mathcal{K} \subset \mathbb{N}$.

Convergence of penalty methods

Passing to the limit in (4) gives, using the continuity of f:

$$f(\bar{x}) = \lim_{\mathcal{K}\ni k\to\infty} f(x_k) \leq \lim_{\mathcal{K}\ni k\to\infty} (f(x_k) + c_k P(x_k)) \leq f(x^*) + \epsilon.$$

This proves (3). It remains to show that $\bar{x} \in X_{ad}$: We had

$$\lim_{\mathcal{K}\ni k\to\infty} (f(x_k) + c_k P(x_k)) = f(\bar{x}) + \lim_{\mathcal{K}\ni k\to\infty} c_k P(x_k) \leq f(x^*) + \epsilon.$$

This means

$$\lim_{\mathcal{K}\ni k\to\infty} c_k P(x_k) \leq f(x^*) + \epsilon - f(\bar{x}) < \infty.$$

Since $P(x_k) \geq 0$ for all k and $c_k \to \infty$, we get

$$\lim_{\mathcal{K}\ni k\to\infty} P(x_k) = 0.$$

Continuity of the penalty function gives $P(\bar{x}) = 0$, i.e., $\bar{x} \in X_{ad}$ by definition of penalty functions. $\square$

Advantages and disadvantages of penalty methods

+ Every algorithm for unconstrained problems can be used.
– Effort: Nested iteration: in every step we have to solve (at least approximately) an unconstrained problem.
– Recall: convergence behavior of methods for unconstrained problem depends on eigenvalues of Hessian matrix $\nabla^2 f(x)$:
  For the penalty cost function $f + c_k P$ we get

$$\nabla^2(f + c_k P)(x) = \nabla^2 f(x) + c_k \nabla^2 P(x).$$

$\rightsquigarrow$ eigenvalues of $\nabla^2(f + c_k P)(x) \approx$ eigenvalues of $\nabla^2 f(x) + \underbrace{c_k \cdot \text{eigenvalues of } \nabla^2 P(x)}_{\text{grow if } c_k \text{ grows}}$.

- Remark: This is just a rough idea, eigenvalues can not be added this way!

* sequence $(c_k)_k$ is increasing.
* penalty functions are quadratic $\rightsquigarrow$ their Hessian matrix also has positive eigenvalues.

$\rightsquigarrow$ bad convergence properties.

What is important

- In constrained problems, the admissible set X_{ad} is a real subset of $\mathbb{R}^n$.
- It is usually defined by given functions.
- We distinguish between equality and inequality constraints.
- For constrained problems, we need adapted or different algorithms.
- In a descent method, we may try to just adapt the line search to always stay in the admissible set.
- We also may project the point found in the line search onto the admissible set ...
- ... or we directly project the search direction onto the admissible set.
- For linear constraints, the admissible set is convex.
- For convex admissible sets, the projection is given by matrix that can be easily computed.
- A different idea is to use penalty methods, where the constrained problem is transferred to an unconstrained one.