

Lecture 8: Introduction to Convex Optimization

Reading: *Convex optimization* by Boyd and Vandenberghe.

GU4241/GR5241 Statistical Machine Learning

Linxi Liu

February 22, 2019

Optimization Problems

Terminology

An **optimization problem** for a given function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is a problem of the form

$$\min_{\mathbf{x}} f(\mathbf{x})$$

which we read as "find $\mathbf{x}_0 = \arg \min_{\mathbf{x}} f(\mathbf{x})$ ".

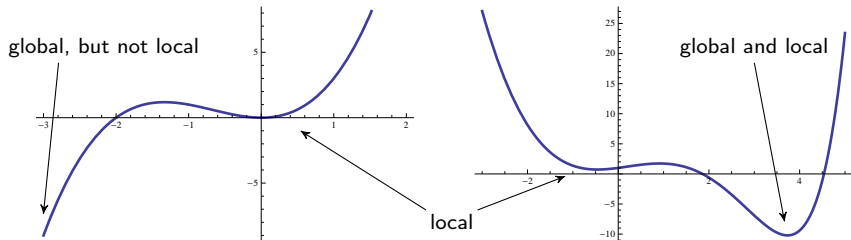
A **constrained optimization problem** adds additional requirements on \mathbf{x} ,

$$\begin{array}{ll} \min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in G, \end{array}$$

where $\text{dom} f \cap G \subset \mathbb{R}^d$ is called the **feasible set**. The set G is often defined by equations, e.g.

$$\begin{array}{ll} \min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{subject to} & g(\mathbf{x}) \geq 0 \end{array}$$

Types of Minima



Local and global minima

A minimum of f at x is called:

- ▶ **Global** if f assumes no smaller value on its domain.
- ▶ **Local** if there is some open neighborhood U of x such that $f(x)$ is a global minimum of f restricted to U .

Optima

Analytic criteria for local minima

Recall that \mathbf{x} is a local minimum of f if

$$f'(\mathbf{x}) = 0 \quad \text{and} \quad f''(\mathbf{x}) > 0 .$$

In \mathbb{R}^d ,

$$\nabla f(\mathbf{x}) = 0 \quad \text{and} \quad H_f(\mathbf{x}) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}) \right)_{i,j=1,\dots,n} \text{ positive definite.}$$

The $d \times d$ -matrix $H_f(\mathbf{x})$ is called the **Hessian matrix** of f at \mathbf{x} .

Optima

Numerical methods

All numerical minimization methods perform roughly the same steps:

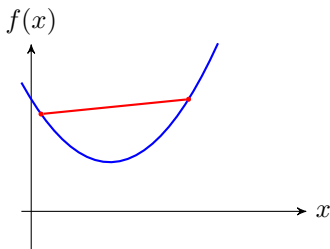
- ▶ Start with some point x_0 .
- ▶ Our goal is to find a sequence x_0, \dots, x_m such that $f(x_m)$ is a minimum.
- ▶ At a given point x_n , compute properties of f (such as $f'(x_n)$ and $f''(x_n)$).
- ▶ Based on these values, choose the next point x_{n+1} .

The information $f'(x_n)$, $f''(x_n)$ etc is always *local at x_n* , and we can only decide whether a point is a local minimum, not whether it is global.

Convex Functions

Definition

A function f is **convex** if every line segment between function values lies above the graph of f .



Analytic criterion

A twice differentiable function is convex if $f''(x) \geq 0$ (or $H_f(\mathbf{x})$ positive semidefinite) for all \mathbf{x} .

Implications for optimization

If f is convex, then:

- ▶ $f'(x) = 0$ is a sufficient criterion for a minimum.
- ▶ Local minima are global.
- ▶ If f is **strictly convex** ($f'' > 0$ or H_f positive definite), there is only one minimum (which is both global and local).

Gradient Descent

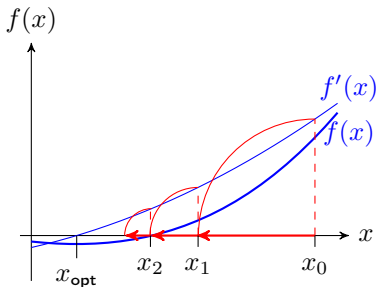
Algorithm

Gradient descent searches for a minimum of f .

1. Start with some point $x \in \mathbb{R}$ and fix a precision $\varepsilon > 0$.
2. Repeat for $n = 1, 2, \dots$

$$x_{n+1} := x_n - f'(x_n)$$

3. Terminate when $|f'(x_n)| < \varepsilon$.



Newton's Method: Roots

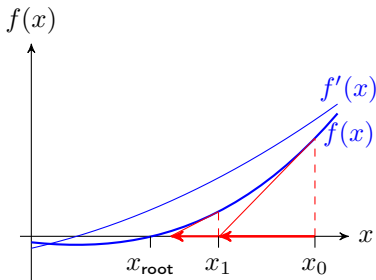
Algorithm

Newton's method searches for a **root** of f , i.e. it solves the equation $f(x) = 0$.

1. Start with some point $x \in \mathbb{R}$ and fix a precision $\varepsilon > 0$.
2. Repeat for $n = 1, 2, \dots$

$$x_{n+1} := x_n - f(x_n)/f'(x_n)$$

3. Terminate when $|f(x_n)| < \varepsilon$.



Basic Applications

Function evaluation

Most numerical evaluations of functions (\sqrt{a} , $\sin(a)$, $\exp(a)$, etc) are implemented using Newton's method. To evaluate g at a , we have to transform $x = g(a)$ into an equivalent equation of the form

$$f(x, a) = 0 .$$

We then fix a and solve for x using Newton's method for roots.

Example: Square root

To evaluate $g(a) = \sqrt{a}$, we can solve

$$f(x, a) = x^2 - a = 0 .$$

This is essentially how `sqrt()` is implemented in the standard C library.

Newton's Method: Minima

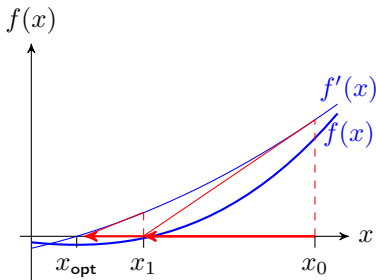
Algorithm

We can use Newton's method for minimization by applying it to solve $f'(x) = 0$.

1. Start with some point $x \in \mathbb{R}$ and fix a precision $\varepsilon > 0$.
2. Repeat for $n = 1, 2, \dots$

$$x_{n+1} := x_n - f'(x_n)/f''(x_n)$$

3. Terminate when $|f'(x_n)| < \varepsilon$.



Multiple Dimensions

In \mathbb{R}^d we have to replace the derivatives by their vector space analogues.

Gradient descent

$$\mathbf{x}_{n+1} := \mathbf{x}_n - \nabla f(\mathbf{x}_n)$$

Newton's method for minima

$$\mathbf{x}_{n+1} := \mathbf{x}_n - H_f^{-1}(\mathbf{x}_n) \cdot \nabla f(\mathbf{x}_n)$$

The inverse of $H_f(\mathbf{x})$ exists only if the matrix is positive definite (not if it is only semidefinite), i.e. f has to be strictly convex.

The Hessian measures the curvature of f .

Effect of the Hessian

Multiplication by H_f^{-1} in general changes the direction of $\nabla f(\mathbf{x}_n)$. The correction takes into account how $\nabla f(\mathbf{x})$ changes away from \mathbf{x}_n , as estimated using the Hessian at \mathbf{x}_n .

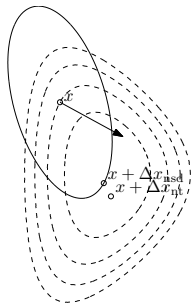


Figure: Arrow is ∇f , $x + \Delta x_{nt}$ is Newton step.

Newton: Properties

Convergence

- ▶ The algorithm always converges if $f'' > 0$ (or H_f positive definite).
- ▶ The speed of convergence separates into two phases:
 - ▶ In a (possibly small) region around the minimum, f can always be approximated by a quadratic function.
 - ▶ Once the algorithm reaches that region, the error decreases at quadratic rate. Roughly speaking, the number of correct digits in the solution doubles in each step.
 - ▶ Before it reaches that region, the convergence rate is linear.

High dimensions

- ▶ The required number of steps hardly depends on the dimension of \mathbb{R}^d . Even in \mathbb{R}^{10000} , you can usually expect the algorithm to reach high precision in half a dozen steps.
- ▶ Caveat: The individual steps can become very expensive, since we have to invert H_f in each step, which is of size $d \times d$.

Next: Constrained Optimization

So far

- ▶ If f is differentiable, we can search for local minima using gradient descent.
- ▶ If f is sufficiently nice (convex and twice differentiable), we know how to speed up the search process using Newton's method.

Constrained problems

- ▶ The numerical minimizers use the criterion $\nabla f(x) = 0$ for the minimum.
- ▶ In a constrained problem, the minimum is *not* identified by this criterion.

Next steps

We will figure out how the constrained minimum can be identified. We have to distinguish two cases:

- ▶ Problems involving only equalities as constraints (easy).
- ▶ Problems also involving inequalities (a bit more complex).

Optimization Under Constraints

Objective

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{subject to} & g(\mathbf{x}) = 0\end{array}$$

Idea

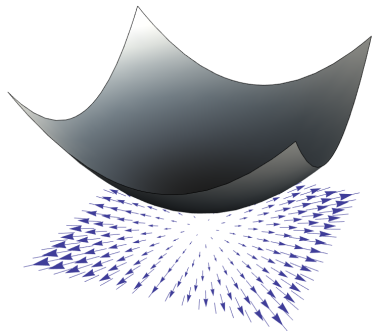
- ▶ The feasible set is the set of points \mathbf{x} which satisfy $g(\mathbf{x}) = 0$,

$$G := \{\mathbf{x} \mid g(\mathbf{x}) = 0\} .$$

If g is reasonably smooth, G is a smooth surface in \mathbb{R}^d .

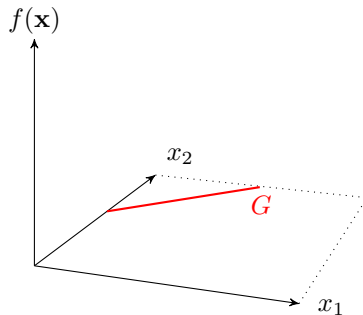
- ▶ We restrict the function f to this surface and call the restricted function f_g .
- ▶ The constrained optimization problem says that we are looking for the minimum of f_g .

Lagrange Optimization



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

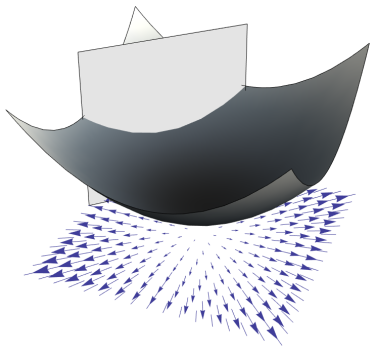
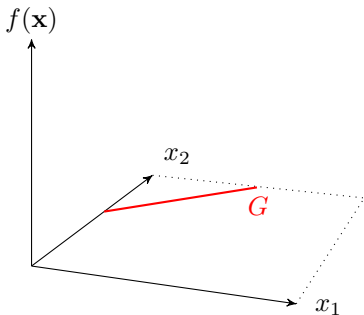
The blue arrows are the gradients $\nabla f(\mathbf{x})$
at various values of \mathbf{x} .



Constraint g .

Here, g is linear, so the graph of g is a
(sloped) affine plane. The intersection of
the plane with the x_1 - x_2 -plane is the set G
of all points \mathbf{x} with $g(\mathbf{x}) = 0$.

Lagrange Optimization



- ▶ We can make the function f_g given by the constraint $g(\mathbf{x}) = 0$ visible by placing a plane vertically through G . The graph of f_g is the intersection of the graph of f with the plane.
- ▶ Here, f_g has parabolic shape.
- ▶ The gradient of f at the minimum of f_g is *not* 0.

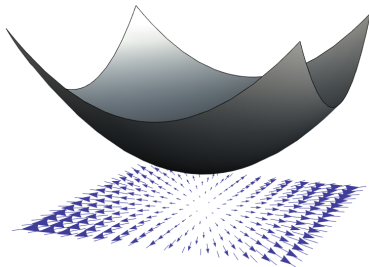
Gradients and Contours

Fact

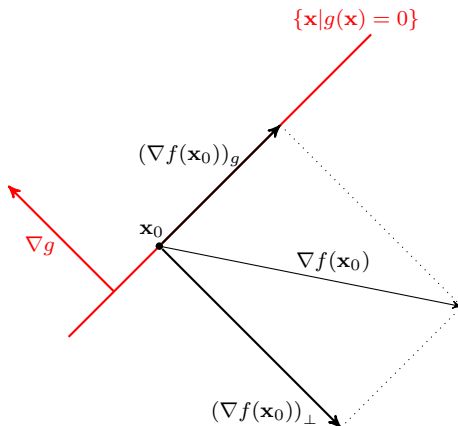
Gradients are orthogonal to contour lines.

Intuition

- ▶ The gradient points in the direction in which f grows most rapidly.
- ▶ Contour lines are sets along which f does not change.



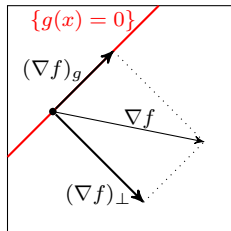
The Crucial Bit



Again, in detail.

Idea

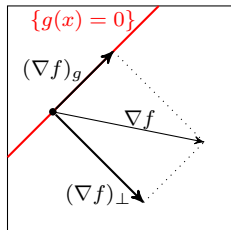
- ▶ Decompose ∇f into a component $(\nabla f)_g$ in the set $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ and a remainder $(\nabla f)_\perp$.
- ▶ The two components are orthogonal.
- ▶ If f_g is minimal within $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$, the component within the set vanishes.
- ▶ The remainder need not vanish.



Again, in detail.

Idea

- ▶ Decompose ∇f into a component $(\nabla f)_g$ in the set $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ and a remainder $(\nabla f)_\perp$.
- ▶ The two components are orthogonal.
- ▶ If f_g is minimal within $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$, the component within the set vanishes.
- ▶ The remainder need not vanish.



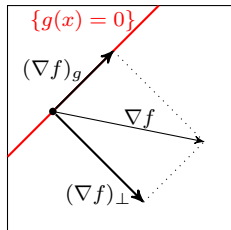
Consequence

- ▶ We need a criterion for $(\nabla f)_g = 0$.

Again, in detail.

Idea

- ▶ Decompose ∇f into a component $(\nabla f)_g$ in the set $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ and a remainder $(\nabla f)_\perp$.
- ▶ The two components are orthogonal.
- ▶ If f_g is minimal within $\{\mathbf{x} \mid g(\mathbf{x}) = 0\}$, the component within the set vanishes.
- ▶ The remainder need not vanish.



Solution

- ▶ If $(\nabla f)_g = 0$, then ∇f is orthogonal to the set $g(\mathbf{x}) = 0$.
- ▶ Since gradients are orthogonal to contours, and the set is a contour of g , ∇g is also orthogonal to the set.
- ▶ Hence: At a minimum of f_g , the two gradients point in the same direction: $\nabla f + \lambda \nabla g = 0$ for some scalar $\lambda \neq 0$.

Solution: Constrained Optimization

Solution

The constrained optimization problem

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & g(\mathbf{x}) = 0\end{array}$$

is solved by solving the equation system

$$\begin{array}{l}\nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0 \\ g(\mathbf{x}) = 0\end{array}$$

The vectors ∇f and ∇g are D -dimensional, so the system contains $D + 1$ equations for the $D + 1$ variables x_1, \dots, x_D, λ .

Inequality Constraints

Objective

For a function f and a convex function g , solve

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{subject to} & g(\mathbf{x}) \leq 0\end{array}$$

i.e. we replace $g(\mathbf{x}) = 0$ as previously by $g(\mathbf{x}) \leq 0$. This problem is called an optimization problem with **inequality constraint**.

Feasible set

We again write G for the set of all points which satisfy the constraint,

$$G := \{\mathbf{x} \mid g(\mathbf{x}) \leq 0\} .$$

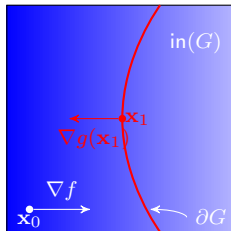
G is often called the **feasible set** (the same name is used for equality constraints).

Two Cases

Case distinction

1. The location \mathbf{x} of the minimum can be in the *interior* of G
2. \mathbf{x} may be on the *boundary* of G .

Decomposition of G



lighter shade of blue = larger value of f

$$G = \text{in}(G) \cup \partial G = \text{interior} \cup \text{boundary}$$

Note: The interior is given by $g(\mathbf{x}) < 0$, the boundary by $g(\mathbf{x}) = 0$.

Criteria for minimum

1. **In interior:** $f_g = f$ and hence $\nabla f_g = \nabla f$. We have to solve a standard optimization problem with criterion $\nabla f = 0$.
2. **On boundary:** Here, $\nabla f_g \neq \nabla f$. Since $g(\mathbf{x}) = 0$, the geometry of the problem is the same as we have discussed for equality constraints, with criterion $\nabla f = \lambda \nabla g$.

However: In this case, the sign of λ matters.

On the Boundary

Observation

- ▶ An extremum on the boundary is a minimum only if ∇f points *into* G .
- ▶ Otherwise, it is a maximum instead.

Criterion for minimum on boundary

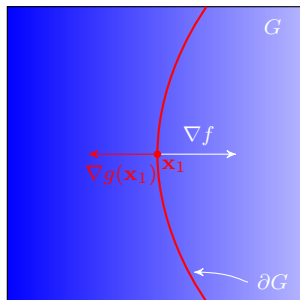
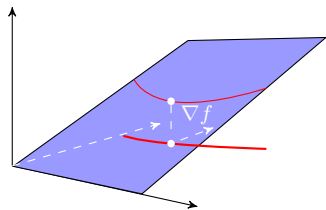
Since ∇g points *away* from G (since g increases away from G), ∇f and ∇g have to point in opposite directions:

$$\nabla f = \lambda \nabla g \quad \text{with } \lambda \leq 0$$

Convention

To make the sign of λ explicit, we constrain λ to positive values and instead write:

$$\begin{aligned} \nabla f &= -\lambda \nabla g \\ \text{s.t. } \lambda &\geq 0 \end{aligned}$$



Combining the Cases

Combined problem

$$\begin{aligned} \nabla f &= -\lambda \nabla g \\ \text{s.t. } g(\mathbf{x}) &\leq 0 \\ \lambda &= 0 \text{ if } \mathbf{x} \in \text{in}(G) \\ \lambda &> 0 \text{ if } \mathbf{x} \in \partial G \end{aligned}$$

Can we get rid of the "if $\mathbf{x} \in \cdot$ " distinction?

Yes: Note that $g(\mathbf{x}) < 0$ if \mathbf{x} in interior and $g(\mathbf{x}) = 0$ on boundary.
Hence, we always have either $\lambda = 0$ or $g(\mathbf{x}) = 0$ (and never both).

That means we can substitute

$$\begin{aligned} \lambda &= 0 \text{ if } \mathbf{x} \in \text{in}(G) \\ \lambda &> 0 \text{ if } \mathbf{x} \in \partial G \end{aligned}$$

by

$$\lambda \cdot g(\mathbf{x}) = 0 \quad \text{and} \quad \lambda \geq 0 .$$

Solution: Inequality Constraints

Combined solution

The optimization problem with inequality constraints

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{subject to} & g(\mathbf{x}) \leq 0\end{array}$$

can be solved by solving

$$\begin{array}{ll}\text{s.t.} & \left. \begin{array}{l} \nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x}) \\ \lambda g(\mathbf{x}) = 0 \\ g(\mathbf{x}) \leq 0 \\ \lambda \geq 0 \end{array} \right\} \leftarrow \begin{array}{l} \text{system of } d+1 \text{ equations for } d+1 \\ \text{variables } x_1, \dots, x_D, \lambda \end{array}\end{array}$$

These conditions are known as the **Karush-Kuhn-Tucker** (or **KKT**) conditions.

Remarks

Haven't we made the problem more difficult?

- ▶ To simplify the minimization of f for $g(\mathbf{x}) \leq 0$, we have made f more complicated and added a variable and two constraints. Well done.
- ▶ However: In the original problem, we *do not know how to minimize* f , since the usual criterion $\nabla f = 0$ does not work.
- ▶ By adding λ and additional constraints, we have reduced the problem to solving a system of equations.

Summary: Conditions

| Condition | Ensures that... | Purpose |
|--|--|--|
| $\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$ | If $\lambda = 0$: ∇f is 0 | Opt. criterion inside G |
| | If $\lambda > 0$: ∇f is anti-parallel to ∇g | Opt. criterion on boundary |
| $\lambda g(\mathbf{x}) = 0$ | $\lambda = 0$ in interior of G | Distinguish cases in(G) and ∂G |
| $\lambda \geq 0$ | ∇f cannot flip to orientation of ∇g | Optimum on ∂G is minimum |

Why Should g be Convex?

More precisely

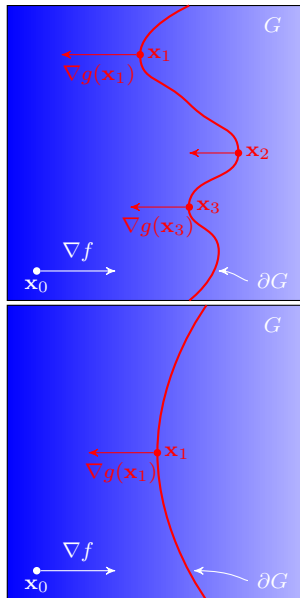
If g is a convex function, then $G = \{\mathbf{x} \mid g(\mathbf{x}) \leq 0\}$ is a convex set. Why do we require convexity of G ?

Problem

If G is not convex, the KKT conditions do not guarantee that \mathbf{x} is a minimum. (The conditions still hold, i.e. if G is not convex, they are necessary conditions, but not sufficient.)

Example (Figure)

- ▶ f is a linear function (lighter color = larger value)
- ▶ ∇f is identical everywhere
- ▶ If G is not convex, there can be several points ($\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$) which satisfy the KKT conditions. Only \mathbf{x}_1 minimizes f on G .
- ▶ G is convex, such problems cannot occur.



Interior Point Methods

Numerical methods for constrained problems

Once we have transformed our problem using Lagrange multipliers, we still have to solve a problem of the form

$$\begin{aligned} \nabla f(\mathbf{x}) &= -\lambda \nabla g(\mathbf{x}) \\ \text{s.t.} \quad \lambda g(\mathbf{x}) &= 0 \quad \text{and} \quad g(\mathbf{x}) \leq 0 \quad \text{and} \quad \lambda \geq 0 \end{aligned}$$

numerically.

Barrier functions

Idea

A constraint in the problem

$$\min f(x) \quad \text{s.t.} \quad g(x) < 0$$

can be expressed as an indicator function:

$$\min f(x) + \text{const.} \cdot \mathbb{I}_{[0,\infty)}(g(x))$$

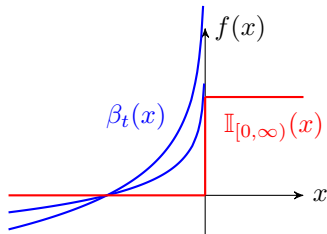
The constant must be chosen large enough to enforce the constraint.

Problem: The indicator function is piece-wise constant and not differentiable at 0. Newton or gradient descent are not applicable.

Barrier function

A **barrier function** approximates $\mathbb{I}_{[0,\infty)}$ by a smooth function, e.g.

$$\beta_t(x) := -\frac{1}{t} \log(-x) .$$



Newton for Constrained Problems

Interior point methods

We can (approximately) solve

$$\min f(x) \text{ s.t. } g_i(x) < 0 \quad \text{for } i = 1, \dots, m$$

by solving

$$\min f(x) + \sum_{i=1}^m \beta_{i,t}(x) .$$

with one barrier function $\beta_{i,t}$ for each constraint g_i .

We do not have to adjust a multiplicative constant since $\beta_t(x) \rightarrow \infty$ as $x \nearrow 0$.

Constrained problems: General solution strategy

1. Convert constraints into solvable problem using Lagrange multipliers.
2. Convert constraints of transformed problem into barrier functions.
3. Apply numerical optimization (usually Newton's method).

Recall: SVM

Original optimization problem

$$\min_{\mathbf{v}_H, c} \|\mathbf{v}_H\|_2 \quad \text{s.t.} \quad y_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 \quad \text{for } i = 1, \dots, n$$

Problem with inequality constraints $g_i(\mathbf{v}_H) \leq 0$ for $g_i(\mathbf{v}_H) := 1 - y_i(\langle \mathbf{v}_H, \tilde{\mathbf{x}}_i \rangle - c)$.

Transformed problem

If we transform the problem using Lagrange multipliers $\alpha_1, \dots, \alpha_n$, we obtain:

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad & W(\boldsymbol{\alpha}) := \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^n \tilde{y}_i \alpha_i = 0 \\ & \alpha_i \geq 0 \quad \text{for } i = 1, \dots, n \end{aligned}$$

This is precisely the "dual problem" we obtained before using geometric arguments. We can find the max-margin hyperplane using an interior point method.

Relevance in Statistics

Minimization problems

Most methods that we encounter in this class can be phrased as minimization problem. For example:

| Problem | Objective function |
|-----------------------|--------------------------------|
| ML estimation | negative log-likelihood |
| Classification | empirical risk |
| Regression | fitting or prediction error |
| Unsupervised learning | suitable cost function (later) |

More generally

The lion's share of algorithms in statistics or machine learning fall into either of two classes:

1. Optimization methods.
2. Simulation methods (e.g. Markov chain Monte Carlo algorithms).