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

$$\min f(\mathbf{x})$$
$$\text{subject to } g(\mathbf{x}) = 0$$

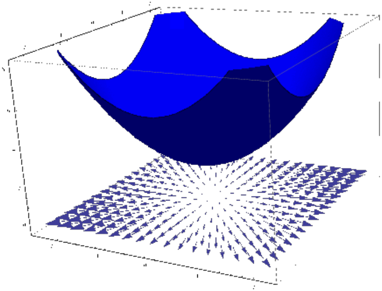
Idea

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

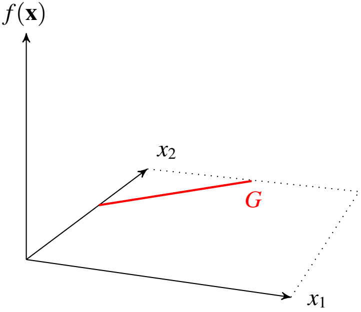
  $$G := \{\mathbf{x} \,|\, g(\mathbf{x}) = 0\} \;.$$

  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

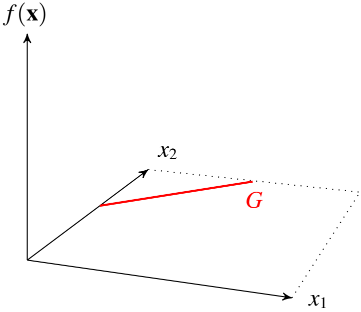


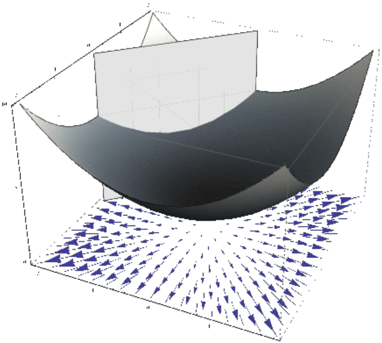
$f(\mathbf{x}) = x_1^2 + x_2^2$



Constraint g: The intersection of the plane with the x_1-x_2-plane is the set G of all points with $g(\mathbf{x}) = 0$.

Lagrange Optimization





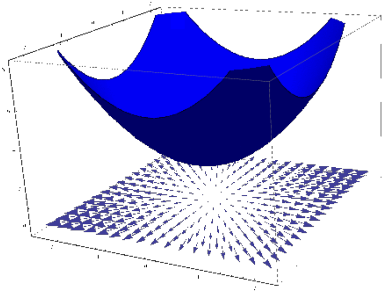
- We can make the function f_g given by the constraint $g(\mathbf{x}) = 0$ visible by placing a plane vertically through G. The graph of f_g is the intersection of the graph of f with the plane.
- Here, f_g has parabolic shape.
- The gradient of f at the miniumum of f_g is *not* 0.

Fact

Gradients are orthogonal to contour lines.

Intuition

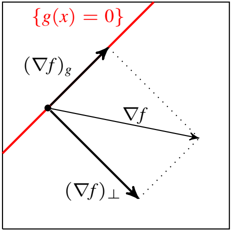
- The gradient points in the direction in which f grows most rapidly.
- Contour lines are sets along which f does not change.



THE CRUCIAL BIT

Idea

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



Consequence

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

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

$$\min_{\mathbf{x}} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad g(\mathbf{x}) = 0$$

is solved by solving the equation system

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

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

Objective

For a function f and a convex function g, solve

$$\min f(\mathbf{x})$$
$$\text{subject to } g(\mathbf{x}) \leq 0$$

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,

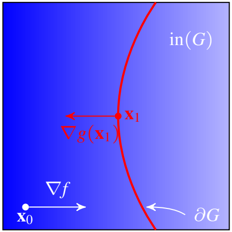
$$G := \{\mathbf{x} \,|\, g(\mathbf{x}) \leq 0\} \ .$$

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.



lighter shade of blue = larger value of f

Decomposition of G

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

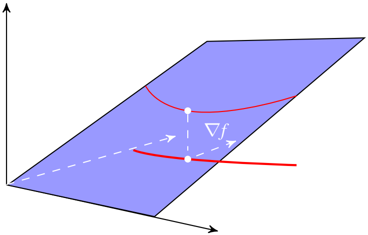
Note: The interior is given by $g(\mathbf{x}) < 0$, the boundary by $g(\mathbf{x}) = 0$.

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.

Observation

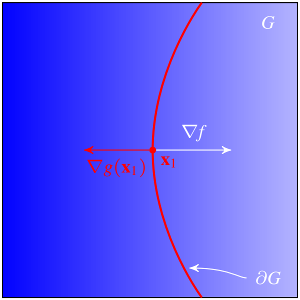
- An extremum on the boundary is a minimum onlyl if ∇f points *into* G.
- Otherwise, it is a maximum instead.



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:

$$\nabla f = \lambda \nabla g \qquad \text{with } \lambda < 0$$



Convention

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

$$\nabla f = -\lambda \nabla g$$
$$\text{s.t. } \lambda > 0$$

Combined problem

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

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

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

Combined solution

The optimization problem with inequality constraints

$$\min f(\mathbf{x})$$
$$\text{subject to } g(\mathbf{x}) \leq 0$$

can be solved by solving

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

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

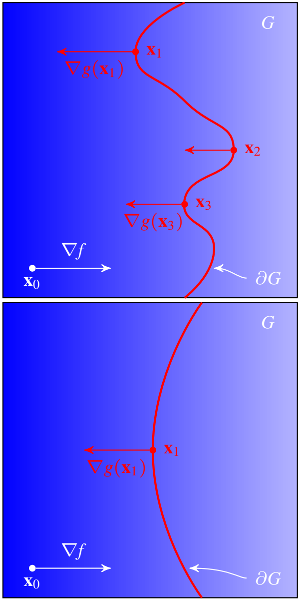
If g is a convex function, then $G = \{\mathbf{x} \mid g(\mathbf{x}) \leq 0\}$ is a convex set. Why do we require convexity of G?

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 (here: $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$) which satisfy the KKT conditions. Only $\mathbf{x}_1$ minimizes f on G.
- If G is convex, such problems cannot occur.



Numerical methods for constrained problems

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

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

numerically.

BARRIER FUNCTIONS

Idea

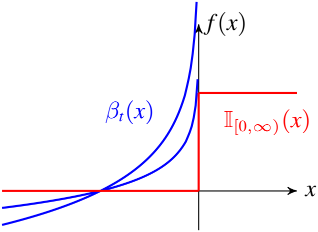
A constraint in the problem

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

can be expressed as an indicator function:

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

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

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

by solving

$$\min f(x) + \sum_{i=1}^{m} \beta_t(x) \; .$$

We do not have to adjust a multiplicative constant since $\beta_t(x) \to \infty$ 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

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

Problem with inequality constraints $g(\mathbf{v}_{\mathrm{H}}) < 0$ for $g(\mathbf{v}_{\mathrm{H}}) := 1 - y_i(\langle \mathbf{v}_{\mathrm{H}}, \tilde{\mathbf{x}}_i \rangle - c)$.

Transformed problem

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

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

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

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).

MULTIPLE CLASSES

Multiple Classes

More than two classes

For some classifiers, multiple classes are natural. We have already seen one:

- Simple classifier fitting one Gaussian per class.

We will discuss more examples soon:

- Trees.
- Ensembles: Number of classes is determined by weak learners.

Exception: All classifiers based on hyperplanes.

Linear Classifiers

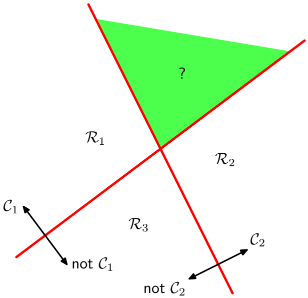
Approaches:

- One-versus-one classification.
- One-versus-all (more precisely: one-versus-the-rest) classification.
- Multiclass discriminants.

The SVM is particularly problematic.

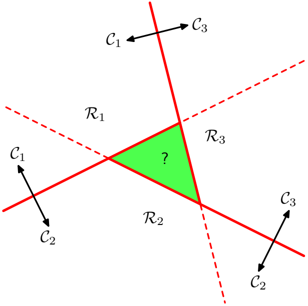
ONE-VERSUS-X CLASSIFICATION

One-versus-all



- One linear classifier per class.
- Classifies "in class k" versus "not in class k".
- Positive class $= \mathcal{C}_k$.
  Negative class $= \cup_{j \neq k} \mathcal{C}_j$.
- Problem: Ambiguous regions (green in figure).

One-versus-one



- One linear classifier for each pair of classes (i.e. $\frac{K(K-1)}{2}$ in total).
- Classify by majority vote.
- Problem again: Ambiguous regions.

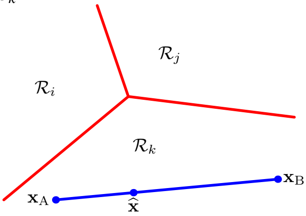
Multiclass Discriminants

Linear classifier

- Recall: Decision rule is $f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{x}, \mathbf{v}_{\mathrm{H}} \rangle - c)$
- Idea: Combine classifiers *before* computing sign. Define

$$g_k(\mathbf{x}) := \langle \mathbf{x}, \mathbf{v}_k \rangle - c_k$$

Multiclass linear discriminant



- Use one classifier g_k (as above) for each class k.
- Trained e.g. as one-against-rest.
- Classify according to

$$f(\mathbf{x}) := \arg\max_k \{g_k(\mathbf{x})\}$$

- If $g_k(\mathbf{x})$ is positive for several classes, a larger value of g_k means that $\mathbf{x}$ lies "further" into class k than into any other class j.
- If $g_k(\mathbf{x})$ is negative for all k, the maximum means we classify $\mathbf{x}$ according to the class represented by the closest hyperplane.
- Regions are convex.

SVMs and Multiple Classes

Problem

- Multiclass discriminant idea: Compare distances to hyperplanes.
- Works if the orthogonal vectors $\mathbf{v}_{\mathrm{H}}$ determining the hyperplanes are normalized.
- SVM: The K classifiers in multiple discriminant approach are trained on separate problems, so the individual lengths of $\mathbf{v}_{\mathrm{H}}$ computed by max-margin algorithm are not comparable.

Workarounds

- Often: One-against-all approaches.
- It is possible to define a single optimization problem for all classes, but training time scales quadratically in number of classes.