Notes, Numerical optimization course, WS1314

Lukas Prokop

February 20, 2014

1 Math fundamentals

$$g(\vec{x})^T \cdot \vec{s} = \vec{s}^T \cdot g(\vec{x}) \quad \neq g(\vec{x}) \cdot \vec{s}^T \qquad g, s \in \mathbb{R}^n \text{ (column vector)}$$

Matrix multiplication is associative, but not commutative.

Linear functions satisfy two conditions:

homogeneous $f(ax) = a \cdot f(x) \qquad \nabla(ab) = a \cdot \nabla(b)$

additive $f(x + y) = f(x) + f(y) \qquad \nabla(a + b) = \nabla(a) + \nabla(b)$

Linear mappings are a subset of affine mappings. The composition of linear functions is also linear. Del (∇) is a linear operator.

$$(\vec{a}\vec{b})^T = \vec{b}^T \cdot \vec{a}^T$$

$\vec{a}^T \cdot \vec{b} \cdot \vec{a} \quad \Rightarrow \quad$ if $\vec{b}$ is positive definite, the term is positive definite, $\vec{a}$ must be a vector

The inverse of a positive definite matrix is positive definite

$$\begin{array}{lrcll} \text{Conventions:} & g(x) & = & \nabla f(x) & \\ & G(x) & = & \nabla^2 f(x) & \text{(Hessian matrix)} \\ & H(x) & = & G^{-1}(x) & \end{array}$$

Schwarz integrability condition:

If $f : \mathbb{R}^N \to R$ has continuous second partial derivatives at any given point in $\mathbb{R}^n$ ($(a_1, a_2, \ldots, a_n)$) then $\forall i, j \in \mathbb{N} \setminus \{0\} : i, j \leq n$.

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(a_1, \ldots, a_n) = \frac{\partial^2 f}{\partial x_j \partial x_i}(a_1, \ldots, a_n)$$

Thus the partial derivations of this function are commutative at that point.

Universal problems ask for the *global extremum.* Compromise solutions (with multiple quality functions given and not all can be optimal at the same time) are called pareto-optimal solution.

Product rule:

$$u \cdot v \Rightarrow u' \cdot v + u \cdot v'$$

Chain rule:

$$u \circ v \Rightarrow u'(v(x)) \cdot v'(x)$$

2 Questions

Those questions are not based on real questions asked in exams. For the exam, it's important to be able to follow a train of thought. How can we derive the optimality conditions? Answer such questions by presenting one thought after another. Typically two topics have to be explained in an exam after 10 minutes preparation time. An important exam question seems to be "optimality conditions of taylor series (univariate, multivariate case)".

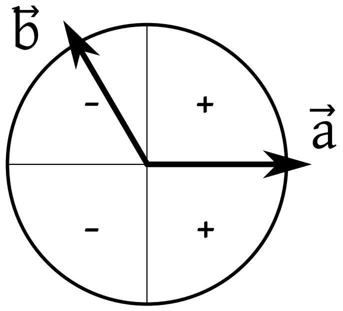

Figure 1: $\vec{a} \cdot \vec{b} < 0$ resulting from the definition of the scalar product via cosine.

2.1 What is an optimization problem?

$\vec{x}$	trial variables
$\vec{x}^*$	optimizer
$f(\vec{x})$	objective function
$c_k(\vec{x})$	constraint

$$\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$$

$$\begin{aligned} \text{s.t. } c_i(\vec{x}) &= 0 \quad \text{(equality constraint)}, \\ c_j(\vec{y}) &\geq 0 \quad \text{(inequality constraint)} \end{aligned}$$

abbr. NCP. nonlinear constraint optimization problem.
The set of all $\vec{x}$, which satisfy all constraints, is called *feasible space.*

2.2 What is the derivation of a function of an optimization problem?

$f(x)$ denotes our function. One-dimensional case:

$$f'(x) = \frac{d\,f(x)}{dx} \quad \text{"slope"}$$

$$f''(x) = \frac{d}{dx}\frac{d}{dx} f(x) \quad \text{"curvature"}$$

$f(\vec{x})$ has an parameter $\vec{x}$ with n dimensions. n-dimensional case:

$$\nabla f(\vec{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \quad \text{"gradient"}$$

$$\nabla^2 f(\vec{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix} \quad \text{“Hessian matrix”}$$

The vector *gradient* points to the direction of maximum ascent at point $\vec{x}$.

2.3 What is a forward, an analysis and a synthesis problem?

The forward and analysis problem is the problem to determine whether some behavior or function target value will be reached provided the given configuration.
The synthesis problem is the problem to find the best strategy to solve the optimization problem.

2.4 What is the directional derivative?

The directional derivative is a multivariate differentiable function at some point $\vec{x}$ and represents the rate of change.

$$\left.\frac{d\,f(\vec{x}(\alpha))}{d\alpha}\right|_{\vec{s}} = (\nabla f(\vec{x}))^T \cdot \vec{s} = \vec{s}^T \cdot \vec{g}(\vec{x})$$

2.5 What is an active set?

The set of currently considered constraints.

2.6 How can we categorize optimization problems?

We can distinguish between stochastic and deterministic methods. We desire methods with fast & stable convergence, but some methods provide slow & unstable convergence in some cases. Furthermore the number of quality functions varies: *single objective optimization* (one quality function) or *multi-objective optimization* (more than one quality function).

- deterministic
  - 0th order
    - Simplex/polytope algorithm
    - Pattern search algorithm
  - 1st order
    - Quasi-Newton method
    - Steepest Decent
    - Conjugate gradient
  - 2nd order
    - Newton method (with 2 methods for improvement, e.g. Levenberg-Marquard)
    - Gauss-Newton method
- stochastic
  - evolutionary strategies
    - $(1+1)$ evolutionary strategy
    - $(1,1)$ evolutionary strategy
    - (μ,λ) evolutionary strategy

    - $(\mu|\rho, \lambda)$ evolutionary strategy
    - $(\kappa(\mu|\rho, \lambda))$ evolutionary strategy
  - Genetic algorithms
  - simulated annealing
  - particle swarm optimization

2.7 Define the optimality conditions

2.7.1 One-dimensional function

Optimality condtions of first order:

$$\left.\frac{dy}{dx}\right|_{x^*} \overset{!}{=} 0$$

Optimality condition of second order:

$$\left.\frac{d^2y}{dx^2}\right|_{x^*} \overset{!}{>} 0$$

2.7.2 Multidimensional function without constraints

Optimality condition of first order:

$$\nabla f(\vec{x}^*) = \vec{g}(\vec{x}^*) \overset{!}{=} \vec{0}$$

Optimality condition of second order:

$$\vec{s}^T G(\vec{x}^*)\vec{s} \overset{!}{>} 0 \qquad \text{with } G(\vec{x}^*) \text{ being positive definite}$$

2.7.3 Multidimensional function with linear equality constraints

Optimality condition of first order:

$$Z^T \vec{g}^* \overset{!}{=} 0$$

Optimality condition of second order:

$$Z^T G^* Z \text{ is positive definite}$$

Equality constraints are hard (yes/no) due to its test with zero.

2.7.4 Multidimensional function with linear inequality constraints

$$A_t \cdot \vec{x}^* - \vec{b}_t \overset{!}{=} 0$$

$$Z_t^T \vec{g}^* \overset{!}{=} 0 \quad \vee \quad \vec{g}^* \overset{!}{=} A_t^T \lambda^*$$

$$\lambda_i^* > 0$$

$$Z_t^T G^* Z_t \text{ is positive definite}$$

2.8 What do taylor series look like until the second degree?

Taylor series for a 1-dimensional function:

$$x \to x + \alpha : \qquad y(x_0 + d) = y(x_0) + \left.\frac{d\,y}{dx}\right|_{x_0} \alpha + \frac{1}{2!}\left.\frac{d^2\,y}{dx^2}\right|_{x_0} \alpha^2$$

Taylor series for a 2-dimensional function:

$$\vec{x} \to \vec{x} + \alpha\vec{p} : \qquad f(\vec{x} + \alpha\vec{p}) = f(\vec{x}) + \frac{1}{1!}\left.\frac{d\,f(\vec{x}(\alpha))}{d\alpha}\right|_{\vec{p}} \alpha + \frac{1}{2!}\left.\frac{d^2\,f(\vec{x}(\alpha))}{d\alpha^2}\right|_{\vec{p}} \alpha^2$$

always with $\alpha \in [0, 1]$.

2.9 Why do we optimize functions with quadratic terms?

Because locally at an extrema, a function looks like a quadratic function (parabola).

2.10 What is a feasible move?

A feasible move is a move within the feasible space meaning that no constraints gets violated due to this move.

2.11 What does the matrix Z stand for?

The matrix Z allows us to map an arbitrary point into the feasible space. Z contains all bases of the feasible subspace and maps $\vec{p}_z$ to a feasible move $\vec{p}$. Thus all constraints are considered for the move $\vec{p}$ implicitly after a multiplication $Z^T \vec{p}_z$.

2.12 Compare stochastic and deterministic methods.

Deterministic methods:

- Same input parameters always lead to the same result.
- Parameters always lead to an improvement.
- Local behavior.
- A lot of effort required to evaluate gradient or hessian matrix.
- Small number of iterations.

Stochastic method:

- Same input parameters can lead to different results.
- Degradation of quality is welcome (to prevent premature convergence).
- In general only function value $f(\vec{x})$ is used.
- Methods are often parallel in themselves.
- Simple implementation.
- Simple integration of constraints.
- Large number of iterations.

2.13 Explain the Line search algorithm.

- Bracketing phase: Find an interval $[a, b]$ which contains $\vec{x}^{k+1}$. First define a ρ line. If area between $f(0)$ and $f(\rho)$ is a valley, find two points a and b which have slope $+\sigma f'(0)$ and $-\sigma f'(0)$ respectively.
- Sectioning phase: Find the iteration value $\vec{x}^{k+1}$ in $[a, b]$ such that $f(\vec{x}^k + \alpha \cdot \vec{s}^k) < f(\vec{x}^k)$ is satisfied. Thus section the area between a and b into equal intervals and evaluate those discrete values. Resize a and b to the interval containing the minimum. Iteratively identify the minimum.

2.14 How do evolutionary strategies work? Which ones exist?

Evolutionary strategies use 4 essential concepts:

1. Reproduction
2. Mutation
3. Competition
4. Selection

In general we distinguish between $(a + b)$ and (a, b). The first one means that children and parents can be the next parent. The second one means that only children can become the next parent.

$(1 + 1)$ **ES** We always compare two configurations. In n iterations we check whether or not a stopping criterion was reached. If number of positive mutations exceeds $\frac{1}{5}$, increase step width else decrease it. Create mutations using those modified parameters (distribute other parameters by normal distribution) which might become the next parent. If any of this mutant represents a quality increase, define this mutant as parent.

$(1, \mu)$ **ES** We mutate parent configuration μ times. $\frac{\mu}{2}$ were created with a larger step size and $\frac{\mu}{2}$ were created with a smaller one. After mutation, evaluate their quality and select the configuration which leads to the best value. Check stopping criterion and terminate when reached.

(μ, λ) **ES** Select one among μ parent configurations using the roulette wheel in one iteration. Use a greater stepsize for half of the mutations and a smaller stepsize for half of the other mutations. Evaluate their qualities and select μ best configurations out of λ children to become new parents.

$(\mu|\delta, \lambda)$ **ES** Select δ among μ parent configurations using the roulette wheel and recombine them using arithmetic crossover. Now λ unmutated children exist. Mutate each of the λ children individually. Either increase, decrease or keep constant the step sizes (with equal probability). Now select the μ best configurations out of λ children to become new parents.

$\kappa(\mu|\delta, \lambda)$ **ES** Just like $(\mu|\delta, \lambda)$ but apply clustering to parent population before recombination. Thus two parents that will be recombined originate from the same cluster with high probability. This algorithm has a more global behavior unlike $(\mu|\delta, \lambda)$.

2.15 Which steps are part of the simplex algorithm?

Mirroring We mirror some vertex with the worst function value to the opposite side.

Expansion If the new function value (after mirroring) is larger than any one before, we expand this function value to a higher distance compared to the two points before.

Contraction If polytopes are circling around the same point all the time, contract the simplex.

2.16 How does the Pattern Search algorithm work?

We distinguish between an exploration and extrapolation move. Given is a start point p_1 and some initial step width w.
You are at position p_1. In the exploration move, we look into some direction s_1. If $p_1 + s_1$ is some improvement, we continue to consider a direction orthogonal to s_1, s_2. If $p_1 + s_1 + s_2$ is an improvement over the previous two values, than take it. Otherwise try $p_1 + s_1 - s_2$ (opposite direction). If this failed again, then try the same with $-p_1$. If everything fails, go back to the last point and decrease w. In any other case you have the initial point p_1 and some movement s. Extrapolate the direction p_1 and s with length w to a new point. Start the same procedure with this new point.

2.17 How does the Newton method work?

A quadratic function is the simplest, analytical model with a defined minimum. Close to the minimum $\vec{x}^*$, $f(\vec{x})$ has a quadratic behavior. Take the current point, create the corresponding tangent and adjust a parabola to this parabola. Your next position is the minimum of this parabola. $\vec{p}$ defines the Newton direction.

$$G_k \cdot \vec{p} = -\vec{g}_k$$

2.18 Which problems do exist with the Newton method?

1. The Newton method does not converge always. $\vec{g}k^T \cdot \vec{p}$ must always be smaller than 0 such that $\vec{p}$ is a gradient descent. $\vec{p}$ is always a gradient descent if G_k is positive definite. However, even in this case $\vec{p}$ is too large and thus we will not match the minimum. We apply the line search algorithm along $\vec{p}$.

2. G_k is not positive definite. We approximate an appropriate matrix but adding identity matrices. Levenberg-Marquard method:

   $$[G_k + vI] \cdot \vec{p} = -\vec{g}k$$

   We add the negative gradient descent to the Newton direction and will get a positive definite matrix at some point.

2.19 How does the Quasi-Newton method work?

We approximate G_k such that

1. H_k is always positive definite.

2. $\vec{s}_k = -H_k \cdot \vec{g}_k$

3. linear search at $\vec{s}_k$: $\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{s}_k$

4. Update from H_k to H_{k+1}

2.20 What are uncovered topics?

- Gauss-Newton method (slides 9, page 2)
- Lagrange functions (slides 9, page 4)
- Quadratic programming (slides 9, page 6)
- Elimination method (slides 9, page 6)

- Lagrange method (slides 11, page 1)
- Linear programming with active set (slides 11, page 3)
- Quadratic programming with linear inequality constraints (slides 12, page 11)