Institute for Systems Theory and Automatic Control 2014
Universität Stuttgart cf
Prof. Dr.-Ing. C. Ebenbauer
Convex Optimization
http://www.ist.uni-stuttgart.de/education/courses/

How to implement your own primal-dual interior-point solver for quadratic programming

The goal of this document is to give a very short introduction into the concept of primal-dual interior-point methods and to provide compact and precise guidelines on how to implement your own convex optimization algorithm using these methods. In the exercise, we will focus on linearly constrained quadratic programming problems, but most of the theoretical concepts presented in the following also hold for more general convex optimization problems, see [1, Chapter 11].

1 General idea and the barrier method

In general, we may consider the following optimization problem

$$\begin{aligned} \min\ & f_0(x) && \text{(1a)} \\ \text{s.t.}\ & f_i(x) \leq 0\ ,\ i = 1, \dots, m && \text{(1b)} \\ & A_{\text{eq}}\, x = b_{\text{eq}}\ , && \text{(1c)} \end{aligned}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 0, \dots, m$ are convex and twice continuously differentiable functions and $A_{\text{eq}} \in \mathbb{R}^{p \times n}$ with $\operatorname{rank}(A_{\text{eq}}) = p < n$. We assume that the problem is strictly feasible, i.e. Slater's condition is satisfied, and that a solution x^* with $p^* = f_0(x^*)$ exists. The basic idea of interior-point optimization methods is to make use of suitable barrier functions in order to solve problem (1) by applying Newton's method to a sequence of equality constrained optimization problems (barrier method) or, respectively, to a sequence of modified KKT conditions (primal-dual interior point methods). Consider the problem

$$\begin{aligned} \min\ & f_0(x) + \mu\, \phi(x) && \text{(2a)} \\ \text{s.t.}\ & A_{\text{eq}}\, x = b_{\text{eq}}\ , && \text{(2b)} \end{aligned}$$

where $\mu \in \mathbb{R}_+$ is the weighting parameter for the so-called *logarithmic barrier* or *log barrier* defined as

$$\phi(x) = -\sum_{i=1}^{m} \log\big(- f_i(x)\big) \tag{3}$$

with $\operatorname{dom} \phi = \{x \in \mathbb{R}^n | f_i(x) < 0,\ i = 1, \dots, m\}$. Note that the new objective function is convex and twice continuously differentiable. Furthermore, independently from the choice of the parameter μ, the objective grows without bound if $f_i(x) \to 0$ for any i due to the logarithmic barrier function. Intuitively, the solution $x^*(\mu)$ of problem (2) should be a good approximation to the solution x^* of problem (1) if the barrier parameter μ is small. Indeed, we can use duality arguments, see [1, page 566], to show that

$$f_0(x^*(\mu)) - p^* \leq m\mu\ , \tag{4}$$

i.e., $x^*(\mu)$ is no more than $m\mu$-suboptimal. This shows that, by decreasing $\mu > 0$, we can obtain a strictly feasible solution $x^*(\mu)$ which approximates the optimal solution x^* arbitrarily close. However, simply choosing $\mu = \varepsilon/m$ for a given accuracy $\varepsilon > 0$ does generally not work well in practice since the Hessian of the function $f_0(x) + \mu\phi(x)$ varies rapidly near the boundary of the feasible set, which may cause numerical problems when applying Newton's method with an arbitrary initial condition. This problem can be circumvented by solving a *sequence* of problems of the form (2), in which the barrier parameter is reduced at each iteration and each Newton minimization is initialized with the solution of the last step. This procedure is called the *barrier method* and is discussed in more detail in [1, page 569ff].

Interestingly, we can interpret the barrier function based smooth approximation of the original problem also as a modification o the corresponding Karush-Kuhn-Tucker optimality conditions. By introducing the variables $\lambda_i(\mu) := -\mu/f_i(x(\mu))$, $i = 1, \ldots, m$, the KKT conditions of problem (2) can be written as

$$\begin{aligned} \nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A_{\text{eq}}^T \nu &= 0 && \text{(5a)} \\ A_{\text{eq}} x &= b_{\text{eq}} && \text{(5b)} \\ -\lambda_i f_i(x) &= \mu \ , \ i = 1, \ldots, m && \text{(5c)} \\ \lambda_i \geq 0 \ , \quad f_i(x) &\leq 0 \ , \ i = 1, \ldots, m \ . && \text{(5d)} \end{aligned}$$

Clearly, the only difference to the KKT conditions of the original problem (1) is the relaxed complementary slackness condition (5c). This shows that the barrier method yields a solution $(x^*(\mu), \lambda^*(\mu), \nu^*(\mu))$ which approximately satisfies the KKT conditions for problem (1).

2 Primal-dual interior-point methods

As outlined above, primal-dual interior point methods also rely on a barrier function based approximation of the original optimization problem and are very similar to the barrier method. However, as their name suggests, primal-dual interior-point methods compute a search direction for both the primal and dual variables by directly applying Newton's method to the modified KKT conditions (5). In general, primal-dual interior-point algorithms are based on the following steps:

2.1 compute a suitable search direction for both primal and dual variables based on the KKT conditions (5);

2.2 perform a line search to determine a step size that results in an improved and strictly feasible solution and use this step size to update the primal and dual variables;

2.3 iterate steps 1 and 2 while decreasing the barrier parameter μ until suitable stopping criteria are satisfied.

Primal-dual interior-point methods are often more efficient and usually outperform the barrier method for standard problem classes like linear, quadratic, second-order cone, and semidefinite programming [1]. In the following paragraphs, the individual steps outlined above are discussed in more detail.

2.1 Search direction for primal and dual variables

In order to find an approximate solution to the original problem (1), we want to find a primal-dual solution $(x^*(\mu), \lambda^*(\mu), \nu^*(\mu))$ which satisfies the modified KKT conditions (5). To do so, we may aim to zero the KKT residual vector

$$r_\mu(x,\lambda,\nu) = \begin{bmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{bmatrix} = \begin{bmatrix} \nabla f_0(x) + Jf(x)^T\lambda + A_{\text{eq}}^T\nu \\ -\operatorname{diag}(\lambda) f(x) - \mu\mathbb{1} \\ A_{\text{eq}}\, x - b_{eq} \end{bmatrix} \ , \tag{6}$$

related to the equality conditions (5a)-(5c) while ensuring that the inequalities $\lambda_i \geq 0$, $f_i(x) \leq 0$, $i = 1, \dots, m$ in (5d) are satisfied. Here, $f : \mathbb{R}^n \to \mathbb{R}^m$ and its derivative matrix or Jacobian $Jf \in \mathbb{R}^{m \times n}$ are defined as

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix} \ , \qquad Jf(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix} \ . \tag{7}$$

Consider now a point x, λ, ν that satisfies $f(x) \leq 0$ and $\lambda \geq 0$. Then, the Newton step for solving the nonlinear equation $r_\mu(x,\lambda,\nu) = 0$ is given by the linear equations

$$r_\mu(x + \Delta x, \lambda + \Delta\lambda, \nu + \Delta\nu) \approx r_\mu(x,\lambda,\nu) + Jr_\mu(x,\lambda,\nu) \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{bmatrix} \ . \tag{8}$$

Hence, in terms of x, λ, and ν, we get

$$\begin{bmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Jf(x)^T & A_{\text{eq}}^T \\ -\operatorname{diag}(\lambda) Jf(x) & -\operatorname{diag}(f(x)) & 0 \\ A_{\text{eq}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{bmatrix} \ . \tag{9}$$

Note that the search directions for primal and dual variables are coupled through both the left- and the right-hand side.

2.2 Backtracking line search and update

In the previous section, we have seen that suitable search directions for decreasing the KKT residual can be computed by solving a system of linear equations. However, since these search directions are based on a linearization, we have to perform a line search in order to ensure feasibility as well as an actual reduction of r_μ.

The line search in the primal-dual interior-point method is a standard backtracking line search based on the norm of the residual. However, the procedure is modified in order to ensure $f(x) < 0$ and $\lambda > 0$. Based on the current iterate (x, λ, ν), we denote the next iterate as

$$x^+ = x + s\Delta x \ , \quad \lambda^+ = \lambda + s\Delta\lambda \ , \quad \nu^+ = \nu + s\Delta\nu \ , \tag{10}$$

where $s \in [0,1]$ is the step length and Δx, $\Delta\lambda$, and $\Delta\nu$ are solutions to problem (9). Of course, we can easily compute the largest positive step length that ensures $\lambda \geq 0$ as

$$s_{\max} = \min\left\{1, \min_{i|\Delta\lambda_i<0} \{-\lambda_i/\Delta\lambda_i\}\right\} \ . \tag{11}$$

Based on this, we start the backtracking with $s = 0.99 s_{\text{max}}$ and multiply s recursively with a reduction factor $\beta \in (0, 1)$ until $f(x^+) < 0$ is satisfied. Then, we continue multiplying s by β until

$$\|r_\mu(x^+, \lambda^+, \nu^+)\|_2 \leq (1 - \alpha s)\|r_\mu(x, \lambda, \nu)\|_2 \tag{12}$$

for a user-defined scalar parameter $\alpha \geq 0$, i.e., until we have a certain decrease in the norm of the residual. A more compact pseudocode representation of this backtracking line search procedure is given in Algorithm 1. According to [1, page 613], typical choices for the backtracking parameters α and β should satisfy $\alpha \in [0.01, 0.1]$ and $\beta \in [0.3, 0.8]$.

Algorithm 1: Backtracking line search

```
Data: problem matrices, current $x$, $\lambda$, $\nu$ and search directions $\Delta x$, $\Delta\lambda$, $\Delta\nu$, current
      barrier parameter $\mu$, current residual $r_\mu(x, \lambda, \nu)$
Result: step length $s^*$ that ensures $f(x^+) < 0$, $\lambda^+ > 0$, and a certain decrease in $r_\mu$

Initialization
  set the line search parameters $\alpha$ and $\beta$ ;
  set $s = s_{\text{max}}$ according to Eq. (11) ;
  set found=false ;

Step length iteration
while found==false do
    set $s = \beta s$ and compute $x^+$, $\lambda^+$, and $\nu^+$ according to Eq.(10) ;
    compute the residual $r_\mu(x^+, \lambda^+, \nu^+)$ and the vector $f(x^+)$ ;
    if $f(x^+) < 0$ and $\|r_\mu(x^+, \lambda^+, \nu^+)\|_2 \leq (1 - \alpha s)\|r_\mu(x, \lambda, \nu)\|_2$ then found=true ;

Return the solution   $s^* = s$ ;
```

2.3 Stopping criteria and choice of the barrier parameter

Based on the search direction and a suitable step length obtained via backtracking line search, we could already design a convergent primal-dual iteration scheme. For the practical implementation, however, it is important to have stopping criteria which tell us that we are close to the optimal solution. In addition, simply multiplying μ by a scalar factor $\gamma \in (0, 1)$ at each iteration might not be the best strategy for decreasing the barrier parameter. One possibility is to make use of the scalar value $\hat{\eta}$ defined as

$$\hat{\eta} = -f(x)^T \lambda \tag{13}$$

for any x that satisfies $f(x) < 0$ and for $\lambda \geq 0$. Clearly, $\hat{\eta}$ describes the cumulated violation of the original complementary slackness conditions $-\lambda_i f_i(x) = 0$, $i = 1, \ldots, m$. Hence, if $f(x) < 0$ and $\lambda \geq 0$ are satisfied and the values of the primal residual r_{pri}, the dual residual r_{dual}, and $\hat{\eta}$ are small, this implies that the KKT conditions of the original problem are "almost" satisfied, which means that we are close to the optimal solution x^*, see also Equation (5). This allows us to define tolerances $\varepsilon_{\text{feas}}$ and ε_{opt} and terminate the primal-dual iteration as soon as

$$\|r_{\text{pri}}\|_2 \leq \varepsilon_{\text{feas}}, \quad \|r_{\text{dual}}\|_2 \leq \varepsilon_{\text{feas}}, \quad \text{and} \quad \hat{\eta} \leq \varepsilon_{\text{opt}} \ . \tag{14}$$

However, the above arguments only allow us to say that we are close to the optimal solution, but not how close we are. The following points show that we can in fact use $\hat{\eta}$ as a direct measure for optimality. Assume that the primal and dual residual are small and that $f(x) < 0$ and $\lambda \geq 0$. Then, the current iterate x, λ, ν approximately solves the KKT conditions (5) with $\mu = \hat{\mu}$ if we define $\hat{\mu} := \hat{\eta}/m$. On the other hand, we know that if x were primal feasible and λ, ν were dual feasible, i.e., if $r_{\text{pri}} = r_{\text{dual}} = 0$ as it is the case in the barrier method, then any solution to (5) is at most $m\mu$-suboptimal, see Equation (4), such that

$$f_0(x^*(\mu)) - p^* \leq \hat{\eta} \ . \tag{15}$$

This means that for small primal and dual residuals we may use $\hat{\eta} \geq f_0(x^*(\hat{\mu})) - p^*$ as a measure of optimality corresponding to the current barrier parameter $\hat{\mu} = \hat{\eta}/m$. Moreover, since Slater's condition is satisfied, we know that the duality gap has to be zero at the optimal solution, such that we can use $\hat{\eta}$ as a measure for optimality. Due to this relation, $\hat{\eta}$ is also called the *surrogate duality gap*, see [1]. The above arguments also provide us a guideline on how to choose the barrier parameter for the next iteration: instead of simply multiplying the last μ with a reduction factor $\gamma \in (0, 1)$, we set $\mu = \gamma\hat{\mu} = \gamma\,\hat{\eta}/m$ for the current $\hat{\eta}$ at each iteration. Note that for the case $r_{\text{pri}} = r_{\text{dual}} = 0$, we have that $\hat{\eta} = m\mu$ and hence $\hat{\mu} = \mu$, which shows that the barrier parameter update procedure from the barrier method is recovered for a primal and dual feasible solution.

In the following section, we consider primal-dual interior-point methods in the context of convex quadratic programming and present a more detailed description of the algorithm.

3 Application to quadratic programming

Consider a linearly constrained convex quadratic programming problem of the form

$$\min_x \ \frac{1}{2}x^T Q x + c^T x \tag{16a}$$

$$\text{s.t.} \quad Ax \leq b \ , \ A_{\text{eq}}\, x = b_{eq} \tag{16b}$$

with $Q \in \mathbb{S}^n_+$, $A \in \mathbb{R}^{m \times n}$, and $A_{\text{eq}} \in \mathbb{R}^{p \times n}$. We again assume that A_{eq} has full row rank, i.e. rank(A_{eq})$= p < n$, and that the problem has a feasible solution. Furthermore, the set $FS_{in} = \{x \in \mathbb{R}^n | Ax < b\}$ is assumed to be nonempty in order to ensure the existence of a strictly feasible starting point. For the above problem, Equation (9) becomes

$$\begin{bmatrix} Q & A^T & A_{\text{eq}}^T \\ -\text{diag}(\lambda)A & -\text{diag}(Ax - b) & 0 \\ A_{\text{eq}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} Qx + c + A^T\lambda + A_{\text{eq}}^T\nu \\ -\text{diag}(\lambda)(Ax - b) - \mu\mathbb{1} \\ A_{\text{eq}}\, x - b_{eq} \end{bmatrix} \ . \tag{17}$$

Remember that this is a system of linear equations, whose solution provides us at each iteration the primal-dual search directions Δx, $\Delta\lambda$, and $\Delta\nu$. Based on this equation and the above theoretical concepts, we can describe a primal-dual interior-point optimization algorithm for convex quadratic programming by the following pseudocode presented in Algorithm 2.

```
Algorithm 2: ipQPsolver
Data: problem matrices, initial condition $x_0$ satisfying $f_i(x_0) < 0$, initial dual
      variables $\lambda > 0$, $\nu \in \mathbb{R}^p$, factor $\gamma \in (0,1)$, tolerances $\varepsilon_{\text{feas}} > 0$, $\varepsilon_{\text{opt}} > 0$
Result: approximate optimizer $\hat{x}^*$ and solution $\hat{p}^*$, approximate optimal dual
        variables $(\hat{\lambda}^*, \hat{\nu}^*)$, surrogate duality gap $\hat{\eta}^*$ as optimality measure

Initialization
  determine the problem dimensions $n$, $m$, $p$ ;
  initialize the dual variables (if not passed to the algorithm) ;
  set found=false ;

Primal-dual iteration scheme
while found==false do
    /* 1.  check stopping criteria and update the barrier parameter  */
    compute the surrogate duality gap: $\hat{\eta} = -f(x)^T\lambda$ ;
    compute the KKT residual vector $r_\mu(x, \lambda, \nu)$ ;
    if $\|r_{pri}\|_2 \leq \varepsilon_{feas}$, $\|r_{dual}\|_2 \leq \varepsilon_{feas}$, and $\hat{\eta} \leq \varepsilon_{opt}$ then
        found=true ;
        continue ;
    else
        update the barrier weighting parameter: $\mu = \gamma\,\hat{\eta}/m$ ;
        /* 2.  compute the primal-dual search direction                */
        solve (17) to determine the primal-dual search directions $\Delta x$, $\Delta\lambda$, $\Delta\nu$ ;
        /* 3.  determine a suitable step size                          */
        determine the length $s$ of the Newton step via backtracking line search ;
        /* 4.  update the primal and dual variables                    */
        $x = x + s\Delta x$ ,   $\lambda = \lambda + s\Delta\lambda$ ,   $\nu = \nu + s\Delta\nu$;

Return the solution
  $\hat{x}^* = x$,   $\hat{p}^* = f_0(\hat{x}^*)$,   $\hat{\lambda}^* = \lambda$,   $\hat{\nu}^* = \nu$,   $\hat{\eta}^* = \hat{\eta}$;
```

Remarks:

- Passing initial values for the dual variables to the algorithm is not really necessary. You can simply set $\lambda_{0,i} = -1/f_i(x_0) \geq 0$ and $\nu_0 = 0$ in the initialization.

- In step 1, the barrier parameter is set to a factor $\gamma \in (0, 1)$ times $\hat{\eta}/m$, which is the value of μ associated with the current surrogate duality gap $\hat{\eta}$, see Section 2.3. Values of γ on the order of 0.1 usually work well in practice.

- In MATLAB, you can use `x=linsolve(A,b)` for solving a linear equation of the form $Ax = b$.

- In principle, Algorithm 2 can of course also be used to solve general nonlinear convex optimization problems. All you have to do is to use the more general Equation (9) for computing the search direction.

References

[1] Boyd, S., and Vandenberghe, L. *Convex Optimization.* Cambridge University Press, 2004.