Continuous optimization: Algorithms

Torkel A. Haufmann
SINTEF Digital

Outline

1. Recap
2. Gradient methods
3. Constrained optimization
4. Exercises

What we did last time

- We defined our continuous problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in C, \end{aligned}$$

What we did last time

- We defined our continuous problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in C, \end{aligned}$$

- Some background on convexity.

What we did last time

- We defined our continuous problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in C, \end{aligned}$$

- Some background on convexity.
- Some discussion of duality.

What we did last time

- We defined our continuous problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in C, \end{aligned}$$

- Some background on convexity.
- Some discussion of duality.
- The KKT conditions:

$$\begin{aligned} \nabla f(x^*) + \textstyle\sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) &= 0, \\ c_i(x^*) &\leq 0, \quad (i \in [m]), \\ \lambda_i^* &\geq 0, \quad (i \in [m]), \\ \lambda_i^* c_i(x^*) &= 0, \quad (i \in [m]). \end{aligned}$$

What we did last time

- We defined our continuous problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in C, \end{aligned}$$

- Some background on convexity.
- Some discussion of duality.
- The KKT conditions:

$$\begin{aligned} \nabla f(x^*) + \textstyle\sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) &= 0, \\ c_i(x^*) &\leq 0, \quad (i \in [m]), \\ \lambda_i^* &\geq 0, \quad (i \in [m]), \\ \lambda_i^* c_i(x^*) &= 0, \quad (i \in [m]). \end{aligned}$$

Today: How can we actually solve these problems?

A convex problem without strong duality

Taken from Blekherman, Parrilo, Thomas.

Let $\alpha \geq 0$. Primal-dual pair:

$$\begin{aligned} \text{minimize} \quad & \alpha X_{11} \\ \text{subject to} \quad & X_{22} = 0, \\ & X_{11} + 2X_{23} = 1, \\ & X \succeq 0, \end{aligned}$$

$$\begin{aligned} \text{maximize} \quad & y_2 \\ \text{subject to} \quad & \begin{bmatrix} y_2 & 0 & 0 \\ 0 & y_1 & y_2 \\ 0 & y_2 & 0 \end{bmatrix} \preceq \begin{bmatrix} \alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

A convex problem without strong duality

Taken from Blekherman, Parrilo, Thomas.

Let $\alpha \geq 0$. Primal-dual pair:

$$\begin{aligned} \text{minimize} \quad & \alpha X_{11} \\ \text{subject to} \quad & X_{22} = 0, \\ & X_{11} + 2X_{23} = 1, \\ & X \succeq 0, \end{aligned}$$

$$\begin{aligned} \text{maximize} \quad & y_2 \\ \text{subject to} \quad & \begin{bmatrix} y_2 & 0 & 0 \\ 0 & y_1 & y_2 \\ 0 & y_2 & 0 \end{bmatrix} \preceq \begin{bmatrix} \alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

For a primal feasible point, positive semidefiniteness means $X_{23} = 0$ and so $X_{11} = 1$, meaning $p^* = \alpha$. Dually y_2 has to be zero so $d^* = 0$, and thus

$$p^* - d^* = \alpha.$$

Support Vector Machine

An exercise from last time.
The primal:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|\beta\|^2 + \tfrac{1}{n}\sum_{i=1}^{n} w_i \\ \text{subject to} \quad & y_i(\beta^T x_i - \alpha) \geq 1 - w_i, \\ & w_i \geq 0. \end{aligned}$$

Support Vector Machine

An exercise from last time.
The primal:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|\beta\|^2 + \tfrac{1}{n}\sum_{i=1}^{n} w_i \\ \text{subject to} \quad & y_i(\beta^T x_i - \alpha) \geq 1 - w_i, \\ & w_i \geq 0. \end{aligned}$$

The dual:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \lambda_i - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^T x_j, \\ \text{subject to} \quad & \sum_{i=1}^{n} y_i \lambda_i = 0, \\ & 0 \leq \lambda_i \leq \tfrac{1}{n}. \end{aligned}$$

mybinder

Go to:

`bit.ly/stk9215-exercises`

Remember to save your work!

Line search

Back to basics:

- Given a function f,
- and an *initial point* x_0,

... find a minimum of f.

Line search

Back to basics:

- Given a function f,
- and an *initial point* x_0,

... find a minimum of f.

Core idea: Iterative approach. Generate a sequence $x_1, x_2, \ldots, x_k$ such that

$$f(x_0) > f(x_1) > f(x_2) > \ldots f(x_k).$$

If $f(x_k) \approx f(x_{k-1})$ we're done.

Line search

We're in iteration k and have a *descent direction* d_k, that is,

$$\nabla f(x_k)^T d_k < 0.$$

We will set

$$x_{k+1} = x_k + s_k d_k.$$

Question

What should we be thinking about when setting s_k?

Line search

While not optimal:

1. Choose d_k with $\nabla f(x_k)^T d_k < 0$.
2. Find $s_k \approx \arg\min_{s_k > 0} f(x_k + s_k d_k)$.
3. $x_{k+1} = x_k + s_k d_k$.

Line search

While not optimal:

1. Choose d_k with $\nabla f(x_k)^T d_k < 0$.
2. Find $s_k \approx \arg\min_{s_k > 0} f(x_k + s_k d_k)$.
3. $x_{k+1} = x_k + s_k d_k$.

Main questions:

- How do we choose d_k?
- How do we choose s_k?
- What do we do about constraints?

Step size

Let's discuss step size first.

Armijo condition: Pick $a_1 \in (0, 1)$. We should find s_k satisfying

$$f(x_k + s_k p_k) \leq f(x_k) + a_1 s_k \nabla f(x_k)^T d_k.$$

This enforces sufficient decrease.

Step size

Let's discuss step size first.

Armijo condition: Pick $a_1 \in (0,1)$. We should find s_k satisfying

$$f(x_k + s_k p_k) \leq f(x_k) + a_1 s_k \nabla f(x_k)^T d_k.$$

This enforces sufficient decrease.

Wolfe conditions: As above, plus choose $a_2 \in (a_1, 1)$ and look for

$$\nabla f(x_k + s_k d_k)^T d_k \geq a_2 \nabla f(x_k)^T d_k.$$

This is called the curvature condition: We don't want steepness where we stop.

Step size

Theorem

Assume f is continuously differentiable, and let d_k be a descent direction at x_k. If f is bounded below on $\{x_k + sd_k : s > 0\}$ and $0 < a_1 < a_2 < 1$, the Wolfe conditions are satisfied along the ray.

Step size

Theorem

Assume f is continuously differentiable, and let d_k be a descent direction at x_k. If f is bounded below on $\{x_k + sd_k : s > 0\}$ and $0 < a_1 < a_2 < 1$, the Wolfe conditions are satisfied along the ray.

Backtracking line search: Given f, x_k, d_k and a_1:

1. Set $s = 1$, $\rho \in (0, 1)$.
2. Until $f(x_k + sd_k) \leq f(x_k) + a_1 s \nabla f(x_k)^T d_k$:
   1. Set $s := \rho s$.
3. Return $s_k = s$.

Step size

Theorem

Assume f is continuously differentiable, and let d_k be a descent direction at x_k. If f is bounded below on $\{x_k + sd_k : s > 0\}$ and $0 < a_1 < a_2 < 1$, the Wolfe conditions are satisfied along the ray.

Backtracking line search: Given f, x_k, d_k and a_1:

1. Set $s = 1$, $\rho \in (0, 1)$.
2. Until $f(x_k + sd_k) \leq f(x_k) + a_1 s \nabla f(x_k)^T d_k$:
   1. Set $s := \rho s$.
3. Return $s_k = s$.

Question

Does this obviously converge?

Step size

Theorem

Assume f is continuously differentiable, and let d_k be a descent direction at x_k. If f is bounded below on $\{x_k + sd_k : s > 0\}$ and $0 < a_1 < a_2 < 1$, the Wolfe conditions are satisfied along the ray.

Backtracking line search: Given f, x_k, d_k and a_1:

1. Set $s = 1$, $\rho \in (0, 1)$.
2. Until $f(x_k + sd_k) \leq f(x_k) + a_1 s \nabla f(x_k)^T d_k$:
   1. Set $s := \rho s$.
3. Return $s_k = s$.

Question

Does this obviously converge?

The answer is yes, if we choose the descent direction "wisely".

Descent direction

We now tackle the second question: Choosing d_k.

Descent direction

We now tackle the second question: Choosing d_k.

One obvious choice: $d_k = -\nabla f(x_k)$. This is the *steepest descent* or *gradient descent* method. Warning: It can be incredibly slow.

Descent direction

We now tackle the second question: Choosing d_k.

One obvious choice: $d_k = -\nabla f(x_k)$. This is the *steepest descent* or *gradient descent* method. Warning: It can be incredibly slow.

Assume f is "smooth enough" and do a second-order Taylor expansion:

$$f(x_k + d) = f(x_k) + d^T \nabla f(x_k) + \frac{1}{2} d^T \nabla^2 f(x_k) d + \cdots,$$

where $\nabla^2 f(x_k)$ is the Hessian. Truncate and set derivative equal to 0:

$$\nabla f(x_k) + \nabla^2 f(x_k) d = 0.$$

From this we get $d = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$. This is the *Newton direction*.

Descent direction

We now tackle the second question: Choosing d_k.

One obvious choice: $d_k = -\nabla f(x_k)$. This is the *steepest descent* or *gradient descent* method. Warning: It can be incredibly slow.

Assume f is "smooth enough" and do a second-order Taylor expansion:

$$f(x_k + d) = f(x_k) + d^T \nabla f(x_k) + \frac{1}{2} d^T \nabla^2 f(x_k) d + \cdots,$$

where $\nabla^2 f(x_k)$ is the Hessian. Truncate and set derivative equal to 0:

$$\nabla f(x_k) + \nabla^2 f(x_k) d = 0.$$

From this we get $d = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$. This is the *Newton direction*.

Let's try it in binder

Descent direction

Newton's method:

- Quadratic when the step size is 1.

Descent direction

Newton's method:

- Quadratic when the step size is 1.
- Much, much faster than steepest descent.

Descent direction

Newton's method:

- Quadratic when the step size is 1.
- Much, much faster than steepest descent.
- But: $\nabla^2 f(x_k)$ is expensive.

Descent direction

Newton's method:

- Quadratic when the step size is 1.
- Much, much faster than steepest descent.
- But: $\nabla^2 f(x_k)$ is expensive.

A third option: Quasi-Newton methods approximate the Hessian. We will look briefly at L-BFGS (Broyden, Fletcher, Goldfarb, Shanno).

L-BFGS

Assume B_k is such that

$$f(x_k + d) \approx m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d,$$

which is quadratic and minimized by

$$d_k = -B_k^{-1} \nabla f(x_k).$$

In (L-)BFGS this is the step direction, and $x_{k+1} = x_k + s_k d_k$ where s_k satisfies the Wolfe conditions.

L-BFGS

Assume B_k is such that

$$f(x_k + d) \approx m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d,$$

which is quadratic and minimized by

$$d_k = -B_k^{-1} \nabla f(x_k).$$

In (L-)BFGS this is the step direction, and $x_{k+1} = x_k + s_k d_k$ where s_k satisfies the Wolfe conditions.

Now: How can we exploit B_k to make a good B_{k+1}?

L-BFGS

Idea: Let's make

$$\nabla m_{k+1}(-s_k d_k) = \nabla f(x_k), \qquad \nabla m_{k+1}(0) = \nabla f(x_{k+1}).$$

L-BFGS

Idea: Let's make

$$\nabla m_{k+1}(-s_k d_k) = \nabla f(x_k), \qquad \nabla m_{k+1}(0) = \nabla f(x_{k+1}).$$

Question

Why on earth?

L-BFGS

Idea: Let's make

$$\nabla m_{k+1}(-s_k d_k) = \nabla f(x_k), \qquad \nabla m_{k+1}(0) = \nabla f(x_{k+1}).$$

Question

Why on earth?

The second is automatic, the first gives us

$$\nabla f(x_{k+1}) - s_k B_{k+1} d_k = \nabla f(x_k).$$

If we set $v_k = x_{k+1} - x_k = s_k d_k$ and $u_k = \nabla f(x_{k+1}) - \nabla f(x_k)$:

$$B_{k+1} v_k = u_k.$$

L-BFGS

Idea: Let's make

$$\nabla m_{k+1}(-s_k d_k) = \nabla f(x_k), \qquad \nabla m_{k+1}(0) = \nabla f(x_{k+1}).$$

Question

Why on earth?

The second is automatic, the first gives us

$$\nabla f(x_{k+1}) - s_k B_{k+1} d_k = \nabla f(x_k).$$

If we set $v_k = x_{k+1} - x_k = s_k d_k$ and $u_k = \nabla f(x_{k+1}) - \nabla f(x_k)$:

$$B_{k+1} v_k = u_k.$$

This is called the *secant equation*.

L-BFGS

We can set:

$$B_{k+1} = B_k - \frac{B_k u_k u_k^T B_k^T}{u_k^T B_k u_k} + \frac{v_k v_k^T}{v_k^T u_k}.$$

L-BFGS

We can set:

$$B_{k+1} = B_k - \frac{B_k u_k u_k^T B_k^T}{u_k^T B_k u_k} + \frac{v_k v_k^T}{v_k^T u_k}.$$

Question

What about B_0?

L-BFGS

We can set:

$$B_{k+1} = B_k - \frac{B_k u_k u_k^T B_k^T}{u_k^T B_k u_k} + \frac{v_k v_k^T}{v_k^T u_k}.$$

Question

What about B_0?

To get the **L-**BFGS: We don't need B_k. In fact we need

$$B_k^{-1} \nabla f(x_k).$$

This can be stored in a cheaper way, with only "a few" vector pairs. *Useful at larger scales!* But implementation is a bit involved.

Final note: If you need box constraints, there's L-BFGS**-B**.

Subgradient Method

Recall:

Theorem

If f is differentiable and defined on a convex set C, f is convex if and only if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

for all $x, y \in C$.

Subgradient Method

Recall:

Theorem

If f is differentiable and defined on a convex set C, f is convex if and only if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

for all $x, y \in C$.

Definition

A subgradient of f at x is a vector g that satisfies

$$f(y) \geq f(x) + g^T (y - x)$$

for all $y \in C$.

In fact, if f is differentiable at x, $\nabla f(x)$ is the only subgradient at x.

Subgradient method

The *subgradient method* is like the gradient descent method, but:

Subgradient method

The *subgradient method* is like the gradient descent method, but:

1. The steps are defined by

$$x_{k+1} = x_k - s_k g_k,$$

   where g_k is a subgradient.

Subgradient method

The *subgradient method* is like the gradient descent method, but:

1. The steps are defined by

$$x_{k+1} = x_k - s_k g_k,$$

   where g_k is a subgradient.
2. The step sizes are different. Two choices:
   - Constant: $s_k = s$ for all k.
   - Nonsummable diminishing: $\lim_{k\to\infty} s_k = 0$, $\sum_{k=1}^{\infty} s_k = \infty$. For example $s_k = 1/\sqrt{k}$.

Subgradient method

The *subgradient method* is like the gradient descent method, but:

1. The steps are defined by

$$x_{k+1} = x_k - s_k g_k,$$

   where g_k is a subgradient.
2. The step sizes are different. Two choices:
   - Constant: $s_k = s$ for all k.
   - Nonsummable diminishing: $\lim_{k\to\infty} s_k = 0$, $\sum_{k=1}^{\infty} s_k = \infty$. For example $s_k = 1/\sqrt{k}$.
3. It is not a descent method.

Subgradient method

The *subgradient method* is like the gradient descent method, but:

1. The steps are defined by

$$x_{k+1} = x_k - s_k g_k,$$

   where g_k is a subgradient.
2. The step sizes are different. Two choices:
   - Constant: $s_k = s$ for all k.
   - Nonsummable diminishing: $\lim_{k\to\infty} s_k = 0$, $\sum_{k=1}^{\infty} s_k = \infty$. For example $s_k = 1/\sqrt{k}$.
3. It is not a descent method.

- Convergence can be proved for convex functions, depending on the step size.
- The method can be extended to deal with constrained optimization, with similar enough properties.

Stochastic (sub)gradient method

Just like in the subgradient descent case:

$$x_{k+1} = x_k - s_k g_k,$$

but we wish to replace g_k with something simpler. Intuition: Say (like for SVM) that

$$g_k = \frac{1}{n} \sum_{i=1}^{n} g_k^{(i)}.$$

Stochastic (sub)gradient method

Just like in the subgradient descent case:

$$x_{k+1} = x_k - s_k g_k,$$

but we wish to replace g_k with something simpler. Intuition: Say (like for SVM) that

$$g_k = \frac{1}{n} \sum_{i=1}^{n} g_k^{(i)}.$$

Now, instead of g_k we pick $g_k^{(i)}$ uniformly at random. Then the equation above tells us that the *expected gradient* is equal to g_k!

Stochastic (sub)gradient method

Just like in the subgradient descent case:

$$x_{k+1} = x_k - s_k g_k,$$

but we wish to replace g_k with something simpler. Intuition: Say (like for SVM) that

$$g_k = \frac{1}{n} \sum_{i=1}^{n} g_k^{(i)}.$$

Now, instead of g_k we pick $g_k^{(i)}$ uniformly at random. Then the equation above tells us that the *expected gradient* is equal to g_k!

Theorem

With diminishing step sizes the stochastic subgradient method converges (in expectation).

Stochastic (sub)gradient method

Just like in the subgradient descent case:

$$x_{k+1} = x_k - s_k g_k,$$

but we wish to replace g_k with something simpler. Intuition: Say (like for SVM) that

$$g_k = \frac{1}{n} \sum_{i=1}^{n} g_k^{(i)}.$$

Now, instead of g_k we pick $g_k^{(i)}$ uniformly at random. Then the equation above tells us that the *expected gradient* is equal to g_k!

Theorem

With diminishing step sizes the stochastic subgradient method converges (in expectation).

Projected gradient methods

We can also include *projection steps* in gradient methods. For any y, let

$$P_C(y) = \arg\min_{x \in C} \|x - y\|.$$

Question

Is this well defined?

Projected gradient methods

We can also include *projection steps* in gradient methods. For any y, let

$$P_C(y) = \arg\min_{x \in C} \|x - y\|.$$

Question

Is this well defined?

Now, change the iteration step to

$$x_{k+1} = P_C(x_k - s_k g_k).$$

The convergence proof (which we didn't do!) can be extended to this case.

Projected gradient methods

We can also include *projection steps* in gradient methods. For any y, let

$$P_C(y) = \arg\min_{x \in C} \|x - y\|.$$

Question

Is this well defined?

Now, change the iteration step to

$$x_{k+1} = P_C(x_k - s_k g_k).$$

The convergence proof (which we didn't do!) can be extended to this case.

Let's try it (very simply!) in binder

Newton's method with constraints

Let's bring back our ambition. Consider

$$\begin{aligned} \text{minimize} \quad & f(x), \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

Newton's method with constraints

Let's bring back our ambition. Consider

$$\begin{aligned} \text{minimize} \quad & f(x), \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

Analoguously to before we use the Taylor expansion and wish to solve

$$\begin{aligned} \text{minimize} \quad & f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_k) d, \\ \text{subject to} \quad & A(x_k + d) = b. \end{aligned}$$

Newton's method with constraints

Let's bring back our ambition. Consider

$$\begin{aligned} \text{minimize} \quad & f(x), \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

Analoguously to before we use the Taylor expansion and wish to solve

$$\begin{aligned} \text{minimize} \quad & f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_k) d, \\ \text{subject to} \quad & A(x_k + d) = b. \end{aligned}$$

Exercise

Find an expression for the solution of this!

Newton's method with constraints

We are left with the following system:

$$\begin{bmatrix} \nabla^2 f(x_k) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \nu \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ 0 \end{bmatrix}.$$

Newton's method with constraints

We are left with the following system:

$$\begin{bmatrix} \nabla^2 f(x_k) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \nu \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ 0 \end{bmatrix}.$$

Like before, this provides a descent direction and the analysis can be adapted.

An interior-point method

Newton's method for nonlinear equations: We want to solve

$$r(x) = 0,$$

where $r : \mathbb{R}^p \to \mathbb{R}^m$.

An interior-point method

Newton's method for nonlinear equations: We want to solve

$$r(x) = 0,$$

where $r : \mathbb{R}^p \to \mathbb{R}^m$.
Still iterative: At iteration k the model used is

$$m_k(d) = r(x_k) + Jr(x_k)d,$$

where Jr is the Jacobian. Then the Newton step comes from

$$Jr(x_k)d_k = -r(x_k).$$

An interior-point method

Newton's method for nonlinear equations: We want to solve

$$r(x) = 0,$$

where $r : \mathbb{R}^p \to \mathbb{R}^m$.
Still iterative: At iteration k the model used is

$$m_k(d) = r(x_k) + Jr(x_k)d,$$

where Jr is the Jacobian. Then the Newton step comes from

$$Jr(x_k)d_k = -r(x_k).$$

Exercise

Apply this method to the system $\nabla f = 0$ where f is a convex and two times smoothly differentiable function.

An interior-point method

At last, we want an algorithm for the following problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & c_i(x) \leq 0, (i \in [m]). \end{aligned}$$

An interior-point method

At last, we want an algorithm for the following problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & c_i(x) \leq 0, (i \in [m]). \end{aligned}$$

We will rewrite it as

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & c_i(x) + w_i = 0, (i \in [m]), \\ & w_i \geq 0, (i \in [m]). \end{aligned}$$

An interior-point method

At last, we want an algorithm for the following problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & c_i(x) \leq 0, (i \in [m]). \end{aligned}$$

We will rewrite it as

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & c_i(x) + w_i = 0, (i \in [m]), \\ & w_i \geq 0, (i \in [m]). \end{aligned}$$

An interior-point method

Now we will define the *barrier problem*:

$$\begin{aligned} \text{minimize} \quad & f(x) - \mu \sum_{i=1}^{m} \log w_i, \\ \text{subject to} \quad & c_i(x) + w_i = 0, (i \in [m]), \end{aligned}$$

where $\mu > 0$.

An interior-point method

Now we will define the *barrier problem*:

$$\begin{aligned} \text{minimize} \quad & f(x) - \mu \sum_{i=1}^{m} \log w_i, \\ \text{subject to} \quad & c_i(x) + w_i = 0, (i \in [m]), \end{aligned}$$

where $\mu > 0$.

Question

What did this accomplish?

Exercise

Find the KKT conditions for the barrier problem.

Exercise

Find the KKT conditions for the original problem.

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.
2. Repeat until “close enough”:

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.
2. Repeat until "close enough":
   1. Get the search direction from Newton's method on the KKT conditions.

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.
2. Repeat until "close enough":
   1. Get the search direction from Newton's method on the KKT conditions.
   2. Compute the step lengths.

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.
2. Repeat until "close enough":
   1. Get the search direction from Newton's method on the KKT conditions.
   2. Compute the step lengths.
   3. Update the variables.

An interior-point method

The KKT conditions for the barrier problem are:

$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.
2. Repeat until "close enough":
   1. Get the search direction from Newton's method on the KKT conditions.
   2. Compute the step lengths.
   3. Update the variables.
3. Update μ - go to step 2 if the problem isn't solved

An interior-point method

The KKT conditions for the barrier problem are:

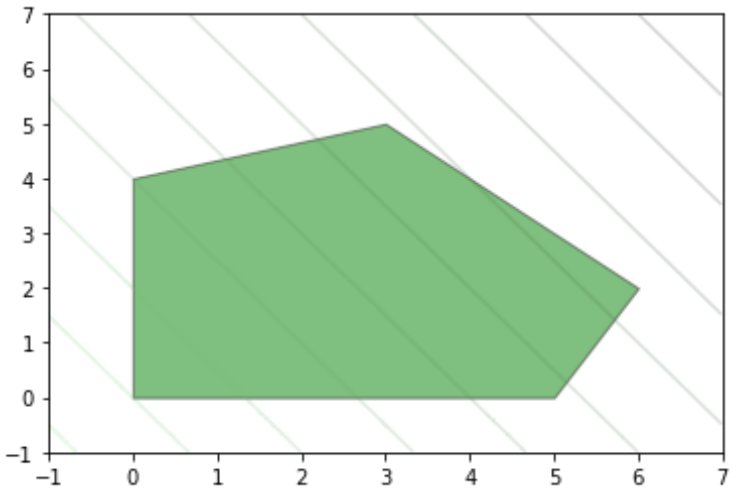
$$\begin{aligned} \nabla f(x) - Jc(x)^T \nu &= 0, \\ c(x) + w &= 0, \\ W\nu - \mu e &= 0. \end{aligned}$$

Short sketch:

1. Choose x_0, w_0 and find an initial ν. Select μ.
2. Repeat until "close enough":
   1. Get the search direction from Newton's method on the KKT conditions.
   2. Compute the step lengths.
   3. Update the variables.
3. Update μ - go to step 2 if the problem isn't solved

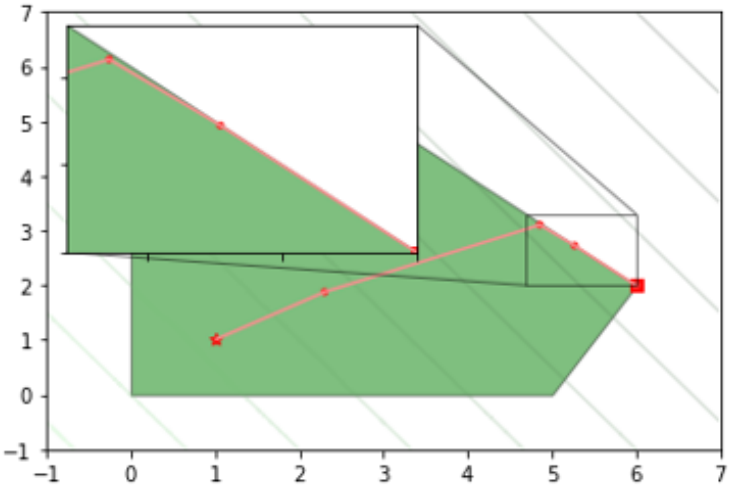
Implementation questions involve step lengths, μ "schedule", when is the inner loop close enough - when is the outer loop close enough, primal-dual reformulations, and *many more*.

Interior-point for LP: An example



$$
\begin{aligned}
\text{minimize} \quad & 3x_1 + 2x_2 \\
\text{subject to} \quad & -x_1 + 3x_2 \leq 12 \\
& x_1 + x_2 \leq 8 \\
& 2x_1 - x_2 \leq 10 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Interior-point for LP: An example



$$
\begin{aligned}
\text{minimize} \quad & 3x_1 + 2x_2 \\
\text{subject to} \quad & -x_1 + 3x_2 \leq 12 \\
& x_1 + x_2 \leq 8 \\
& 2x_1 - x_2 \leq 10 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

SVM, again

Let's try to use the interior-point machinery for solving the SVM. Specifically, we solve:

$$\text{minimize} \quad \frac{\mu}{2}\|\beta\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\beta^T \mathbf{x}_i - \alpha))$$

Let's try it in 

Brief thoughts on complexity

- Imprecise question: clearly depends on f, C, line search, etc. Example bounds:
  - Gradient descent: $f(x_k) - p^* \leq c^k(f(x_0) - p^*)$, where c is a constant that depends on the problem.

Brief thoughts on complexity

- Imprecise question: clearly depends on f, C, line search, etc. Example bounds:
  - Gradient descent: $f(x_k) - p^* \leq c^k(f(x_0) - p^*)$, where c is a constant that depends on the problem.
  - Newton's method: $f(x_k) - p^* \leq c\left(\frac{1}{2}\right)^{2^{k-1}}$, where c depends on the problem *and we start close enough*.

Brief thoughts on complexity

- Imprecise question: clearly depends on f, C, line search, etc. Example bounds:
  - Gradient descent: $f(x_k) - p^* \leq c^k(f(x_0) - p^*)$, where c is a constant that depends on the problem.
  - Newton's method: $f(x_k) - p^* \leq c\left(\frac{1}{2}\right)^{2^{k-1}}$, where c depends on the problem *and we start close enough*.
  - Stochastic subgradient method:

$$\min_{i \in [k]} \left(\mathrm{E}[f(x_i)] - f^*\right) \leq \frac{c}{2\sum_{i=1}^{k} s_k},$$

    where c depends on the problem (and the step size strategy!).
  - Interior-point harder to analyze, but BV:
    - ★ Grows with square root of inequality count,
    - ★ the radius of the feasible set,
    - ★ the gradients of the inequalities, and
    - ★ the initial gap to the optimal objective.

Exercises

Define the following quadratic function, or your favorite convex function:

$$f(x) = (x_1 - 1)^2 + (x_2 + 2)^2.$$

Exercise

Starting at x_0, minimize it with the steepest descent method.

Exercise

As above, but use Newton's method.

Exercise

(Optional!) As above, but use BFGS.

Exercise

Derive the equations that must be solved by an interior-point method for linear programming.

Exercises

Exercise

Extend the gradient descent method to also use projection.

Exercise

$$\text{minimize} \quad \frac{\mu}{2}\|\beta\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\beta^T \mathbf{x}_i - \alpha))$$

Find the subgradient! Train the SVM by using a (stochastic) subgradient method.

References

- Bertsimas, D. and Dunn, J., 2019. Machine Learning Under a Modern Optimization Lens.
- Blekherman, G., Parrilo, P. A. and Thomas, R. R., 2012. Semidefinite Optimization and Convex Algebraic Geometry.
- Boyd, S. and Vandenberghe, L., 2004. Convex Optimization.
- Nocedal, J. and Wright, S. J., 2006. Numerical Optimization - Second Edition.