STATS 230: Computational Statistics

Optimization

Babak Shahbaba

Department of Statistics, UCI

Overview

- In this lecture, we discuss convex optimization problems
- We start by some general concepts and definitions for constrained optimization
- Next, we will discuss some computational methods for solving such problems
- At the end, we will focus on the application of these methods in statistics
- For the most part, this lecture is based on the book on Convex Optimization by Boyd and Vandenberghe (2004)

Least squares regression models

- Consider the least squares problem we discussed before:

$$\text{minimize} \quad \text{RSS}(\beta) = \|y - x\beta\|^2$$

- For quadratic problems like this, we can solve the optimization problem by setting the gradient to zero

$$\begin{aligned} \nabla_\beta \text{RSS}(\beta) &= -2x^\top(y - x\hat{\beta}) = 0 \\ \hat{\beta} &= (x^\top x)^{-1}x^\top y \end{aligned}$$

assuming the Hessian is positive definite:

$$\nabla^2 \text{RSS}(\beta) = 2x^\top x \succ 0$$

which is true iff x has independent columns

Regularized regression models

- Occasionally, we would like to solve the least squares problem while controlling the complexity of the resulting model by imposing constraints on the parameters

- One possible approach is to use Bridge regression models (Frank and Friedman, 1993)

$$\begin{aligned} \text{minimize} \quad & \mathrm{RSS}(\beta) = \|y - x\beta\|^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} |\beta_j|^\gamma \leq s \end{aligned}$$

- Two important special cases are ridge regression (Hoerl and Kennard, 1970) $\gamma = 2$ and Lasso (Tibshirani, 1996) $\gamma = 1$

General optimization problems

- In general, optimization problems have the following form:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0 \qquad i = 1, \ldots, m \\ & h_i(x) = 0 \qquad j = 1, \ldots, p \end{array}$$

- We are usually interested in *convex* optimization problems, where we minimize a convex objective function $f_0(x)$ over a convex set with convex inequality constraints $f_i(x)$ and affine equality constraints $h_j(x) = Ax - b$.

Affine sets

- Affine set, C: lines through any two distinct points in C remains in C,

$$\alpha x + \beta y \in C$$

$$\text{if } x, y \in C$$

$$\alpha, \beta \in \mathbb{R}$$

$$\alpha + \beta = 1$$

Affine functions

- A function $f(x) : \mathbb{R}^n \to \mathbb{R}^m$ is an affine function if it is a sum of a linear function and a constant

$$f(x) = Ax + b$$

$$A \in \mathbb{R}^{m \times n}$$

$$b \in \mathbb{R}^m$$

Convex sets

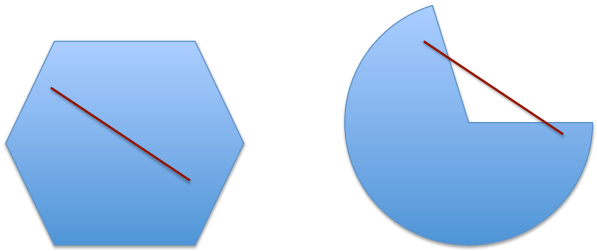
- Convex set, C: line segments between two points in C remains in C,

$$\alpha x + \beta y \in C$$

$$\text{if } x, y \in C$$

$$0 \leq \alpha, \beta \leq 1$$

$$\alpha + \beta = 1$$



Convex Set Non-convex Set

- If C is a convex set in $\mathbb{R}^n$ and $f(x) : \mathbb{R}^n \to \mathbb{R}^m$ is an affine function, then $f(C)$, i.e., the image of C under f is also a convex set.

Convex functions

- A function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain is a convex set and

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y)$$
$$\alpha \geq 0, \ \beta \geq 0, \ \alpha + \beta = 1$$

- In general, for convex function $f(E[x]) \leq E[f(x)]$ (Jensen's inequality)

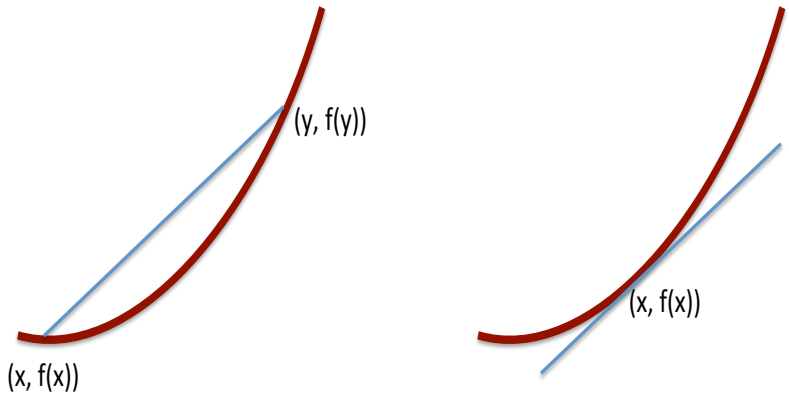
- For example, all norms are convex functions

$$\|x\|_p = (\sum_i |x_i|^p)^{1/p}, \qquad p \geq 1$$

Convex functions

- For convex functions,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x), \quad \forall x, y \in D_f$$



- Also, the Hessian is positive semidefinite $\nabla^2 f(x) \succeq 0, \quad \forall x \in D_f$

Terminology and notations

- Optimal value $p^* = \inf\{f_0(x) | f_i(x) \leq 0, h_j(x) = 0\}$
- x is feasible if $x \in D = \bigcap_{i=0}^{m} f_i \cap \bigcap_{i=1}^{p} h_i$ and satisfies the constraints
- A feasible x^* is optimal if $f_0(x^*) = p^*$
- Assuming f_0 is convex and differentiable, x is optimal iff its feasible and $\nabla f_0(x)^\top (y - x) \geq 0$, for all feasible y
- For unconstrained problems, x is optimal iff $\nabla f_0(x) = 0$

Terminology and notations

- x is locally optimal if for a given $R > 0$, it is optimal for

$$\begin{array}{lll} \text{minimize} & f_0(z) & \\ \text{subject to} & f_i(z) \leq 0 & i = 1, \ldots, m \\ & h_i(z) = 0 & j = 1, \ldots, p \\ & \|z - x\| \leq R & \end{array}$$

- In convex optimization problems, any locally optimal point is also globally optimal.

Lagrangian

- Consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0 \qquad i = 1, \ldots, m \\ & h_i(x) = 0 \qquad j = 1, \ldots, p \end{array}$$

- To take the constraints into account, we augment the objective function with a weighted sum of the constraints.

- Lagrangian: $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ defined as follows:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x)$$

  where λ and ν are dual variables or Lagrange multipliers.

- This incorporates the constraints in the objective function

Ridge regression

Original problem:

$$\begin{aligned} \text{minimize} \quad & \text{RSS}(\beta) = \|y - x\beta\|^2 \\ \text{Subject to} \quad & \|\beta\|^2 \leq s \end{aligned}$$

Lagrangian:

$$\begin{aligned} L = (y - x\beta)^\top (y - x\beta) + \lambda \beta^\top \beta & \\ \nabla_\beta L = -2x^\top (y - x\hat{\beta}) + 2\lambda\hat{\beta} &= 0 \\ -x^\top y + x^\top x\hat{\beta} + \lambda\hat{\beta} &= 0 \\ -x^\top y + (x^\top x + \lambda I)\hat{\beta} &= 0 \\ (x^\top x + \lambda I)\hat{\beta} &= x^\top y \\ \hat{\beta} &= (x^\top x + \lambda I)^{-1} x^\top y \end{aligned}$$

Lagrange dual function

- We define the Lagrange dual function as follows:

$$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu)$$

- g is concave in λ and ν since it is a pointwise infimum of a family of affine functions in terms of (λ, ν)

- If $\lambda \succeq 0$ then for each feasible point $\tilde{x}$

$$\inf_{x \in D} L(x, \lambda, \nu) = g(\lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) \leq f_0(\tilde{x})$$

- Therefore, $g(\lambda, \nu) \leq p^*$ so $g(\lambda, \nu)$ is a lower bound for the optimal value

Lagrange dual function

- We define the **Lagrange dual problem** as follows:

$$\begin{aligned} \text{maximize} \quad & g(\lambda, \nu) \\ \text{Subject to} \quad & \lambda \succeq 0 \end{aligned}$$

- Therefore, the above problem is also a convex optimization problem (i.e., minimizing $-g$)

- We denote the optimal value as d^*; the corresponding solution (λ^*, ν^*) is called the dual optimal point

- In contrast, the original problem is called the **primal problem**, whose solution x^* is called primal optimal

Weak vs. strong duality

- d^* is the best lower bound for p^*
- $d^* \leq p^*$ is called weak duality
- $p^* - d^*$ is called the optimal duality gap
- Strong duality: $d^* = p^*$

Slater's condition

- Strong duality doesn't hold in general, but if the primal is convex, it usually holds under some conditions referred to as "constraint qualifications"
- A well known constraint qualification is Slater's condition which states that we have strong duality if besides convexity we also have strict feasibility

$$f_i(x) < 0 \quad \forall i, \qquad Ax = b$$

Complementary slackness

- Consider primal optimal x^* and dual optimal (λ^*, ν^*) points
- If strong duality holds

$$\begin{aligned} f_0(x^*) &= g(\lambda^*, \nu^*) \\ &= \inf_x [f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{j=1}^p \nu_j^* h_j(x)] \\ &\leq f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{j=1}^p \nu_j^* h_j(x^*) \\ &\leq f_0(x^*) \end{aligned}$$

- Therefore, these are all equalities

Complementary slackness

- Conclusions:
  - x^* minimizes $L(x, \lambda^*, \nu^*)$
  - $\sum_{i=1}^{m} \lambda_i^* f_i(x^*) = 0$
- The latter called complementary slackness, which indicates:

$$\begin{aligned} \lambda_i^* > 0 \quad &\Rightarrow \quad f_i(x^*) = 0 \\ f_i(x^*) < 0 \quad &\Rightarrow \quad \lambda_i^* = 0 \end{aligned}$$

- In theory, we can find (λ^*, ν^*) from the dual problem (if it's easier to solve), then minimize $L(x, \lambda^*, \nu^*)$
- For many practical problems in statistics, however, we use cross-validation to choose (λ, ν)

Optimization through solving the dual problem

- When strong duality holds and a dual optimal exists, λ^*, ν^*, then any primal optimal is also a minimizer of $L(x, \lambda^*, \nu^*)$
- If the resulting solution is primal feasible then it is primal optimal
- We can use this fact to solve the optimization problem when the dual problem is easier to solve

Entropy maximization

- In information theory, the information content for a specific outcome x is defined as

$$h(X = x) = \log \frac{1}{P(X = x)}$$

- For a set of possible outcomes, $x_1, \ldots, x_n$, the entropy is defined as the expectation of information content:

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i$$

  where $p_i = P(X = x_i)$

- We choose an optimal probability model by maximizing entropy, or equivalently, minimizing negative entropy

Entropy maximization

- Primal:

$$\text{Minimize} \quad \sum_{i=1}^{n} p_i \log p_i$$

$$\text{Subject to} \quad \sum_{i=1}^{n} p_i = 1$$

For simplicity, I omitted the inequality constraints $p_i \geq 0$

- Lagrangian

$$L(p, \nu) = \sum p_i \log p_i + \nu(\sum p_i - 1)$$

- We minimize $L(p, \nu)$ by setting the gradient with respect to p to zero

$$\log \hat{p}_i + 1 + \nu = 0 \Rightarrow \hat{p}_i = \exp(-\nu - 1)$$

Entropy maximization

- Therefore, the dual function is

$$\begin{aligned} g(\nu) &= (-\nu - 1)\sum \exp(-\nu - 1) + \nu([\sum \exp(-\nu - 1)] - 1) \\ &= -n\exp(-\nu - 1) - \nu \end{aligned}$$

- Dual:

$$\text{Maximize} \quad g(\nu) = -n\exp(-\nu - 1) - \nu$$

- We find the dual optimal

$$n\exp(-\nu^* - 1) - 1 = 0$$

$$\nu^* = -1 - \log(1/n)$$

Entropy maximization

- We now minimize $L(p, \nu^*)$

$$\log p_i^* + 1 + \nu^* = 0 \Rightarrow p_i^* = 1/n$$

- Therefore, the optimal probability model is the discrete uniform distribution

- Exercise: Show that maximizing the entropy while fixing the first k moments at $m_1, \ldots, m_k$ results in a member of the exponential family of distributions

Karush-Kun-Tucker (KKT) optimality conditions

- Suppose the functions $f_0, f_1, \ldots, f_m, h_1, \ldots, h_p$ are all differentiable; also x^* and (λ^*, ν^*) primal and dual optimal points with zero duality gap

- Since x^* minimizes $L(x, \lambda^*, \nu^*)$, the gradient vanishes at x^*

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{j=1}^{p} \nu_j^* \nabla h_j(x^*) = 0$$

- Additionally

$$\begin{aligned} f_i(x^*) &\leq 0 \qquad i = 1, \ldots, m \\ h_i(x^*) &= 0 \qquad j = 1, \ldots, p \\ \lambda_i^* &\geq 0 \qquad i = 1, \ldots, m \\ \lambda_i^* f_i(x^*) &= 0 \qquad i = 1, \ldots, m \end{aligned}$$

- These are called Karush-Kun-Tucker (KKT) optimality conditions

Karush-Kun-Tucker (KKT) optimality conditions

- When the primal is convex, then KKT conditions are sufficient for the points to be primal and dual optimal with zero duality gap
- Therefore, for convex optimization problems with differentiable functions that satisfy Slater's condition, then KKT provides the necessary and sufficient conditions for optimality
- Many convex optimization problems can be expressed as methods for solving KKT conditions

Example

- Consider the following equality constrained quadratic problem:

$$\begin{array}{ll} \text{Minimize} & (1/2)x^\top Px + q^\top x + r; \quad P \succeq 0 \\ \text{Subject to} & Ax = b \end{array}$$

- KKT conditions:

$$\begin{aligned} Px^* + q + A^\top \nu^* &= 0 \\ Ax^* &= b \end{aligned}$$

- In the matrix form,

$$\begin{pmatrix} P & A^\top \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \nu^* \end{pmatrix} = \begin{pmatrix} -q \\ b \end{pmatrix}$$

- To find x^*, ν^*, we can solve the above system of $n + m$ equations

Descent methods

- We now focus on numerical solutions for unconstrained optimization problems

$$\text{Minimize} \quad f(x)$$

with twice differentiable $f : R^n \to R$

- In theory, we could find the optimal value by solving $\nabla f(x^*) = 0$; in practice however, we need iterative methods to solve such problems

- To this end, we can use descent methods that produce a minimizing sequence $x^{(k)}$,

$$f(x^{(k+1)}) < f(x^{(k)}), \qquad k = 1, \ldots$$

except when $x^{(k)}$ is optimal.

Descent methods

- We set up the sequence as

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)}, \qquad t^{(k)} > 0$$

- $\Delta x^{(k)}$ is called the "step" or "search direction"; $t^{(k)}$ is called the "step size".

- Using the first order Taylor approximation, we have

$$f(x + t\Delta x) \approx f(x) + t \nabla f(x)^\top \Delta x$$

- To be a descent method, the search direction must satisfy

$$\nabla f(x^{(k)})^\top \Delta x^{(k)} < 0$$

- Note that for convex functions,

$$f(x + t\Delta x) \geq f(x) + t \nabla f(x)^\top \Delta x$$

so the first order Taylor approximation underestimates the function

Backtracking

- To find the step size, suppose $0 < \alpha < 0.5$ and $0 < \beta < 1$, then

$$f(x + t\Delta x) \approx f(x) + t\nabla f(x)^\top \Delta x < f(x) + \alpha t \nabla f(x)^\top \Delta x$$

We can then start with a relatively large step size and decrease it until it satisfies the above condition

---

Set $\alpha \in (0, 0.5), \beta \in (0, 1)$ and $t := 1$

While $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^\top \Delta x$, Set $t := \beta t$

Gradient descent method

- A reasonable choice for the search direction is the negative gradient

$$\Delta x = -\nabla f(x)$$

- Combine with the backtracking method, we repeat these steps until a stopping criterion is satisfied

---

Set $\Delta x = -\nabla f(x)$

Set $\alpha \in (0, 0.5), \beta \in (0, 1)$ and $t := 1$

While $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^\top \Delta x$, Set $t := \beta t$

Set $x \leftarrow x + t\Delta x$

Steepest descent method

- We can write the first-order Taylor approximation as follows

$$f(x+v) \approx f(x) + \nabla f(x)^\top v$$

  where $\nabla f(x)^\top v$ is the directional derivative of f at x in the direction of v.

- Note that v is assumed to be a descent direction: $\nabla f(x)^\top v < 0$.

- Our objective is to find v, where $||v|| = 1$, such that the directional derivative is as negative as possible.

- This is called the *normalized steepest descent* direction,

$$\Delta x = \text{argmin}\{\nabla f(x)^\top v| \;\; ||v|| = 1\}$$

Steepest descent method

- The choice of the metric of course makes a difference.
- In general, given the metric P (i.e., $||v||_P = (v^\top P v)^{1/2}$), we can find the [unnormalized] steepest descent direction
- As discussed before, we use the change of variable $z = P^{1/2}x$, whose Euclidian norm, $\|z\|$ is the same as the quadratic norm $||v||_P$

$$\begin{aligned} x &= P^{-1/2}z \\ \nabla_z f &= P^{-1/2}\nabla_x f \end{aligned}$$

- In the space of z with Euclidian norm, we set $\Delta z = -\nabla_z f$ as before

$$\Delta z = -P^{-1/2}\nabla_x f$$

- For the original parameter, the corresponding step is

$$\Delta x = P^{-1/2}(-P^{-1/2}\nabla_x f) = -P^{-1}\nabla f(x)$$

Newton's method

- If we use the Euclidean metric, $P = I$, the steepest descent direction is simply the negative gradient, and the steepest descent method simply becomes the gradient descent method.

- However, if we use the Hessian metric, $P = \nabla^2 f(x)$, the steepest descent method becomes Newton's method, with the following Newton step:

$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

Newton's method

- Newton's method can also be interpreted as the second-order Taylor approximation of f at x,

$$\begin{aligned} f(x+\Delta x) &\approx f(x) + \nabla f(x)^\top \Delta x + \frac{1}{2}\Delta x^\top \nabla^2 f(x) \Delta x \\ &= \tilde{f}(x) \end{aligned}$$

- We find the optimal Δx by minimizing $\tilde{f}(x)$ with respect to Δx,

$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

The Newton decrement

- Because $\nabla^2 f(x) \succ 0$

$$\nabla f(x)^\top \Delta x = -\nabla f(x)^\top [\nabla^2 f(x)]^{-1} \nabla f(x) < 0$$

- The term

$$\lambda(x) = (\nabla f(x)^\top [\nabla^2 f(x)]^{-1} \nabla f(x))^{1/2}$$

  is called the Newton decrement, which measures the proximity of x to x^*

- By plugging Δx in the second-order Taylor approximation,

$$\begin{aligned} f(x) - \min \tilde{f}(x) &= f(x) - (f(x) - \frac{1}{2} \nabla f(x)^\top [\nabla^2 f(x)]^{-1} \nabla f(x)) \\ &= \frac{1}{2} \lambda^2(x) \end{aligned}$$

Newton's algorithm

Specify the tolerance level ε

Set $\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$

Calculate $\lambda^2 = \nabla f(x)^\top [\nabla^2 f(x)]^{-1} \nabla f(x)$

If $\lambda^2/2 \leq \varepsilon$ then quit; Otherwise

Set $\alpha \in (0, 0.5), \beta \in (0, 1)$ and $t := 1$

While $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^\top \Delta x$, Set $t := \beta t$

Set $x \leftarrow x + t\Delta x$

Quasi-Newton method

- When finding the Hessian exactly is computationally expensive, we can approximate it with another positive definite matrix $M \succ 0$ which is easier to use
- Then,

$$\Delta x = -M^{-1}\nabla f(x)$$

- One possible approach is to use a rank 1 update
- At each iteration, we find $M^{(k+1)}$ based on its previous value $M^{(k)}$

$$\begin{aligned} \Delta x &= x^{(k+1)} - x^{(k)} \\ y &= \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) \\ v &= y - M^{(k)}\Delta x \end{aligned}$$

Quasi-Newton method

- Then,

$$M^{(k+1)} = M^{(k)} + vv^\top / v^\top \Delta x$$

- Note that in general for rank 1 updates we have (matrix inversion lemma)

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}$$

- Therefore, we can find the inverse of $M^{(k+1)}$ directly from the previously computed inverse of $M^{(k)}$

- The BFGS (Broyden-Fletcher-Goldfarb-Shanno) method uses a rank 2 update

$$M^{(k+1)} = M^{(k)} + \frac{yy^\top}{y^\top \Delta x} - \frac{M^{(k)}\Delta x (M^{(k)}\Delta x)^\top}{\Delta x^\top M^{(k)} \Delta x}$$

Coordinate descent method

- For high dimensional problems, it would be easier to perform optimization one parameter at a time (Tseng, 2001)

Start with $x^{(0)} \in D_f$

At each iteration k, for $i = 1, \dots n$

$x_i^{(k)} = \text{argmin}_{x_i} f(x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_i, x_{i+1}^{(k)}, \dots, x_n^{(k)})$

Quit if $\|x^{(k+1)} - x^{(k)}\| \leq \varepsilon$

- The convergence to the optimal value x^* is guaranteed for strictly convex and differentiable functions

Coordinate descent method

- Instead of "alternating optimization" approach discussed above, we could use a gradient descent in one direction at a time

$$f(x_1, \ldots, x_n) = f_0(x_1, \ldots, x_n) + \sum_{i=1}^{n} f_i(x_i)$$

where f_i is non-differentiable but convex, and f_0 is convex and differentiable

- This condition for example holds for Lasso models

$$\text{Minimize} \|y - x\beta\|^2 + \lambda \|\beta\|_1$$

Coordinate descent method

- When f is not differentiable, the convergence to the optimal solution is not guaranteed in general, but it works if the non-differentiable part of f is separable

$$x_i^{(k+1)} = x_i^{(k)} - t_{ki} \nabla_i f(x_1^{(k+1)}, \ldots, x_i^{(k)}, \ldots, x_n^{(k)})$$

- Also, instead of one coordinate at a time, we could update a block of coordinates; this is called "block coordinate descent"

Newton's method with equality constraints

- We can use a modified version of Newton's method for optimization problems with equality constraints:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

- We assume the the initial point is feasible: $x^{(0)} \in D_f, Ax^{(0)} = b$.

- Our goal is to modify the Newton step, Δx, by taking the equality constraints into account so the subsequent points are also feasible.

- For this, we need to have $A\Delta x = 0$ so at each step $A(x^{(k)} + \Delta x) = b$.

Newton's method with equality constraints

- To find Δx, we can replace the objective function with its second-order taylor approximation at a given point x and rewrite the optimization problem as follows:

$$\begin{array}{ll} \text{minimize} & f(x+v) = f(x) + \nabla f(x)^\top v + \frac{1}{2} v^\top \nabla^2 f(x) v \\ \text{subject to} & Av = 0 \end{array}$$

- The modified Newton step, Δx, is the solution for this optimization problem.

- This is of course an equality constrained quadratic problem (discussed above), for which we can use the KKT conditions to rewrite the optimization problem as a system of linear equations

$$\begin{pmatrix} \nabla^2 f(x) & A^\top \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \omega \end{pmatrix} = \begin{pmatrix} -\nabla f(x) \\ 0 \end{pmatrix}$$

Newton's method with equality constraints

- Here we use ω to denote the optimal dual variable for the approximate quadratic problem.
- If the original objective function, $f(x)$, is quadratic, then ω is the same as ν, the optimal dual variable for the original problem.
- After we find Δx, we follow the same Newton algorithm we discussed for the unconstrained prblem.

Newton's method with inequality constraints

- We now discuss Newton's method for convex optimization problems with inequality constraints:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0 \qquad i = 1, \ldots, m \\ & Ax = b \end{array}$$

- We assume that Slater's constraint qualification holds: there exists $x \in D$ for which $Ax = b$ and $f_i(x) < 0, i = 1, \ldots, m$.

- Therefore, the primal and dual optimal points satisfy the KKT conditions.

Newton's method with inequality constraints

- For such problems, we can use Newton's methods based on *interior-point* algorithms.
- More specifically, we focus on a specific interior-point algorithm called the *barrier method*.
- In this approach, we bring the inequality constraints into the objective junction and rewrite the optimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & f_0(x) + \sum_{i=1}^{m} I_-(f_i(x)) \\ \text{subject to} \quad & Ax = b \end{aligned}$$

where the indicator function is defined as

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases}$$

Newton's method with inequality constraints

- This creates a barrier at 0 such that violating the inequality constraints adds an infinite cost to the objective function.
- The problem however is that the resulting objective function is not differentiable so we cannot use Newton's method.
- To solve this issue, we can use an approximate logarithmic barrier, which is smoother and differentiable, instead:

$$\hat{I}_-(u) = -\frac{1}{t}\log(-u)$$

where $t > 0$ controls the accuracy of the approximation: higher t, better approximation.

Newton's method with inequality constraints

- We can now rewrite our optimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & f_0(x) + \sum_{i=1}^{m} -\frac{1}{t}\log(-f_i(x)) \\ \text{subject to} \quad & Ax = b \end{aligned}$$

- Or alternatively,

$$\begin{aligned} \text{minimize} \quad & tf_0(x) + \phi(x) \\ \text{subject to} \quad & Ax = b \end{aligned}$$

  where $\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x))$.

- We can now use Newton's method for this optimization problem as discussed before.

Newton's method with inequality constraints

- The solution we find is of course for an approximation of the original problem.
- We can of course use very large t values to improve the quality of approximation, but this would lead to computational problems when we approach the boundary of the feasible set.
- To address this issue, we can start with a small value for t and gradually increase it to improve the approximation accuracy.
- This way, we will find a sequence of solutions $x^*(t)$ for $t > 0$. This is called the *central path* for the original optimization problem.
- At each iteration k, we initialize Newton's method at $x^*(t_{k-1})$ to find the new *central point* $x^*(t_k)$, where $t_k > t_{k-1}$.

Optimization methods in statistics

- In the frequentist framework, we typically perform statistical inference by maximizing log-likelihood $\ell(\theta)$, or alternatively minimizing negative log-likelihood, which is also known as the energy function
- Additionally, we have
  - Score function: $s(\theta) = \nabla_\theta \ell(\theta)$
  - Observed Fisher information: $J(\theta) = -\nabla^2_\theta \ell(\theta)$
  - Fisher information: $I(\theta) = E[-\nabla^2_\theta \ell(\theta)]$
- The step in Newton's method is

$$\Delta\theta = [J(\theta)]^{-1} s(\theta)$$

- That is, in iteration k,

$$\theta^{(k+1)} = \theta^{(k)} + [J(\theta^{(k)})]^{-1} s(\theta^{(k)})$$

Fisher scoring algorithm

- If instead of the observed information, we use the Fisher information (i.e., expectation of the observed information), the resulting method is called the *Fisher scoring* algorithm

$$\Delta\theta = [I(\theta)]^{-1} s(\theta)$$

- That is,

$$\theta^{(k+1)} = \theta^{(k)} + [I(\theta^{(k)})]^{-1} s(\theta^{(k)})$$

- In general, the Fisher scoring algorithm tends to be less sensitive to the initial guess. On the other hand, the Newton's method tends to converge faster

- For exponential family models with natural parameters and GLM with canonical links, the two methods are identical

Exponential family and GLM

- Recall that the single parameter exponential family has the following general form:

$$P(y|\mu) = \exp\{g(\mu)t(y) + c(\mu) + h(y)\}$$

- For models with multiple parameters, we have

$$P(y|\mu) = \exp\{\sum_{k=1}^{K} g_k(\mu)t_k(y) + c(\mu) + h(y)\}$$

where g, t, c, and h are vectors.

- We can change the parameter using the transformation $\phi_k = g_k(\mu)$, and write the distribution in terms of natural parameter ϕ:

$$P(y|\mu) = \exp\{\sum_{k=1}^{K} \phi_k t_k(y) + c^*(\phi) + h(y)\}$$

Poisson model

- Consider the following Poisson model:

$$\begin{aligned} P(y_i|\mu) &= e^{-\mu_i}\mu_i^{y_i}/y_i! \\ &= \exp\{\log(\mu_i)y_i - \mu_i - \log(y_i!)\} \end{aligned}$$

where $\phi_i = g(\mu_i) = \log(\mu_i)$, $t(y_i) = y_i$, $c(\mu_i) = -\mu_i$, and $h(y_i) = -\log(y_i!)$.

- We have

$$\begin{aligned} \phi_i = \log(\mu_i) \quad &\Rightarrow \quad \mu_i = \exp(\phi_i) \\ c^*(\phi_i) = \exp(\phi_i) & \\ E_{\phi_i}[t(y_i)] = E(y_i) \quad &= \quad -\frac{\partial c^*(\phi_i)}{\partial \phi_i} = \exp(\phi_i) = \mu_i \\ var_{\phi_i}[t(y_i)] = var(y_i) \quad &= \quad -\frac{\partial^2 c^*(\phi_i)}{\partial \phi_i^2} = \exp(\phi_i) = \mu_i \end{aligned}$$

Poisson model

- The score function with respect to ϕ_i can be obtained as follows:

$$\begin{aligned} s(\phi_i) &= \frac{\partial \ell(\phi_i)}{\partial \phi_i} \\ &= t(y_i) + \frac{\partial c^*(\phi_i)}{\partial \phi_i} \\ &= y_i - \exp(\phi_i) \\ &= y_i - \mu_i \end{aligned}$$

- The total score function based on n observations is

$$s(\phi) = \sum_i y_i - \exp(\phi_i) = \sum_i y_i - \mu_i$$

- As the result, the likelihood equation is:

$$\sum_i y_i - \exp(\hat{\phi}_i) = \sum_i y_i - \hat{\mu}_i = 0$$

Poisson model

- For Poisson regression model, we are of course interested in regression parameters β.
- Therefore, we would like to write the score function in terms of β.
- To do this, we first need to specify the link function.
- Suppose we use the log link function

$$g(\mu_i) \quad = \quad \log(\mu_i) = x_i\beta$$

- Since we have $\phi_i = g(\mu_i)$, we can write the link function as follows:

$$\phi_i \quad = \quad \log(\mu_i) = x_i\beta$$

- The link function that transforms the mean to the natural parameter is referred to as the *canonical link*.

Poisson model

- Using the link function, we can now write the score function in terms of β.
- For the j^{th} element of β, we have

$$\begin{aligned} s(\beta_j) &= \sum_i \frac{\partial \ell(\beta)}{\partial \beta_j} \\ &= \sum_i \frac{\partial \ell(\phi)}{\partial \phi_i} \frac{\partial \phi_i}{\partial \beta_j} \\ &= \sum_i [y_i - \exp(x_i \beta)] x_{ij} \end{aligned}$$

- As the result, the likelihood equation in terms of β_j is

$$\sum_i [y_i - \exp(x_i \hat{\beta})] x_{ij} = 0$$

Poisson model

- We can now easily obtain the Fisher information matrix in terms of β.

$$\begin{aligned} I(\beta_j \beta_k) &= E[-\frac{\partial^2 \ell(\beta)}{\partial \beta_j \partial \beta_k}] \\ &= E[\sum_i x_{ij} x_{ik} \exp(x_i \beta)] \\ &= \sum_i x_{ij} x_{ik} \exp(x_i \beta) \end{aligned}$$

- In a matrix format

$$I(\beta) = x^\top W x$$

where W is a diagonal matrix whose i^{th} element is $\exp(x_i \beta)$.

GLM

- In general, we can show that for the jth parameter

$$s(\beta_j) = \sum_i \frac{[y_i - \mu_i]x_{ij}}{var(y_i)} \frac{\partial \mu_i}{\partial \eta_i}$$

where $\partial \mu_i / \partial \eta_i$ depends on the link function we choose

- It is also easy to show that for a general link function, the Fisher information matrix is

$$\begin{aligned} I(\beta_j, \beta_k) &= E\Big(- \frac{\partial^2 \ell(\beta)}{\partial \beta_j \partial \beta_k} \Big) \\ &= \sum_i \frac{x_{ij} x_{ik}}{var(y_i)} \Big(\frac{\partial \mu_i}{\partial \eta_i} \Big)^2 \\ I(\beta) &= x^\top W x \\ W_{ii} &= \frac{\big(\frac{\partial \mu_i}{\partial \eta_i} \big)^2}{var(y_i)} \end{aligned}$$

Iterative re-weighted least squares

- For GLM, Fisher scoring is related to the weighted least squares method (e.g., linear regression with non-constant variance for error terms)

- We can write the Fisher scoring algorithm for updating β as

$$I(\beta^{(k)})\beta^{(k+1)} = I(\beta^{(k)})\beta^{(k)} + s(\beta^{(k)})$$

- since $I(\beta) = x^\top W x$,

$$(x^\top W^{(k)} x)\beta^{(k+1)} = (x^\top W^{(k)} x)\beta^{(k)} + s(\beta^{(k)})$$

- After few simple steps, we have

$$(x^\top W^{(k)} x)\beta^{(k+1)} = x^\top W^{(k)} z^{(k)}$$

where

$$z_i^{(k)} = \eta_i^{(k)} + (y_i - \mu_i^{(k)})\frac{\partial \eta_i^{(k)}}{\partial \mu_i^{(k)}}$$

Iterative re-weighted least squares

- At each iteration, we can find the next estimate for β as follows:

$$\beta^{(k+1)} = (x^\top W^{(k)} x)^{-1} x^\top W^{(k)} z^{(k)}$$

- The above estimate is similar to the weighted least squares estimate. In this case, $W^{(k)}$ is a diagonal matrix whose i^{th} element is

$$W_{ii}^{(k)} = \frac{\left(\frac{\partial \mu_i^{(k)}}{\partial \eta_i^{(k)}}\right)^2}{var(y_i)}$$

- Note that for GLM, the weights W and the response variable z change from one iteration to another based on the current estimate of β.

- We iteratively estimate β until the algorithm converges.