Quadratic Programming

Outline

- Linearly constrained minimization
  - Linear equality constraints
  - Linear inequality constraints
- Quadratic objective function

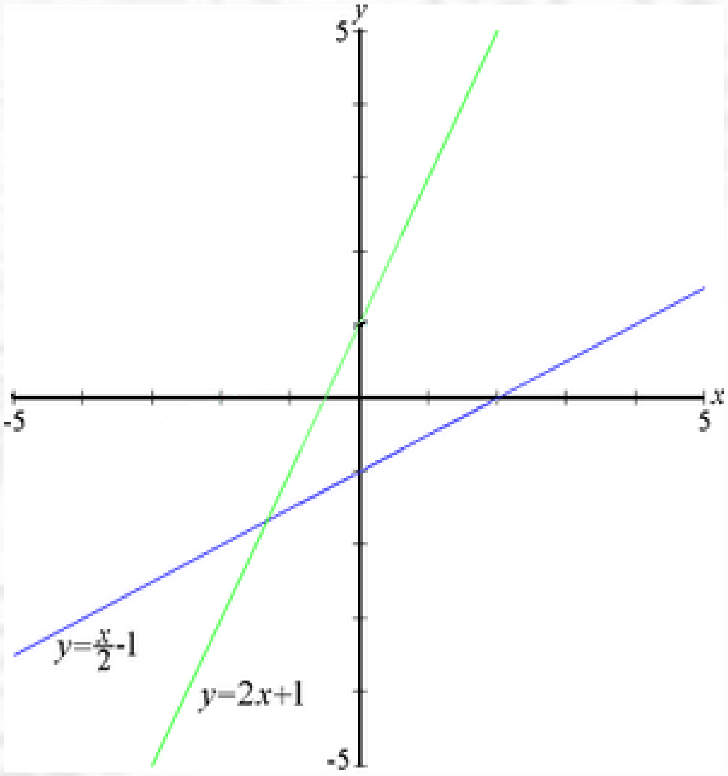
SideBar: Matrix Spaces

- Four fundamental subspaces of a matrix
  - Column space, col(A)
  - Row space, row(A)
  - Null space Ax=0, null(A)
  - Left Null space $x^T A=0$, lnull(A)
  - Rank =dim(col(A))=dim(row(A))
  - Dim(col(A))+Dim(lnull(A)) = # column
  - col(A) ad lnull(A) are orthogonal
  - Dim(row(A))+Dim(null(A)) =# row
  - row(A) and null(A) are orthogonal

Linear Equality Constraints

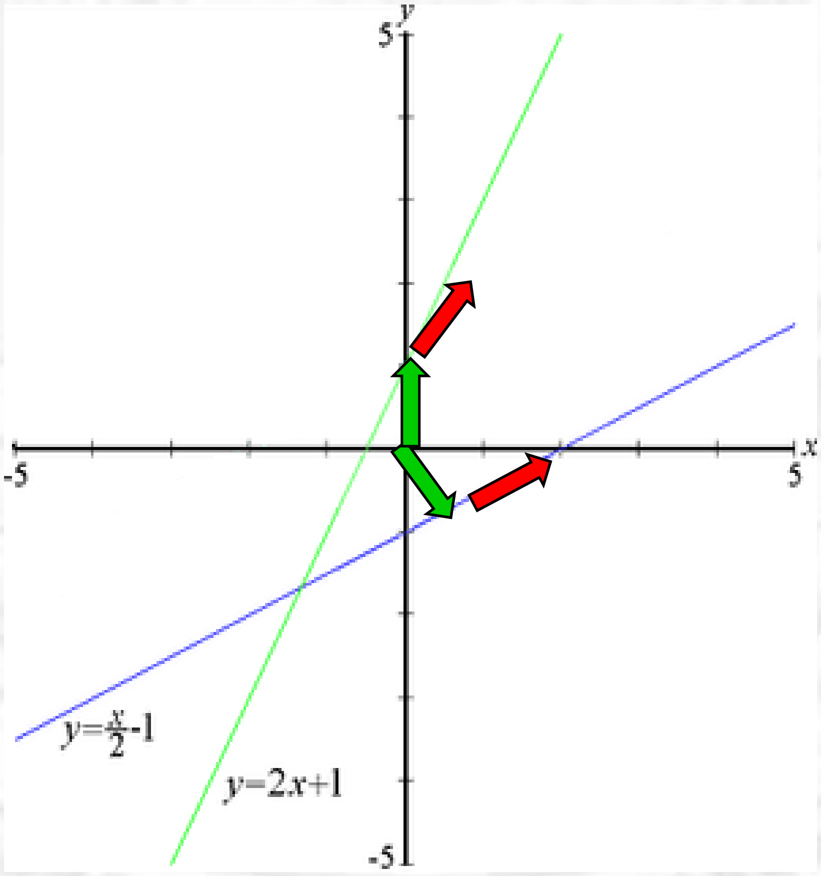
- $\min_x F(x)$
  - s.t. $Ax = b$
- Assume constraints are consistent and linearly independent
- t contraints remove t degrees of freedom solution x
- $x = A^T x_a + Z x_z$

Row space Null space



Graphical Interpretation

- $x = A^T x_a + Z x_z$
  - $A^T x_a$: a particular solution (AX=b)
  - $Z x_z$: a homogeneous solution (AX=0)



Feasible Search Directions

- Feasible *points* x_1, x_2 have $Ax_1 = Ax_2 = b$
- Feasible *step* p satisfies $Ap = A(x_1 - x_2) = 0$
- If Z is a basis for null(A), feasible directions p are such that $p = Zp_z$
- I.e., direction of change (p) should be in the null space of A
  - $Ap=0$
  - $Ax_2=A(x_1+p) = Ax_1=b$

Optimality Conditions

- Taylor series expansion along feasible direction
  - $F(x + \epsilon Z p_z) = F(x) + \epsilon p_z^T Z^T g(x) + \frac{1}{2}\epsilon^2 p_z^T Z^T G(x + \epsilon \Theta Z p_z) Z p_z$
- g is the gradient $[f_1, f_2, \ldots, f_n]^T$
- $\epsilon p_z^T Z^T g(x) =$ feasible direction * gradient = change
- Projected gradient $p_z^T Z^T g(x) = 0$ for all p_z at constrained stationary points
- Therefore, $Z^T g(x) = 0$ is first-order optimality condition
- This implies that
  - $g(x) \in \text{null}(Z^T)$
  - g(x) must in row(A)
  - so $g(x) = A^T \lambda$ at local minimum

$$\begin{bmatrix} - & z_1^T & - \\ \cdots & \cdots & \cdots \\ - & z_k^T & - \end{bmatrix} g(x) = 0$$

- Gradient direction is orthogonal to the feasible direction
- Change is zero or local landscape is flat (extreme or saddle point)

Optimality Conditions

- First-order condition necessary but not sufficient; only guarantees critical point
- Second order condition: projected Hessian G is positive semi-definite
- Positive semi-definite G guarantees weak minimum

Summary

- Necessary conditions for constrained minumum:
  - $Ax = b$
  - $Z^T g(x) = 0$
  - $Z^T G(x) Z$ positive semi-definite

Algorithm

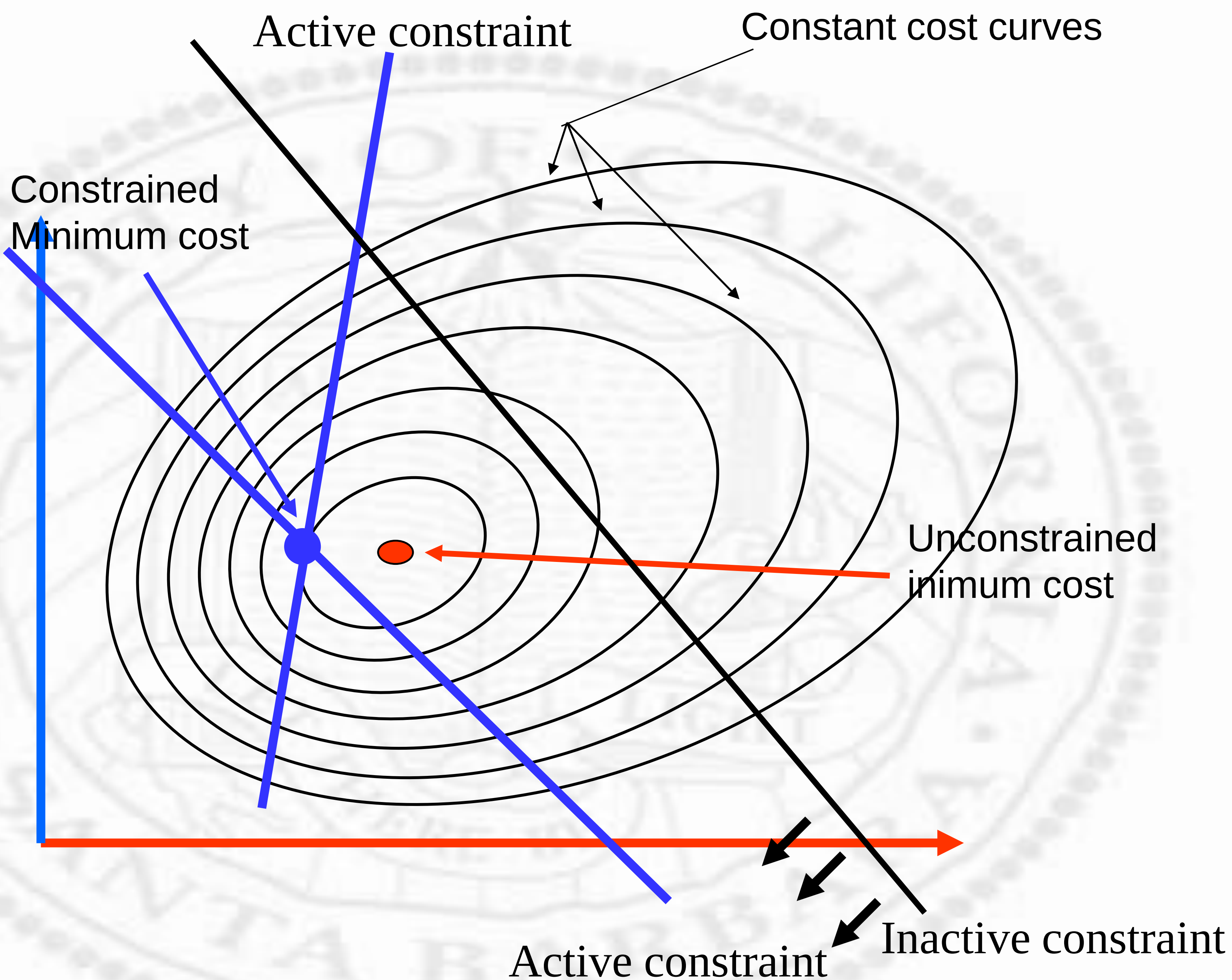
- Step 1: If conditions satisfied, terminate
- Step 2: Compute feasible search direction
- Step 3: Compute step length
- Step 4: Update estimate of minimum
- Search direction computed by Newton's Method:
  - $F(x + \epsilon Z p_z) = F(x) + \epsilon p_z^T Z^T g(x) + \frac{1}{2}\epsilon^2 p_z^T Z^T G(x + \epsilon \Theta Z p_z) Z p_z$
  - $F(x + \epsilon Z p_z)' = 0$ (derivative with respect to p_z)
  - $Z^T g + Z^T G Z p_z = 0$
  - solve $Z^T G Z p_z = -Z^T g$ for p_z and set $p = Z p_z$
  - Cf. $g + H(f)\, p = 0$ (for 1D case), This says that 1D condition is true along the direction p

Linear Inequality Constraints

- $\min_x F(x)$
  - s.t. $Ax <= b$
  - Each row $a^T x <= b$ is a half plane

$$\begin{bmatrix} a_1^T \\ a_2^T \\ a_k^T \end{bmatrix} x \leq \begin{bmatrix} b_1 \\ b_2 \\ b_k \end{bmatrix}$$

- Active constraint: $a^T x = b$
- Inactive constraint: $a^T x < b$
- If set of active constraints at solution was known, could convert to equality constraints
- Active Set Methods: maintain current active constraints, use equality constraint methods
- KKT condition applies here – those inactive constraints have lambda of zero


Active constraint
Constant cost curves
Constrained
Minimum cost
Unconstrained
inimum cost
Inactive constraint
Active constraint

Feasible Search Directions

- Recall that, before a new search
  - $a^T x = b$ (active) or $a^T x < b$ (inactive, don't care)
- Feasible search must not invalid these constraints
- Concentrate on the active set
- *Binding* perturbation: $a^T p = 0$; constraint remains active ($a^T (x+tp) = a^T x = b$)
- *Non-binding* perturbation: $a^T p < 0$; constraint becomes inactive ($a^T (x+tp) = a^T x + t\, a^T p = b + t\, a^T p < b$)

Optimality Conditions

- First and second order conditions from linear equality case apply for binding perturbations
- Added condition: $g(x)^T p <= 0$ for all non-binding perturbations p satisfying $Ap <= 0$
  - $g(x)^T p \; != 0$ means gradient * direction = change
  - If $g(x)^T p > 0$, then some constraints will be violated (because we start with $Ax=b$)
- Since $g(x) = A^T\lambda$, $g(x)^T p <= 0$ implies $\lambda Ap <= 0$
- This holds only if all $\lambda \geq 0$
  - Because, If $\lambda_j < 0$, choose p such that $(a_j)^T p = 1$, $(a_i)^T p = 0$, then:
  - $g(x)^T p = \lambda_j (a_j)^T p = \lambda_j < 0$

Summary

- Necessary conditions for constrained minimum:
  - $Ax = b$
  - $Z^T g(x) = 0$
  - $Z^T G(x) Z$ positive semi-definite
  - $\lambda_i \geq 0, i = 1,\ldots,t$

Algorithm

- Step 1: If conditions satisfied, terminate
- Step 2: Decide if a constraint should be deleted from working set; if so, go to step 6
- Step 3: Compute feasible search direction
- Step 4: Compute step length
- Step 5: Add a constraint to working set if necessary, go to step 7
- Step 6: Delete a constraint from the working set and update Z
- Step 7: Update estimate of minimum

Computing Search Direction

- Newton's method computes feasible step with respect to currently active constraints
- Need to check if $a^T p < 0$ for any inactive constraints
- Find intersection $x + \alpha p$ to closest constraint
- Line search between x and $x + \alpha p$ determines optimal step ϵp
- If $\epsilon = \alpha$, new constraint added to working set

Quadratic Programming

- Simplifications possible when using quadratic objective function
- Hessian becomes constant matrix
- Newton's method becomes exact rather than approximate

Quadratic Programming

- Newton method finds minimum in 1 iteration
- Line search not needed; either take full step, or shorten to nearest constraint
- Constant Hessian need not be evaluated at each iteration

Quadratic Programming

- Special factorization updates can be applied
- Example: Cholesky factor of G is updated by a single column when a constraint deleted
- Decomposition need only be done once at the beginning of execution