Unconstrained Optimization Algorithms

General Ideas

- Select step size s_k and direction v_k such that $f_0(x_k + s_k v_k)$ is substantially smaller than $f_0(x_k)$.
- v_k is usually selected such that $\nabla f_0(x_k)^\top v_k < 0$
- First order descent

Armijo backtracking step size rule

1. Chosse $\alpha, \beta \in (0,1)$, set $s = 1$
2. If $f_0(x_k + sv_k) \le f_0(x_k) + s\alpha \nabla f_0(x_k)^\top v_k$, then return $s_k = s$.
3. Set $s = \beta s$ and return to step 2.

Newton-type methods

- Newton's method: $v_k = -\nabla^2 f_0(x_k)^{-1} \nabla f_0(x_k)$
- Interpretation: Minimizing second order approximation: $\min_x f_0(x_k) + \nabla f_0(x_k)^\top x + \frac{1}{2} x^\top \nabla^2 f_0(x_k) x$
- Quasi-Newton and variable-metric: $v_k = -H_k \nabla f_0(x_k)$
- Combined with any step size rule

Constrained Optimization Algorithms

Methods

- **Projected Descent:** Use a direction v_k from unconstrained algorithms and compute $x'_{k+1} = x_k + s_k v_k$ and then project x'_{k+1} onto the feasible set.
- Construct an unconstrained problem that approximates the original problem.

Barrier method

- Consider the problem $\min_x f_0(x)$ subject to $f_i(x) \le 0$ for $i = 1...m$.
- Define the approximate problem for some parameter $t \ge 0$: $\min_x f_0(x) + \frac{1}{t}\phi(x)$, where the barrier function $\phi(x)$ tends to ∞ as x tends to the boundary of the feasible set from inside.
- We can use the logarithmic barrier function: $\phi(x) = -\sum_{i=1}^m log(-f_i(x))$
- If f_0, f_i are convex, the approximated problem is convex
- The KKT condition is $t\nabla f_0(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) = 0$
- The approximation error: $0 \le f_0(x^*(t)) - p^* \le \frac{m}{t}$

Machine Learning

Linear Model

- Linear model: $y(x) = w^\top \phi(x) + e(x)$, where $\phi(x)$ is a feature map and $e(x)$ is a random error
- Least square: $\hat{w} \in \arg\min_w \sum_i (w^\top \phi(x_i) - y_i)^2$
- Quantile for $\alpha \in [0,1)$: $\hat{w} \in \arg\min_w \frac{1}{m(1-\alpha)} \sum_i max\{0, w^\top \phi(x_i) - y_i\} - \frac{1}{m} \sum_i w^\top \phi(x_i) - y_i$. Still a linear program.

Support Vector Machine

- Classification rule: $\hat{y}(x) = sign(\phi(x)^\top x)$
- Convex error function(hinge): $\max\{0, 1-\alpha\}$
- Objective: $\min_w \frac{1}{m} \sum_i \max\{0, 1 - y_i \phi(x_i)^\top w\}$
- We can add L_1 or L_2 norm regularization on w

Logistic Regression

- Variant of SVM with another loss function: $\min_w \frac{1}{m} \sum_i log[1 + exp(-y_i \phi(x_i)^\top w)]$
- Interpretation: $P\{Y = y | X = x\} = \frac{1}{1 + exp(-y\phi(x)^\top w)}$

Fisher discrimination

- Idea: find a direction in $\mathbb{R}^n$, such that the projection of data points onto the direction results in those points with positive label are from those with negative label
- Define $A_+ \in R^{n, m_+}$ is the matrix whose column cosists of x_i such that $y_i = 1$, and A_- similarly for negive labeled points. The centered(mean subtracted) data matrices are $\widetilde{A}_\pm = A_\pm (I_{m_\pm} - \frac{1}{m_\pm} \mathbf{1}\mathbf{1}^\top)$
- The mean-squared variation of the data projected on u around its centroid is: $u^\top M u = u^\top (\frac{1}{m_-} \widetilde{A}_- \widetilde{A}_-^\top + \frac{1}{m_+} \widetilde{A}_+ \widetilde{A}_+^\top) u$
- The goal is: $\max_{u \ne 0} \frac{(u^\top c)^2}{u^\top M u} = \min u^\top M u$ subject to $u^\top c = 1$
- The problem is convex, hence KKT condition implies: $u = \frac{1}{c^\top M^{-1} c} M^{-1} c$

SPCA and NNMF

- The original PCA problem: $\max_z z^\top \widetilde{X} \widetilde{X}^\top z$ s.t. $\|z\|_2 = 1$
- In sparse PCA, we add constraint $card(z) \le k$
- The problem leads to the rank 1 approximation problem: $\min_{p,q} \|X - pq^\top\|_F$ s.t. $card(p) \le k, card(q) \le h$
- One example of rank 1 approximation is: $\min_{p \ge 0, q \ge 0} \|X - pq^\top\|_F$ s.t. $card(p) \le k, card(q) \le h$. A standard solution is coordinate descent.

Finance

Mean-Variance (Markowitz) Models

- $x \in \mathbb{R}^n$ is the wealth allocation in n groups, r is the random vector of unit wealth return for each group. $\hat{r} = E[r], \Sigma = Cov(r)$
- Hence, $\hat{r}^\top x$ is the expected return and $x^\top \Sigma x$ is the variance of the portfolio.
- The problems are usually maximize return with variance constraints and minimize variance, with return constraints.
- Buget constraint: $x^\top \mathbf{1} \le b$, no short-selling: $x \ge 0$
- Sector bounds: $\sum_{i \in S} x_i \le \alpha x^\top \mathbf{1}$ for some $\alpha \in (0,1)$ and sector S.
- Diversification: sum of k largest x_i must not exceed $\eta x^\top \mathbf{1}$ for $\eta \in (0,1)$. This is equivalent to $kt + s^\top \mathbf{1} \le \eta x^\top \mathbf{1}$, $s \ge 0, x - t\mathbf{1} \le s$ where $t \in \mathbb{R}$ and $s \in \mathbb{R}^n$ are additional variables.
- The Sharpe Ratio of a portfolio is $(\hat{r}^\top x - r_f)/\sqrt{x^\top \Sigma x}$, where r_f is the return of a risk-free asset.
- If $\hat{r}^\top x > r_f$ and $\Sigma \succ 0$, then the portfolio that maximizes SR corresponds to a point on the efficient frontier. Max SR can be formulated as SOCP.

Value at Risk and Conditional Value at Risk

- The value-of-risk of a portfolio $r^\top x$ at probability level $\alpha \in (0,1)$ is defined as $VaR_\alpha(r^\top x) = -F^{-1}(\alpha)$, where F is the CDF of $r^\top x$.
- With probability no smaller that $1 - \alpha$, the investor is guaranteed to receive a return higher than $-VaR_\alpha$
- If r is Gaussian with mean $\hat{r}$ and covariance Σ, we have $VaR_\alpha(r^\top x) \le \gamma$ is equvalent as SOC constraint $\Phi^{-1}(1-\alpha)\|\Sigma^{\frac{1}{2}}\|_2 \le \gamma + \hat{r}^\top x$
- The conditional value at risk is the average of outcomes that are more extreme than VaR_α

Capital Asset Pricing Model

- The CAPM says that the expected return should be $E[r_i] = r_f + \beta_i(E[r_m] - r_f)$, where r_f is the return of a risk-free asset, r_m is the random return of the market and the systematic risk $\beta_i = Cov(r_i, r_m)/Var(r_m)$

Epi-Spline Curve Fitting

Epi-splines

- Epi-spline: Left continuous, piece-wise polynomial
- On the k^{th} segment, $q^k(x) = \sum_{i=0}^p a_i^k x^i$. Hence if we split the domain into N segments with degree p polynomial, our total number of parameters would be $N(p+1)$
- Continuity constraint: Let $m_1 ... m_{N-1}$ be the mesh points, we have $q^k(m_k) = q^{k+1}(m_k)$ for $k = 1...N-1$.
- Continuous differentiability: Continuity constraint plus $(q^k(m_k))' = (q^{k+1}(m_k))'$ for $k = 1...N-1$.
- Fixed values: $q^k(x) = s(k)$
- Monotonicity (non-increasing): $(q^k(x))' \le 0$ for $x \in (m_{k-1}, m_k)$, and $q^k(m_k) \ge q^{k+1}(m_{k+1})$. The second requirement is satisfied with continuity.
- Convexity: Continuity constraint and $(q^k(x))'' \ge 0$, $(q^k(m_k))' \le (q^{k+1}(m_{k+1}))'$ for $k = 1...N-1$.
- All constraints above are linear.

Optimization in Control

Continuous LTI Systems

- A continuous LTI system is described by a system ODEs: $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t)$
- The solution is: $x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^t e^{A(t-\tau)} Bu(\tau) d\tau$, $t \ge t_0$

Discrete-time LTI System and Discretization

- We discretize the time interval $t = k\Delta$ for $k = ... -1, 0, 1, ...$ and DTLTI system can be described as $x(k+1) = Ax(k) + Bu(k), y(k) = Cx(k)$
- The solution is: $x(k) = A^{k-k_0} x(k_0) + \sum_{i=k_0}^{k-1} A^{k-i-1} Bu(i)$ for $k >= k_0$.
- Assuming constant control input during one time interval, i.e. $u(t) = u(k\Delta), \forall t \in [k\Delta, (k+1)\Delta)$, we can discretize a CTLTI system by letting $A_\Delta = e^{A\Delta}, B_\Delta = \int_0^\Delta e^{A\tau} d\tau$. The we get the system $x(k+1) = A_\Delta x(k) + B_\Delta u(k)$

Optimization-based Control Synthesis

- Consider a generic DTLTI system, with a scalar input signal $u(k)$ and scalar output signal $y(k)$ (SISO)
- Given $x(0) = x_0$, determinte control sequence $u(k), k = 0, ...T-1$ such that $x(T) = x_T$
- $x(T) = A^\top x_0 + \sum_{i=0}^{T-1} A^{T-i-1} Bu(i) = A^\top x_0 + [A^{T-1}B, A^{T-2}B ... AB, B][u(0), u(1) ... u(T-1)]^\top = A^\top x_0 + R_T \mu_T$ where $R_T = [A^{T-1}B ... B]$
- Define $\xi_T = x_T - A^\top x_0$, the optimization problem becomes finding a vector $\mu_T \in \mathbb{R}^T$ such that $R_T \mu_T = \xi_T$.
- We assume that $T > n$ and that R_T is full row rank, then the problem admits an infinite number of possible solutions. We want to choose the one with minimum effort.
- Mimimizing energy: $\min_{\mu_T} \|\mu_T\|_2^2$ s.t. $R_T \mu_T = \xi_T$, the solution is $R_T^\top (R_T R_T^\top)^{-1} \xi_T$
- Minimum fuel control: $\min_{\mu_T} \|\mu_T\|_1$ s.t. $R_T \mu_T = \xi_T$. The problem can be formulated as LP.

Control Synthesis for Trajectory Tracking

- Finding a control sequence such that the output $y(k)$ tracks as closely as possible an assigned reference trajectory $y_{ref}(k)$
- We assume SISO, $x(0) = 0$ and $y_{ref}(0) = 0$, then we get $x(k) = A^{k-1}Bu(0) + ... + ABu(k-2) + Bu(k-1)$
- The output sequence $y(k) = Cx(k) = CA^{k-1}Bu(0) + ... + CABu(k-2) + CBu(k-1)$ for $k = 1, ..., T$
- Rewrite in matrix form, we have
$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{T-1}B & \dots & CAB & CB \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(T-1) \end{bmatrix}$$
that is $y_T = \Phi_T \mu_T$
- Define $\mathcal{Y}_{ref} = [y_{ref}(1), \dots, y_{ref}(T)]^\top$. The problem now becomes $\min_{\mu_T} \|\Phi_T \mu_T - \mathcal{Y}_{ref}\|_2^2$
- We can also add a L_2 norm or instantaneous rate of variation of μ_T to penalize energy output.

Continuous-time Lyapunov Stability Analysis

- The CTLTI system $\dot{x}(t) = Ax(t) + Bu(t)$ is said to be (asymptotically) stable if, for $u(t) = 0$, it holds that $\lim_{t\to\infty} x(t) = 0, \forall x(0) = x_0$
- A necessary condition is the existance of a quadratic Lyapunov function $V(x) = x^\top Px \succ 0$ such that $\dot{V}(x) = x^\top (A^\top P + PA)x \prec 0$. The condition is equivalent to $\exists P \succ 0$ such that $A^\top P + PA \prec 0$.
- We can reformulate the strict LMIs into non-strict LMIs: $\exists W \succeq I$ such that $WA^\top + AW \preceq -I$.
- Then we can use optimization technique to find P: $\min_{W,v} v$ subject to $WA^\top + AW \preceq -I$, $I \preceq W \preceq vI$

Stabilizing State-feedback Design

- Assume that the control input takes the following state-feedback form: $u(t) = Kx(t)$ where K is the state-feedback gain matrix.
- We have the controlled system: $\dot{x}(t) = (A + BK)x(t)$
- The stability condition is: $\exists W \succ 0, WA^\top + AW + (KW)^\top B^\top + B(KW) \prec -I$
- Define $Y = KW$, the condition becomes $\exists W \succ 0, WA^\top + AW + Y^\top B^\top + BY \prec -I$
- Then we can solve: $\min_{W,Y,v} v + \eta\|Y\|_2$ subject to $WA^\top + AW + Y^\top B^\top + BY \prec -I, I \preceq W \preceq vI$ and then recover K as $K = YW^{-1}$. $\eta \geq 0$ is the trade off parameter.

Robust Feedback Design

- Consider the system $\dot{x}(t) = Ax(t) + Bu(t)$, where now the matrices A, B can vary inside a polytope of given matrices $(A_i, B_i), i = 1, \dots, N$. That is $(A(t), B(t)) \in co\{(A_1, B_1), \dots, (A_N, B_N)\}$.
- A sufficient condition for stability of the uncertain system is the existence of a common quadratic Lyapunov function for all the vertex systems, that is $\exists W \succ I : WA_i^\top + A_iW \prec -I, i = 1, \dots, N$
- Based on this condition, we can build a controller.
- The controlled system is described by $\dot{x}(t) = (A + BK)x(t)$. It is stable if $\exists W \succ I : W(A_i + B_iK)^\top + (A_i + B_iK)W \prec -I, i = 1, \dots, N$
- Introducing the variable $Y = KW$, the feedback gain matrix K of a stabilizing controller can be found by solving the following problem. $\min_{W,Y,v} v + \eta\|Y\|_2$ subject to $WA_i^\top + A_iW + Y^\top B_i^\top + B_iY \prec -I, i = 1, \dots, N$, $I \preceq W \preceq vI$, and then recover K as $K = YW^{-1}$