Doğrusal Olmayan En Küçük Kareler

T.C. Trakya Üniversitesi
Mühendislik Fakültesi
Elektrik - Elektronik Mühendisliği Bölümü
Kontrol Anabilim Dalı

Dr. Öğr. Üyesi Işık İlber Sırmatel
sirmatel.github.io

Kaynak (source)

Lecture Slides for Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares.
Stephen Boyd, Lieven Vandenberghe

Konu listesi

Bölüm 1

Doğrusal olmayan denklemler ve en küçük kareler

Doğrusal olmayan denklemler

- n bilinmeyenli $(x_1, \ldots, x_n)$ m denklemli **doğrusal olmayan** (*nonlinear*) denklem takımı:

$$f_i(x_1, \ldots, x_n) = 0, \quad i = 1, \ldots, m$$

- $f_i(x) = 0$: i. denklem; $f_i(x)$: i. kalıntı
- $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}$: bilinmeyenler vektörü ($x \in \mathbb{R}^n$)
- denklem takımını vektör denklemi $f(x) = 0$ olarak yazalım ($f : \mathbb{R}^n \to \mathbb{R}^m$)

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}$$

- f bir afin fonksiyon (yani, $f(x) = Ax - b$) ise, vektör denklemi m denklemli doğrusal denklem takımına (yani, $Ax = b$'ye) dönüşür

Doğrusal olmayan denklemler

$f(x) = 0$ denklemine $\left(f : \mathbb{R}^n \to \mathbb{R}^m\right)$

- $m > n$ ise aşırı belirli
- $m = n$ ise kare
- $m < n$ ise eksik belirli

denir

Doğrusal olmayan en küçük kareler

- doğrusal olmayan en küçük kareler (*nonlinear least squares* (NLS)) problemi:

$$\|f(x)\|^2 = f_1(x)^2 + \cdots + f_m(x)^2$$

  ifadesini minimize eden $\hat{x}$'i bulmak
- bu problem,

$$\underset{x}{\text{minimize}} \quad \|f(x)\|^2$$

  formunda bir optimizasyon problemidir
- $f(x) = 0$ denklemini çözme problemi, bu problemin bir özel halidir
- (doğrusal) en küçük karelere benzer şekilde, çok kullanışlı bir yöntemdir

Optimalite koşulu

- problem:
$$\underset{x}{\text{minimize}} \quad \|f(x)\|^2$$
- optimalite koşulu: $\nabla\|f(\hat{x})\|^2 = 0$
- her optimal nokta bu koşulu sağlar, ancak bu koşulu sağlayan her nokta optimal değildir (koşul, gerek koşuldur, yeter koşul değildir)
- koşul şu şekilde ifade edilebilir: $2Df(\hat{x})^T f(\hat{x}) = 0$
- buradaki $Df(\hat{x})$'ye **Jakobi** (*Jacobian*) matrisi denir ($Df(\hat{x}) \in \mathbb{R}^{m\times n}$):
$$Df(\hat{x})_{ij} = \frac{\partial f_i}{\partial x_j}(\hat{x}), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

Doğrusal olmayan en küçük kareler

doğrusal olmayan en küçük kareler problemini çözmenin zorluğu:

- doğrusal olmayan denklemleri veya doğrusal olmayan en küçük kareler problemini çözmek (genel olarak) doğrusal denklemleri çözmekten **çok daha zordur**
- bir çözümün mevcut olup olmadığını tespit etmek bile zordur
- dolayısıyla bunları çözmek için **buluşsal** (*heuristic*) algoritmalar kullanırız (k-ortalamalarda olduğu gibi):
  - bu tip algoritmaların daima çalışacağının garantisi yoktur
  - ancak pratikte genellikle düzgün çalışırlar

Örnek: Denge noktalarının hesaplanması

denge (*equilibrium*) noktalarının hesaplanması

- denge fiyatları: $S(p) = D(p)$'yi sağlayan fiyat vektörü p'yi bul ($p \in \mathbb{R}^n$)
  - $S(p)$: fiyatların fonksiyonu olarak n adet malın **arzı** (*supply*)
  - $D(p)$: fiyatların fonksiyonu olarak n adet mala olan **talep** (*demand*)
  - denklem: $f(p) = S(p) - D(p)$
- kimyasal/biyolojik denge: $C(c) = G(c)$'yi sağlayan **derişim** (*concentration*) vektörü c'yi bul ($c \in \mathbb{R}^n$)
  - $C(c)$: derişimlerin fonksiyonu olarak n adet maddenin/türün **tüketim**i (*consumption*)
  - $G(c)$: derişimlerin fonksiyonu olarak n adet maddenin/türün **üretim**i (*generation*)
  - denklem: $f(c) = C(c) - G(c)$

Örnek: Erim ölçümleriyle konum kestirme

- $x \in \mathbb{R}^3$: 3 boyutlu konum; x'i kestirmek istiyoruz
- **erim ölçümleri** (*range measurements*) bilinen konumlara olan gürültülü uzaklıkları verir:

$$\rho_i = \|x - a_i\| + v_i, \quad i = 1, \ldots, m$$

- a_i: bilinen konumlar; v_i: gürültüler
- en küçük kareler konum kestirme:

$$\sum_{i=1}^{m} \left(\|x - a_i\| - \rho_i\right)^2$$

  ifadesini minimize eden $\hat{x}$'i bulmak
- **küresel konum belirleme sistemi** (*global positioning system* (GPS)) bu şekilde çalışır

Bölüm 2

Levenberg-Marquardt algoritması

Levenberg-Marquardt algoritması

Levenberg-Marquardt algoritması için temel fikir:

- $f(x)$ için herhangi bir z noktasında

  $$\hat{f}(x; z) = f(z) + Df(z)(x - z)$$

  şeklinde bir afin yaklaşıklık oluşturabiliriz
- x, z'ye yakın ise $\hat{f}(x; z) \approx f(x)$ olur
- $\|\hat{f}(x; z)\|^2$'yi doğrusal en küçük kareler ile minimize edebiliriz
- z'yi şu anki **döngü adımı** (*iterate*) olarak kullanarak **döngülü** (*iterative*) şekilde çözümü hesaplamaya çalışırız
- Levenberg-Marquardt algoritmasına "**sönümlü** (*damped*) en küçük kareler yöntemi" de denir

Levenberg-Marquardt algoritması

- döngü adımlarını $x^{(1)}, x^{(2)}, \ldots$ ile gösterelim
- döngü k'de, f'in $x^{(k)}$'deki afin yaklaşıklığını oluşturalım:

$$\hat{f}(x; x^{(k)}) = f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})$$

- $$\|\hat{f}(x; x^{(k)})\|^2 + \lambda^{(k)}\|x - x^{(k)}\|^2$$

  ifadesini minimize eden noktayı, $x^{(k+1)}$ döngü adımı olarak seçelim (burada $\lambda^{(k)} > 0$ **sönüm çarpanı** (*damping factor*))

- $\|\hat{f}(x; x^{(k)})\|^2$ küçük olsun isteriz ancak $x^{(k)}$'ten çok fazla uzaklaşmak istemeyiz (çünkü $x^{(k)}$'ten çok fazla uzaklaşınca $\hat{f}(x; x^{(k)}) \approx f(x)$ geçerliliğini yitirir)

Levenberg-Marquardt döngüsü

- $x^{(k+1)}$,

$$\underset{x}{\text{minimize}} \quad \|f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})\|^2 + \lambda^{(k)}\|x - x^{(k)}\|^2$$

  şeklindeki en küçük kareler probleminin çözümüdür
- çözüm aşağıdaki şekilde hesaplanabilir

$$x^{(k+1)} = x^{(k)} - \left(Df(x^{(k)})^T Df(x^{(k)}) + \lambda^{(k)} I\right)^{-1} Df(x^{(k)})^T f(x^{(k)})$$

- çözüm ifadesindeki matris tersi daima mevcuttur (çünkü $\lambda^{(k)} > 0$)
- ancak $Df(x^{(k)})^T f(x^{(k)}) = 0$ olduğunda (yani, optimalite koşulu sağlandığında) $x^{(k+1)} = x^{(k)}$ olur

Levenberg-Marquardt algoritması

$\lambda^{(k)}$'yı ayarlamak - fikir:

- $\lambda^{(k)}$ çok büyük ise $x^{(k+1)}$ $x^{(k)}$'e çok yakın olur ve algoritmanın ilerlemesi yavaş olur
- $\lambda^{(k)}$ çok küçük ise $x^{(k+1)}$ $x^{(k)}$'e çok uzak olur ve afin yaklaşıklık $(\hat{f}(x; x^{(k)}) \approx f(x))$ kötü olur (başka bir deyişle: $\hat{f}(x; x^{(k)})$, $f(x)$'i yeteri kadar iyi temsil edemez)

$\lambda^{(k)}$'yı ayarlamak - güncelleme mekanizması:

- $\|f(x^{(k+1)})\|^2 < \|f(x^{(k)})\|^2$ ise, hesaplanan yeni $x^{(k+1)}$ değerini kabul et ve λ'yı azalt:

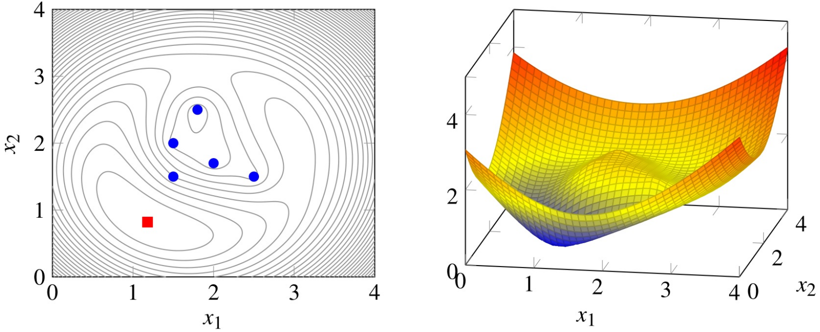
$$\lambda^{(k+1)} = 0.8\lambda^{(k)}$$

- aksi halde, λ'yı arttır ve $x^{(k)}$'i güncelleme:

$$\lambda^{(k+1)} = 2\lambda^{(k)} \qquad x^{(k+1)} = x^{(k)}$$

Levenberg-Marquardt algoritması

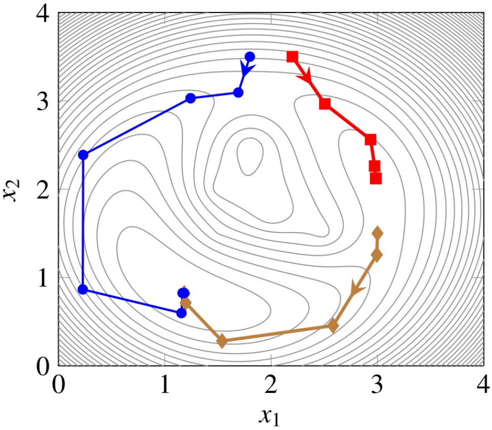
örnek: erim ölçümleriyle konum kestirme



- soldaki grafik: $\|f(x)\|^2$'in **seviye eğrileri** (*level curves* veya *contour lines*)
- sağdaki grafik: $\|f(x)\|$
- mavi daireler: 5 konum (bu konumlara olan ölçümler veri)
- kırmızı kare: konum kestirimi $\hat{x}$

Levenberg-Marquardt algoritması

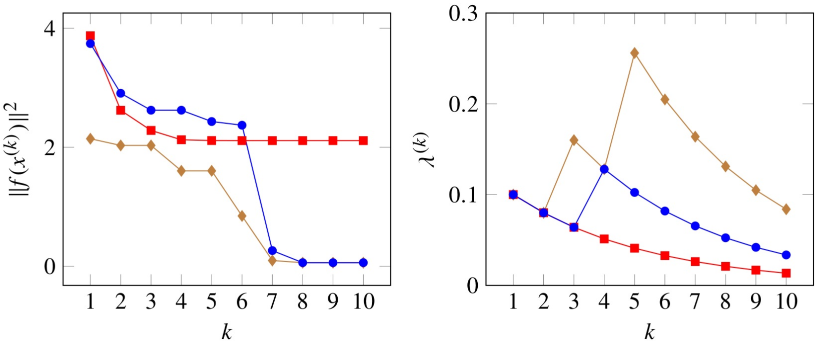
örnek: erim ölçümleriyle konum kestirme



algoritmanın 3 farklı başlangıç noktasından çalıştırılmasıyla elde edilen döngü adımı yörüngeleri

Levenberg-Marquardt algoritması

örnek: erim ölçümleriyle konum kestirme



algoritmanın 3 farklı başlangıç noktasından çalıştırılmasıyla elde edilen "döngü adımı için amaç fonksiyonu değeri" (yani, $\|f(x^{(k)})\|^2$) ve sönüm çarpanı $\lambda^{(k)}$ grafikleri

Bölüm 3

Doğrusal olmayan model uydurma

Doğrusal olmayan model uydurma

doğrusal olmayan model uydurma problemi:

$$\underset{\theta}{\text{minimize}} \quad \sum_{i=1}^{N} \left(\hat{f}(x^{(i)}; \theta) - y^{(i)} \right)^2$$

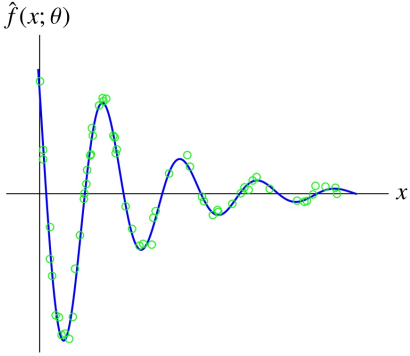
- $x^{(i)}, \ldots, x^{(N)}$: öznitelik vektörleri
- $y^{(i)}, \ldots, y^{(N)}$: karşılık gelen sonuçlar
- model $\hat{f}(x; \theta)$, parametreler $\theta_1, \ldots, \theta_p$ ile parametrelendirilmiştir
- bu model formu,

  $$\hat{f}(x; \theta) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

  formundaki "parametrelere göre doğrusal" modelin genelleştirilmiş halidir
- burada $\hat{f}(x; \theta)$'in, θ'nın doğrusal olmayan bir fonksiyonu olmasına izin veriyoruz
- minimizasyon, model parametreleri θ üzerinden olur

Doğrusal olmayan model uydurma - Örnek

model: $\hat{x}(x;\theta) = \theta_1 \exp(\theta_2 x)\cos(\theta_3 x + \theta_4)$



dört parametreli $(\theta_1, \theta_2, \theta_3, \theta_4)$ bir doğrusal olmayan en küçük kareler problemi:

$$\underset{\theta}{\text{minimize}} \quad \sum_{i=1}^{N} \left(\theta_1 \exp(\theta_2 x^{(i)})\cos(\theta_3 x^{(i)} + \theta_4) - y^{(i)}\right)^2$$

Bölüm 4

Doğrusal olmayan en küçük kareler sınıflandırma

NLS sınıflandırma

doğrusal en küçük kareler sınıflandırıcı:

- sınıflandırıcı: $\hat{f}(x) = \mathrm{sign}(\tilde{f}(x))$
- $\tilde{f}(x) = \theta_1 f_1(f) + \cdots + \theta_p f_p(x)$
- $\sum_{i=1}^{N} (\tilde{f}(x_i) - y_i)^2$ (buna opsiyonel olarak düzenlileştirme terimi eklenebilir) minimize edilerek θ seçilir

doğrusal olmayan en küçük kareler (NLS) sınıflandırıcı:

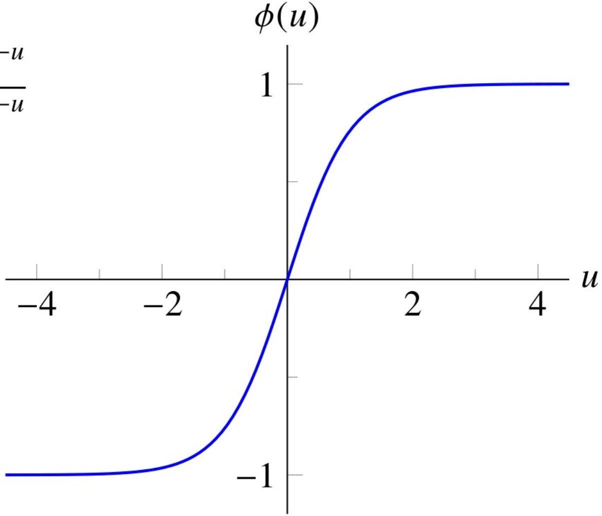
- asıl problem

$$\sum_{i=1}^{N} (\mathrm{sign}(\tilde{f}(x_i)) - y_i)^2$$

  minimize edilerek θ'yı seçme şeklinde formüle edilir
- buradaki sign fonksiyonu, pürüzsüz yaklaşıklığı ϕ (örneğin, sigmoit fonksiyonu) ile değiştirilir
- Levenberg-Marquardt algoritması kullanılarak $\sum_{i=1}^{N} (\phi(\tilde{f}(x_i)) - y_i)^2$ minimize edilir

Sigmoit fonksiyonu

$$\phi(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

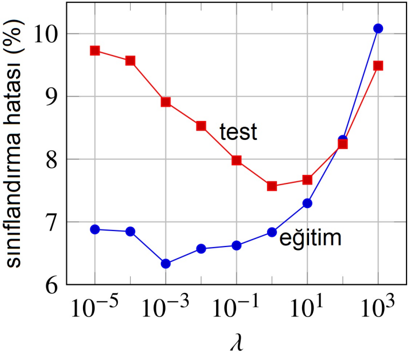


NLS sınıflandırma - Örnek: El yazısı rakam

- MNIST veri kümesi; öznitelik vektörü $x \in \mathbb{R}^{493}$ piksel yoğunlukları (her x bir el yazısı rakam görüntüsü)
- 10-taraflı çok-sınıflı NLS sınıflandırıcı: $\%7.5$ test hatası
- Boole sınıflandırıcılar

$$\underset{\beta,\nu}{\text{minimize}} \quad \sum_{i=1}^{N} (\phi((x^{(i)})^T \beta + \nu) - y^{(i)})^2 + \lambda \|\beta\|^2$$

şeklindeki NLS problemleri çözülerek hesaplanır



NLS sınıflandırma - Örnek: El yazısı rakam

öznitelik mühendisliği:

- formülasyona 5000 adet rastgele öznitelik ekleyelim
- bu durumda test veri kümesi için $\%2$'lik hata oluşur
- bu seviyedeki hata insan başarımına denktir
- daha fazla/daha iyi öznitelik mühendisliği ile insan başarımından çok daha yüksek başarım elde edilebilir

Bölüm 5

Kısıtlı doğrusal olmayan en küçük kareler

Alt Bölüm 1

Problem formülasyonu

Kısıtlı doğrusal olmayan en küçük kareler

- doğrusal olmayan en küçük kareler problemine eşitlik kısıtları ekleyelim:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_1(x)^2 + \cdots + f_m(x)^2 \\ \text{bağlı} & g_1(x) = 0, \ \ldots, \ g_p(x) = 0 \end{array}$$

- $f_i(x)$: i. (skaler) kalıntı; $g_i(x) = 0$: i. (skaler) eşitlik kısıtı
- vektör notasyonu ile, yani

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix} \qquad g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_p(x) \end{bmatrix} = 0$$

ile, problem aşağıdaki şekilde yazılabilir:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|f(x)\|^2 \\ \text{bağlı} & g(x) = 0 \end{array}$$

Kısıtlı doğrusal olmayan en küçük kareler

- $g(x) = 0$ koşulunu (yani, kısıtları) sağlayan x'e **olanaklı** (*feasible*) denir
- bir $\hat{x}$ noktası olanaklı ise ve bütün olanaklı x'ler için $\|f(x)\|^2 \geq \|f(\hat{x})\|^2$ koşulu sağlanıyorsa, $\hat{x}$ noktası bir **çözüm**dür (*solution*)
- kısıtlı doğrusal olmayan en küçük kareler (*constrained nonlinear least squares* (CNLS)) problemini çözmek genel olarak zordur, ancak yararlı **buluşsal yöntemler** (*heuristics*) mevcuttur

Lagrange çarpanları

- CNLS probleminin **Lagrange fonksiyonu** (*Lagrangian*)

$$\begin{aligned} L(x,z) &= \|f(x)\|^2 + z_1 g_1(x) + \ldots + z_m g_m(x) \\ &= \|f(x)\|^2 + g(x)^T z \end{aligned}$$

  şeklinde tanımlı fonksiyondur
- $z = \begin{bmatrix} z_1 & \ldots & z_p \end{bmatrix}^T$: Lagrange **çarpan**ları (*multiplier*) vektörü ($z \in \mathbb{R}^p$)
- Lagrange çarpanları yöntemi: $\hat{x}$ bir çözüm ise

$$\frac{\partial L}{\partial x_i}(\hat{x},\hat{z}) = 0, \quad i = 1,\ldots,n \qquad \frac{\partial L}{\partial z_i}(\hat{x},\hat{z}) = 0, \quad i = 1,\ldots,p$$

  koşullarını sağlayan bir $\hat{z}$ mevcuttur (gradyanlardan ($\nabla g_1(\hat{x}),\ldots,\nabla g_p(\hat{x})$) oluşan vektör kümesinin doğrusal bağımsız olması koşuluyla)
- $\hat{z}$ bir optimal Lagrange çarpanıdır

Optimalite koşulu

- Lagrange fonksiyonunun x'e göre gradyanı:

$$\nabla_x L(\hat{x}, \hat{z}) = 2Df(\hat{x})^T f(\hat{x}) + Dg(\hat{x})^T \hat{z}$$

- Lagrange fonksiyonunun z'ye göre gradyanı:

$$\nabla_z L(\hat{x}, \hat{z}) = g(\hat{x})$$

- optimalite koşulu: $\hat{x}$ optimal ise

$$2Df(\hat{x})^T f(\hat{x}) + Dg(\hat{x})^T \hat{z} = 0, \qquad g(\hat{x}) = 0$$

  koşullarını sağlayan bir $\hat{z}$ mevcuttur ($Dg(\hat{x})$'in satırlarından oluşan vektör kümesinin doğrusal bağımsız olması koşuluyla)
- bu koşul optimalite için **gerekli**dir (*necessary*) ancak **yeterli** (*sufficient*) değildir

Kısıtlı (doğrusal) en küçük kareler

- kısıtlı en küçük kareler problemini hatırlayalım:

$$\begin{array}{rl} \underset{x}{\text{minimize}} & \|Ax-b\|^2 \\ \text{bağlı} & Cx = d \end{array}$$

- bu, doğrusal olmayan problemin $f(x) = Ax - b$, $g(x) = Cx - d$ ile oluşan bir özel halidir
- probleme genel optimalite koşulunu uygulayalım:

$$2Df(\hat{x})^T f(\hat{x}) + Dg(\hat{x})^T \hat{z} = 2A^T(A\hat{x} - b) + C^T \hat{z} = 0$$
$$g(\hat{x}) = C\hat{x} - d = 0$$

- bunlar KKT denklemleridir

$$\begin{bmatrix} 2A^TA & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} 2A^Tb \\ d \end{bmatrix}$$

Alt Bölüm 2

Ceza yöntemi

Ceza yöntemi

- (kısıtsız) doğrusal olmayan en küçük kareler problemleri dizisini çözelim

$$\underset{x}{\text{minimize}} \quad \|f(x)\|^2 + \mu\|g(x)\|^2 = \left\| \begin{bmatrix} f(x) \\ \sqrt{\mu}g(x) \end{bmatrix} \right\|^2$$

- μ: **ceza** (*penalty*) parametresi ($\mu > 0$)
- $g(x) = 0$ koşulunun sağlanmasında ısrar etmek yerine, burada sıfırdan olan sapmalar için bir ceza terimi oluşturuyoruz
- artan $\mu^{(1)}, \mu^{(2)}, \ldots$ dizisi için

$$\|f(x)\|^2 + \mu^{(k)}\|g(x)\|^2$$

  minimize edierek $x^{(k+1)}$ hesaplanır
- burada $x^{(k+1)}$, $x^{(k)}$ noktasından başlatılan Levenberg-Marquardt algoritmasıyla hesaplanır

Ceza yönteminin sonlanması

- CNLS için optimalite koşulunu hatırlayalım:

$$2Df(\hat{x})^T f(\hat{x}) + Dg(\hat{x})^T \hat{z} = 0, \qquad g(\hat{x}) = 0$$

- $x^{(k)}$, doğrusal LS problemi için normal denklemleri sağlar:

$$2Df(x^{(k)})^T f(x^{(k)}) + 2\mu^{(k-1)} Dg(x^{(k)})^T g(x^{(k)}) = 0$$

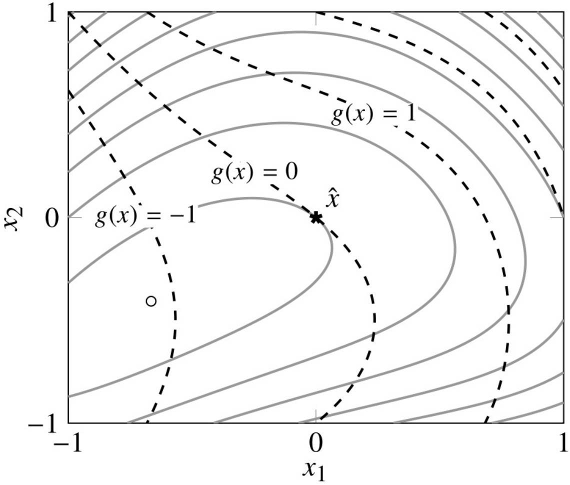
- bunu, $z^{(k)} = 2\mu^{(k-1)} g(x^{(k)})$ tanımıyla

$$2Df(x^{(k)})^T f(x^{(k)}) + Dg(x^{(k)})^T z^{(k)} = 0$$

  şeklinde yazabiliriz
- buradan $x^{(k)}, z^{(k)}$'nin optimalite koşulundaki birinci denklemi sağladığını görüyoruz
- olanaklılık koşulu $g(x^{(k)}) = 0$ yeteri kadar büyük $\mu^{(k-1)}$ için ve ancak yaklaşık olarak sağlanır
- $\|g(x^{(k)})\|$ yeteri kadar küçük olduğunda ceza yöntemi sonlanır (**sonlanma**: *termination*)

Ceza yöntemi - Örnek

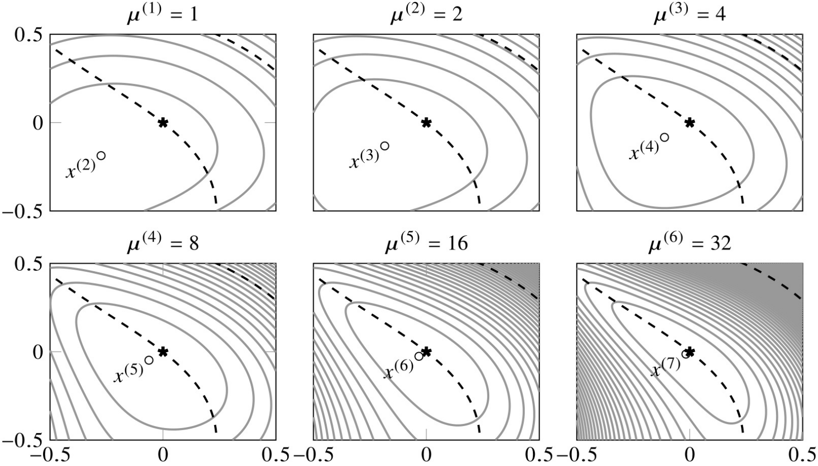
$$f(x_1, x_2) = \begin{bmatrix} x_1 + \exp(-x_2) \\ x_1^2 + 2x_1 + 1 \end{bmatrix} \qquad g(x_1, x_2) = x_1 + x_1^3 + x_2 + x_2^2$$



- düz çizgi: $\|f(x)\|^2$ için seviye eğrileri
- kesikli çizgi: $g(x)$ için seviye eğrileri
- $\hat{x}$: problemin çözümü

Ceza yöntemi - Örnek

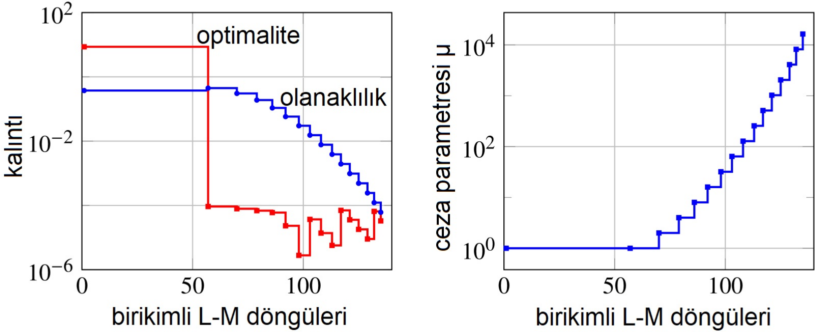
ilk 6 yineleme adımı



- düz çizgi: $\|f(x)\|^2 + \mu^{(k)}\|g(x)\|^2$ için seviye eğrileri
- kesikli çizgi: $g(x)$ için seviye eğrileri

Ceza yöntemi - Örnek

yakınsama (*convergence*)



- soldaki grafik optimalite koşulundaki kalıntıları gösteriyor
- mavi eğri: $\|g(x^{(k)})\|$ (olanaklılık için bu küçük olsun (idealde 0 olsun) isteriz)
- kırmızı eğri: $2Df(x^{(k)})^T f(x^{(k)}) + Dg(x^{(k)})^T z^{(k)}$ (optimalite için bu küçük olsun (idealde 0 olsun) isteriz)

Alt Bölüm 3

Artırılmış Lagrange fonksiyonu yöntemi

Ceza yönteminin dezavantajı

- $\mu^{(k)}$ hızlı bir şekilde artar ve $g(x)$'in 0'a yakın bir değer almasını sağlamak için $\mu^{(k)}$ büyük bir değer almak zorundadır
- büyük $\mu^{(k)}$ için doğrusal olmayan en küçük kareler **altproblem**ini (**subproblem**) çözmek zorlaşır
- büyük $\mu^{(k)}$ için, Levenberg-Marquardt yöntemi çok sayıda döngü yapmak zorunda kalabilir, veya başarısız olabilir (yani, altprobleme çözüm bulamadan sonlanabilir)

Artırılmış Lagrange fonksiyonu

- kısıtlı doğrusal olmayan en küçük kareler problemi için **artırılmış** (*augmented*) Lagrange fonksiyonu:

$$\begin{aligned} L_\mu(x,z) &= L(x,z) + \mu\|g(x)\|^2 \\ &= \|f(x)\|^2 + g(x)^T z + \mu\|g(x)\|^2 \end{aligned}$$

- $L_\mu(x,z)$, (standart) Lagrange fonksiyonu $L(x,z)$'nin bir karesel ceza terimi ($\|g(x)\|^2$) ile artırılmış halidir
- μ: **ceza** (*penalty*) parametresi ($\mu > 0$)
- artırılmış Lagrange fonksiyonu $L_\mu(x,z)$,

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|f(x)\|^2 + \mu\|g(x)\|^2 \\ \text{bağlı} & g(x) = 0 \end{array}$$

  şeklindeki (asıl probleme denk) problemin Lagrange fonksiyonudur

Artırılmış Lagrange fonksiyonu

artırılmış Lagrange fonksiyonunun minimize edilmesi:

- artırılmış Lagrange fonksiyonu $L_\mu(x,z)$ için denk ifadeler

$$\begin{aligned} L_\mu(x,z) &= \|f(x)\|^2 + g(x)^T z + \mu\|g(x)\|^2 \\ & \|f(x)\|^2 + \mu\|g(x) + \frac{1}{2\mu}z\|^2 - \frac{1}{2\mu}\|z\|^2 \\ &= \left\| \begin{bmatrix} f(x) \\ \sqrt{u}g(x) + \frac{z}{2\sqrt{\mu}} \end{bmatrix} \right\|^2 - \frac{1}{2\mu}\|z\|^2 \end{aligned}$$

- $L_\mu(x,z)$, Levenberg-Marquardt yöntemiyle x'e göre (sabit μ ve z için) minimize edilebilir:

$$\underset{x}{\text{minimize}} \quad \left\| \begin{bmatrix} f(x) \\ \sqrt{u}g(x) + \frac{z}{2\sqrt{\mu}} \end{bmatrix} \right\|^2$$

Lagrange çarpanlarının güncellenmesi

- $L_\mu(x,z)$'yi minimize eden $\tilde{x}$ noktası

$$2Df(\tilde{x})^T f(\tilde{x}) + Dg(\tilde{x})^T(2\mu g(\tilde{x}) + z) = 0$$

  denklemini sağlar
- $\tilde{z} = z + 2\mu g(\tilde{x})$ tanımıyla, bu denklemi aşağıdaki şekilde yazabiliriz:

$$2Df(\tilde{x})^T f(\tilde{x}) + Dg(\tilde{x})^T \tilde{z} = 0$$

- bu denklem,

$$2Df(\hat{x})^T f(\hat{x}) + Dg(\hat{x})^T \hat{z} = 0, \qquad g(\hat{x}) = 0$$

  şeklinde yazdığımız optimalite koşulundaki birinci denklemdir
- bu denklem, $g(\tilde{x}) = 0$ ise $\tilde{x}$'in optimal olduğunu gösterir
- ayrıca, $g(\tilde{x})$ küçük değilse, $\tilde{z}$'nin z için iyi bir güncelleme olduğuna işaret eder

Artırılmış Lagrange fonksiyonu algoritması

1. başlangıç noktası $x^{(k)}$ için Levenberg-Marquardt algoritmasını kullanarak

$$\|f(x)\|^2 + \mu^{(k)}\|g(x) + \frac{1}{2\mu^{(k)}}z^{(k)}\|^2$$

ifadesini minimize eden noktayı $x^{(k+1)}$ olarak seç

2. *Lagrange çarpanlarını güncelle*:

$$z^{(k+1)} = z^{(k)} + 2\mu^{(k)}g(x^{(k+1)})$$

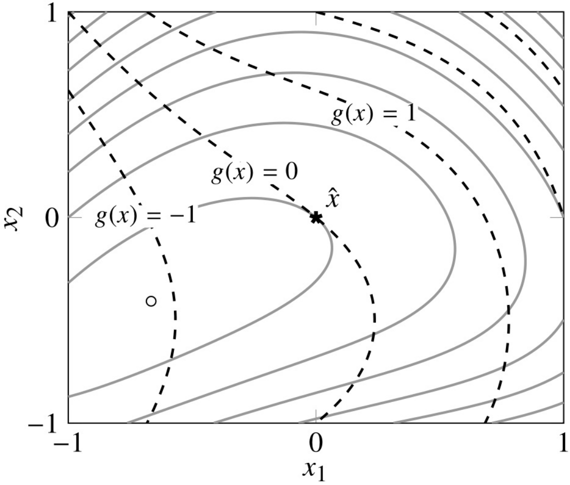
3. *ceza parametresini güncelle*:

$$\mu^{(k+1)} = \begin{cases} \mu^{(k)} & \|g(x^{(k+1)})\| < 0.25\|g(x^{(k)})\| \text{ ise} \\ 2\mu^{(k)} & \text{aksi halde} \end{cases}$$

- döngü $z^{(1)} = 0$, $\mu^{(1)} = 1$ ve verilen başlangıç noktası $x^{(1)}$ ile başlar
- μ, ceza yöntemine göre daha yavaş bir şekilde ve sadece gerektiğinde arttırılır

Artırılmış Lagrange fonks. yöntemi - Örnek

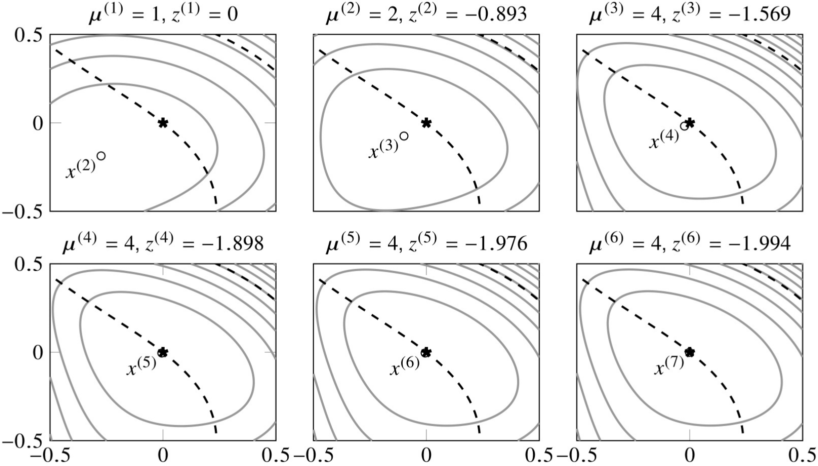
$$f(x_1, x_2) = \begin{bmatrix} x_1 + \exp(-x_2) \\ x_1^2 + 2x_1 + 1 \end{bmatrix} \qquad g(x_1, x_2) = x_1 + x_1^3 + x_2 + x_2^2$$



- düz çizgi: $\|f(x)\|^2$ için seviye eğrileri
- kesikli çizgi: $g(x)$ için seviye eğrileri
- $\hat{x}$: problemin çözümü

Artırılmış Lagrange fonks. yöntemi - Örnek

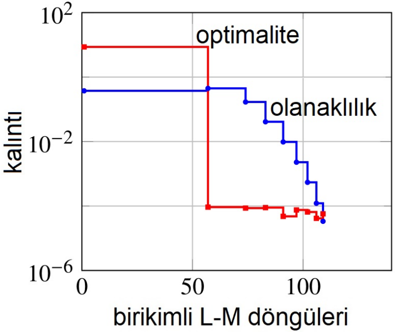
ilk 6 yineleme adımı

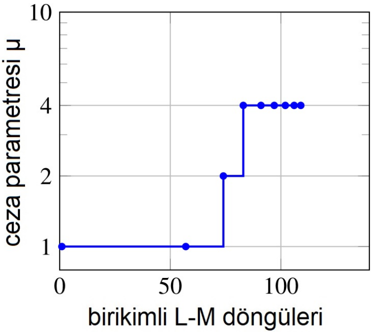


- düz çizgi: $L_{\mu^{(k)}}(x, z^{(k)})$ için seviye eğrileri
- kesikli çizgi: $g(x)$ için seviye eğrileri

Artırılmış Lagrange fonks. yöntemi - Örnek

yakınsama (*convergence*)



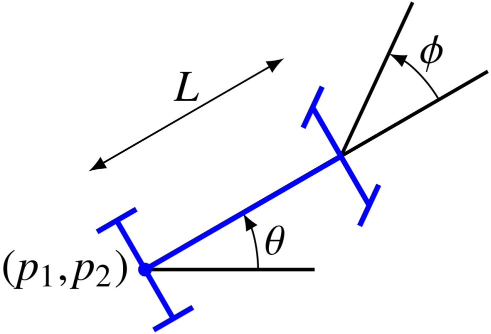


- soldaki grafik optimalite koşulundaki kalıntıları gösteriyor
- mavi eğri: $\|g(x^{(k)})\|$ (olanaklılık için bu küçük olsun (idealde 0 olsun) isteriz)
- kırmızı eğri: $2Df(x^{(k)})^T f(x^{(k)}) + Dg(x^{(k)})^T z^{(k)}$ (optimalite için bu küçük olsun (idealde 0 olsun) isteriz)

Alt Bölüm 4

Örnek: Doğrusal olmayan kontrol

Arabanın basit modeli



$$\frac{dp_1}{dt} = s(t)\cos(\theta(t))$$
$$\frac{dp_2}{dt} = s(t)\sin(\theta(t))$$
$$\frac{d\theta}{dt} = \frac{s(t)}{L}\tan(\phi(t))$$

- $s(t) \in \mathbb{R}$: **sürat** (*speed*)
- $\phi(t) \in \mathbb{R}$: **direksiyon açısı** (*steering angle*)
- $p(t) \in \mathbb{R}^2$: **konum** (*position*)
- $\theta(t) \in \mathbb{R}$: **yönelim** (*orientation*)

Zamanda ayrıklaştırılmış model

- zamanda **ayrıklaştırılmış** (*discretized*) model (küçük bir **zaman aralığı** (*time interval*) h değeri için):

$$p_1(t+h) \approx p_1(t) + hs(t)\cos(\theta(t))$$
$$p_2(t+h) \approx p_2(t) + hs(t)\sin(\theta(t))$$
$$\theta(t+h) \approx \theta(t) + h\frac{s(t)}{L}\tan(\phi(t))$$

- durum (x_k) ve giriş (u_k) vektörlerini tanımlayalım:

$$x_k = \begin{bmatrix} p_1(kh) \\ p_2(kh) \\ \theta(kh) \end{bmatrix} \qquad u_k = \begin{bmatrix} s(kh) \\ \phi(kh) \end{bmatrix}$$

- zamanda ayrıklaştırılmış modeli

$$f(x_k, u_k) = \begin{bmatrix} (x_k)_1 + h(u_k)_1\cos((x_k)_3) \\ (x_k)_2 + h(u_k)_1\sin((x_k)_3) \\ (x_k)_3 + h\frac{(u_k)_1}{L}\tan((u_k)_2) \end{bmatrix}$$

ile $x_{k+1} = f(x_k, u_k)$ formunda yazabiliriz

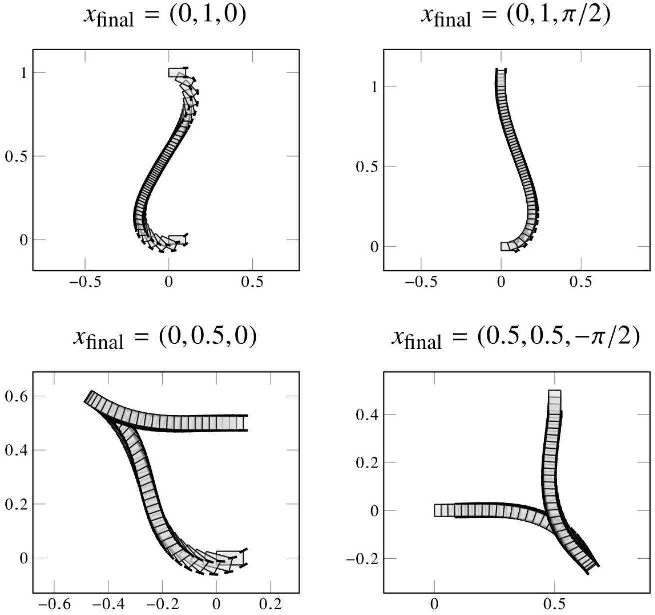
Kontrol problemi

- arabayı verilen başlangıç konumu ve yöneliminden istenen son konuma ve yönelime doğru hareket ettirmek istiyoruz
- bu işlemi küçük ve yavaş değişen bir giriş dizisi kullanarak yapmak istiyoruz
- bu, bir kısıtlı doğrusal olmayan en küçük kareler problemidir:

$$
\begin{array}{rl}
\text{minimize} & \displaystyle\sum_{k=1}^{N} \|u_k\|^2 + \gamma \sum_{k=1}^{N-1} \|u_{k+1} - u_k\|^2 \\
\text{bağlı} & x_1 = x_{\text{başlangıç}} = 0 \\
& k = 1, 2, \ldots, N-1 \text{ için :} \\
& \quad x_{k+1} = f(x_k, u_k) \\
& x_{\text{son}} = f(x_N, u_N)
\end{array}
$$

- problemin değişkenleri: $x_2, \ldots, x_N$, $u_1, \ldots, u_N$

Dört çözüm yörüngesi



Dört çözüm yörüngesi için giriş dizileri