

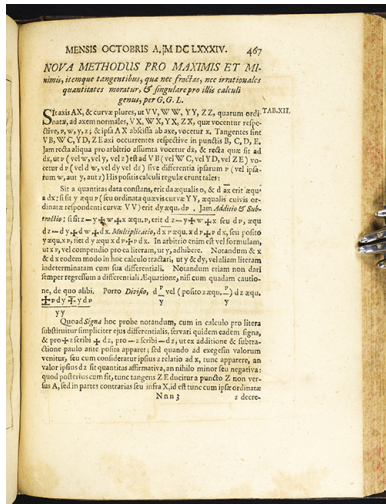
# Yapay Öğrenmede Optimizasyon: Temel Fikirler

İlker Birbil

Sabancı Üniversitesi  
Mühendislik ve Doğa Bilimleri Fakültesi  
Endüstri Mühendisliği Programı

Nesin Matematik Köyü  
Şirince - Nisan, 2017

- Önceden belirlenmiş kısıtlar altında bir fonksiyonun en büyük ya da en küçük değerini bulmak.
- Bu derste problemin *düzgünce* tanımlı olduğunu varsayacağız. Yani bir minimum (ya da maksimum) noktasının olduğu problemlerle ilgileneceğiz.



Şekil: Leibniz'in 1684 makalesi.

# Matematiksel Programlama Modeli

Genel bir **optimizasyon (eniyeleme) problemi** şu şekilde gösterilebilir:

$$\begin{array}{ll} \text{enküçüle} & f(x) \\ \text{öyle ki} & c_j(x) = 0, \quad j \in \mathcal{E}, \\ & c_j(x) \geq 0, \quad j \in \mathcal{I}. \end{array} \quad (1)$$

Burada  $x \in \mathbb{R}^n$  vektörü ile **karar değişkenleri (bilinmeyenler)**,  $f : \mathbb{R}^n \mapsto \mathbb{R}$  ile **amaç fonksiyonu**,  $c_j : \mathbb{R}^n \mapsto \mathbb{R}$ ,  $j \in \mathcal{E} \cup \mathcal{I}$  ile de **kısıtlar** gösterilmiştir.

## Not

Bir enbüyükleme problemi kolayca enküçükleme problemine dönüştürülebilir:

$$\max f(x) = -\min\{-f(x)\}.$$

İki kısıtlı bir matematiksel model şu şekilde<sup>1</sup> verilmiş olsun:

$$\begin{array}{ll} \text{enküçüle} & f(x) \\ \text{öyle ki} & x_1^2 - x_2 \leq 0, \\ & x_1 + x_2 \leq 2. \end{array}$$

Burada

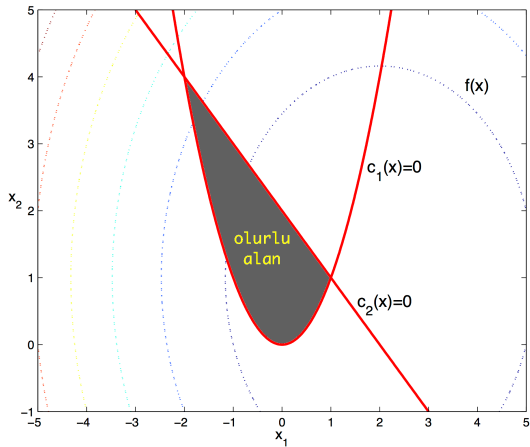
$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \end{bmatrix} = \begin{bmatrix} -x_1^2 + x_2 \\ -x_1 - x_2 + 2 \end{bmatrix}, \quad \mathcal{I} = \{1, 2\}, \quad \mathcal{E} = \emptyset.$$

---

<sup>1</sup>Nocedal, J., Wright, S. J., Numerical Optimization, 2. Basım, New York:Springer, 2006.  
(Notları hazırlarken bu kaynaktan sık sık yararlandım - İlker)

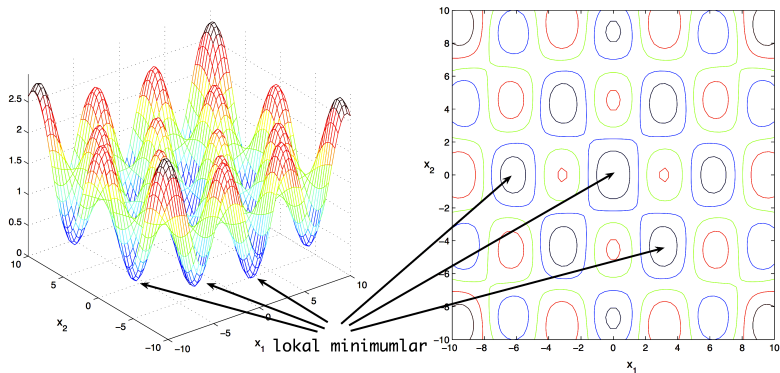
## Örnek (devam)



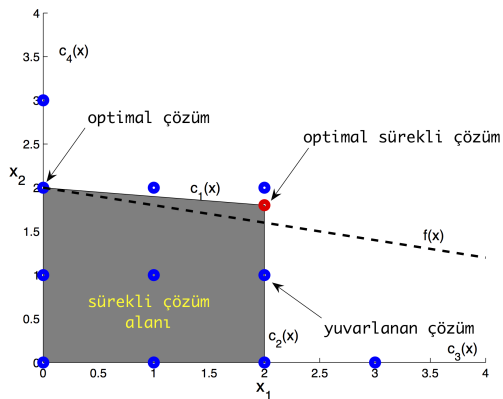
Amaç fonksiyonunun, kısıtların ve değişkenlerin özelliklerine göre problemleri ya da algoritmaları sınıflandırabiliriz:

- ▶ **Sürekli** (continuous) ya da kesikli (discrete) optimizasyon
- ▶ Kısıtlı (constrained) ya da **kısıtsız** (unconstrained) optimizasyon
- ▶ Global ve **lokal** optimizasyon.
- ▶ Rassal (stochastic) ve **belirli** (deterministic) optimizasyon.

# Lokal ve Global Optimizasyon



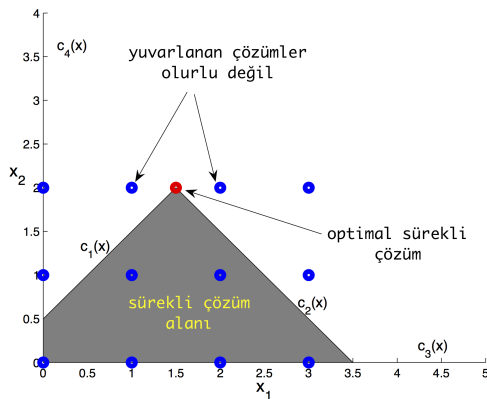
# Kesikli ve Sürekli Optimizasyon



$$\max\{x_1 + 5x_2 : x_1 + 10x_2 \leq 20, x_1 \leq 2, x_1 \geq 0, x_2 \geq 0, x_1, x_2 \in \mathbb{Z}\}.$$



## Kesikli ve Sürekli Optimizasyon (devam)



$$\max\{x_2 : -x_1 + x_2 \leq 1/2, x_1 + x_2 \leq 7/2, x_1 \geq 0, x_2 \geq 0, x_1, x_2 \in \mathbb{Z}\}.$$

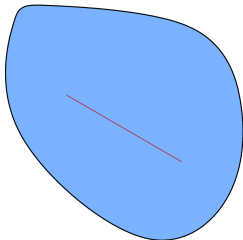
# Dışbükey Küme

## Tanım

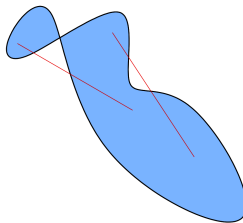
$S \in \mathbb{R}^n$  kümesinin dışbükey (convex) olması için  $S$  kümesinden herhangi iki noktayı birleştiren doğru parçasının tamamının  $S$  kümesinde olması gerekir. Matematiksel olarak gösterirsek, her  $x, y \in S$  çifti ve tüm  $\alpha \in [0, 1]$  değerleri için

$$\alpha x + (1 - \alpha)y \in S$$

olmalıdır.



Dışbükey küme



Dışbükey olmayan küme

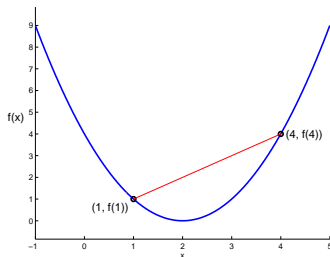
# Dışbükey Fonksiyon

## Tanım

$f(\cdot)$  ile gösterilen bir fonksiyonun **dışbükey** olması için tanım kümesinin dışbükey olması ve bu kümeden seçilen herhangi  $x, y$  çifti için  $f(\cdot)$  grafiğinin  $(x, f(x))$  ve  $(y, f(y))$  noktalarını birleştiren doğru parçasının altında kalması gerekir. Matematiksel olarak ifade edersek, her  $x, y$  ve tüm  $\alpha \in [0, 1]$  değerleri için

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

eşitsizliği sağlanmalıdır.



# Dışbükeylik Hakkında

- ▶ Bir  $f(\cdot)$  fonksiyonunun içbükey (concave) olması,  $-f(\cdot)$  fonksiyonunun dışbükey olduğunu gösterir.
- ▶ Dışbükey bir fonksiyon ile yazılan kısıtsız problemlerde lokal minimum noktası, global minimum noktası olur.
- ▶ Eğer bir kısıt  $\leq$  şeklinde bir eşitsizlikse ve dışbükey fonksiyonlar ile oluşturulmuşsa, ortaya çıkan olurlu alan da dışbükey bir kümedir.
- ▶ Kabaca söylemek gerekirse, pek çok durumda dışbükey fonksiyonlar ile çalışıldığında minimum noktasını belirlemek için kullanılan gerek şartlar aynı zamanda yeter şartlar olurlar.
- ▶ Dışbükey fonksiyonlar, lokal olarak daha karmaşık ve dışbükey olmayan fonksiyonların yaklaşık gösteriminde kullanılırlar.

## Dışbükeylik Hakkında (devam)

En başta (1) ile gösterdiğimiz matematiksel programlama modelinde

- ▶ amaç fonksiyonu  $f(\cdot)$  dışbükeyse,
- ▶ eşitlik  $c_j(\cdot), j \in \mathcal{E}$  kısıtları doğrusalsa,
- ▶ ve eşitsizlik fonksiyonları  $c_j(\cdot), j \in \mathcal{I}$  içbükeyse,

elde edilen model **dışbükey optimizasyon** modeli olur.

enküçüle

$f(x)$

öyle ki

$$c_j(x) = 0, \quad j \in \mathcal{E},$$

$$c_j(x) \geq 0, \quad j \in \mathcal{I}.$$

# Kısıtsız Optimizasyon

Kısıtsız optimizasyon problemlerinde eşitlikler ve eşitsizlikler yoktur. Yani (1) modelinde  $\mathcal{I} \equiv \mathcal{E} \equiv \emptyset$  olarak alınır. Bu durumda model kısaca

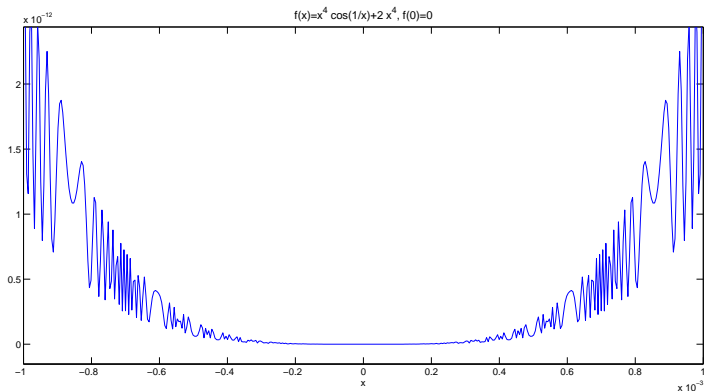
$$\min_{x \in \mathbb{R}^n} f(x)$$

olarak yazılabilir.

## Tanım

- ▶ Eğer tüm  $x$  değerleri için  $f(x^*) \leq f(x)$  eşitsizliği sağlanırsa,  $x^*$  noktası **global minimum** noktasıdır.
- ▶ Eğer  $x^*$  noktasının  $\mathcal{N}$  ile gösterilen komşuluğunda tüm  $x \in \mathcal{N}$  için  $f(x^*) \leq f(x)$  eşitsizliği sağlanırsa,  $x^*$  noktası **lokal minimum** noktasıdır.
- ▶ Eğer  $x^*$  noktasının  $\mathcal{N}$  ile gösterilen komşuluğunda tüm  $x \in \mathcal{N}$  ve  $x \neq x^*$  için  $f(x^*) < f(x)$  eşitsizliği sağlanırsa,  $x^*$  noktası **kati (strict) lokal minimum** noktasıdır.
- ▶ Eğer  $x^*$  noktası  $\mathcal{N}$  ile gösterilen komşuluğundaki tek lokal minimum noktası ise,  $x^*$  noktası **ayrık (isolated) lokal minimum** noktasıdır.

## Kısıtsız Optimizasyon (devam)



Şekil:  $x^* = 0$  noktası kati ama ayırık olmayan bir lokal minimum.

## Çok Boyutlu Uzay

Vektörlerle çalışan ve bir gerçekte sayı döndüren fonksiyonlar  $f : \mathbb{R}^n \mapsto \mathbb{R}$  şeklinde gösterilir. Birinci türevi elde etmek için  $n$  boyutun her birine göre kısmi türev alınarak **gradyant (gradient) vektörü** elde edilir:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

İkinci türevleri ise **Hesyan (Hessian) matrisini** verecektir:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

Görüldüğü üzere  $\nabla f : \mathbb{R}^n \mapsto \mathbb{R}^n$  ve  $\nabla^2 f : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$  fonksiyonlarını elde ediyoruz.



# Taylor Teoremi

Kısıtsız optimizasyondaki gerek ve yeter şartları ispatlamak için Taylor teoremini kullanmak yeterlidir.

## Teorem

Sürekli türevlenebilir bir fonksiyonu  $f : \mathbb{R}^n \mapsto \mathbb{R}$  ile gösterelim. Herhangi bir  $p \in \mathbb{R}^n$  vektörü için

$$f(x + p) = f(x) + \nabla f(x + tp)^\top p \quad (2)$$

eşitliğini sağlayan en az bir  $t \in (0, 1)$  değeri vardır. Eğer  $f(\cdot)$  fonksiyonu iki kez türevlenebilir ise, benzer şekilde

$$f(x + p) = f(x) + \nabla f(x)^\top p + \frac{1}{2} p^\top \nabla^2 f(x + tp) p \quad (3)$$

eşitliğini sağlayan en az bir  $t \in (0, 1)$  değeri bulunabilir. Bu eşitlik aynı zamanda

$$f(x + p) = f(x) + \nabla f(x)^\top p + \frac{1}{2} p^\top \nabla^2 f(x) p + o(\|p\|^2) \quad (4)$$

olarak da yazılabilir.

# Gerek Şartlar

## Teorem (Birinci Dereceden Gerek Şartlar)

Eğer  $x^* \in \mathbb{R}^n$  bir lokal minimum ve  $\nabla f(\cdot)$  fonksiyonu  $x^*$  noktasının açık komşuluğunda sürekli ise,  $\nabla f(x^*) = 0$  eşitliği sağlanır.

Burada  $\nabla f(x^*) = 0$  eşitliğini sağlayan  $x^*$  noktasına **kararlı (stationary) nokta** denir.

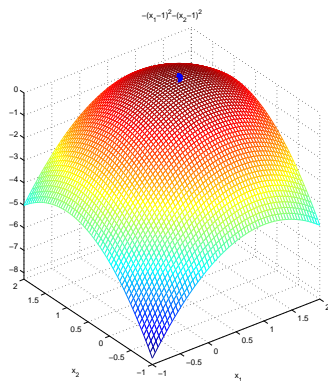
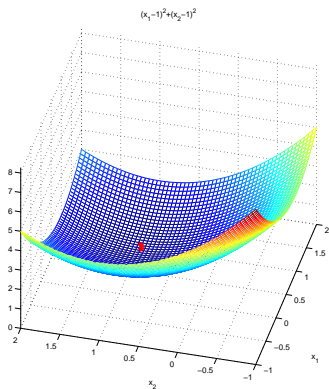
## Teorem (İkinci Dereceden Gerek Şartlar)

Eğer  $x^* \in \mathbb{R}^n$  bir lokal minimum ve  $\nabla^2 f(\cdot)$  fonksiyonu  $x^*$  noktasının açık komşuluğunda sürekli ise,  $\nabla f(x^*) = 0$  eşitliği sağlanır ve  $\nabla^2 f(x^*)$  matrisi **pozitif yarı belirli (positive semi-definite)** olur. Yani, tüm  $p \in \mathbb{R}^n$  vektörleri için

$$p^T \nabla^2 f(x^*) p \geq 0$$

eşitsizliğini sağlar.

## Gerek Şartlar (devam)



**Şekil:** Her iki durumda da birinci dereceden gerek şartlar geçerliyken, ikinci dereceden gerek şartlar sadece ilk durumda sağlanır.

# Yeter Şartlar

## Teorem (İkinci Dereceden Yeter Şartlar)

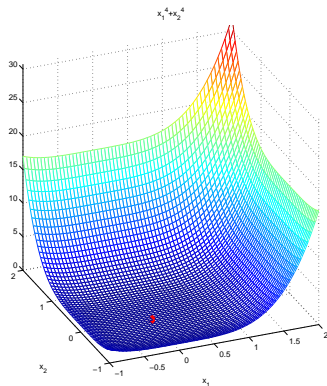
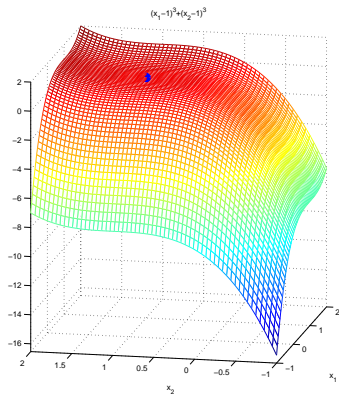
Eğer  $\nabla^2 f(\cdot)$  fonksiyonu  $x^*$  noktasının açık komşuluğunda sürekliyse,  $\nabla f(x^*) = 0$  eşitliği sağlanırsa ve  $\nabla^2 f(x^*)$  matrisi **pozitif belirli (positive definite)**<sup>2</sup> ise,  $x^*$  noktası **kati** lokal minimumdur.

## Teorem

Eğer  $f(\cdot)$  fonksiyonu dışbükey ise, herhangi bir lokal minimum noktası **global** minimumdur. Dışbükey fonksiyon ayrıca türevlenebilir ise herhangi bir kararlı nokta **global** minimum olur.

<sup>2</sup>Tüm  $p \in \mathbb{R}^n$  vektörleri için  $p^T \nabla^2 f(x^*) p > 0, p \neq 0$ .

## Yeter Şartlar (devam)



**Şekil:** Her iki durumda da ikinci dereceden gerek şartlar sağlanırken, yeter şartlar geçerli değildir.

# Kısıtsız Optimizasyon Algoritmaları

- ▶ Bu algoritmalarda bir dizi adım hesaplanır:  $x_0, x_1, x_2, \dots$
- ▶ Çoğu zaman başlangıç noktası  $x_0$  kullanıcı tarafından belirlenir.
- ▶ Her adımda algoritma  $f(\cdot)$ ,  $\nabla f(\cdot)$  ya da  $\nabla^2 f(\cdot)$  fonksiyonlarını kullanır.
- ▶ Pek çok algoritmada  $\{f(x_i)\}_{i=0}^{\infty}$  dizisi monoton olarak azalır. Ancak monoton olmayan algoritmalar da mevcuttur.
- ▶ Kısıtsız optimizasyon için iki temel strateji vardır: **doğru arama (line search)** ve **güven bölgesi (trust region)**.

## Kısıtsız Optimizasyon Algoritmaları (devam)

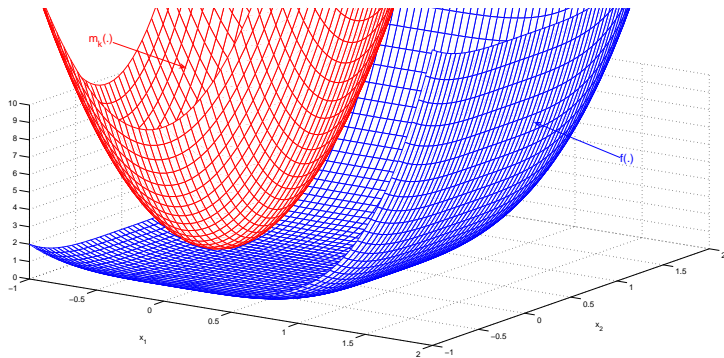
- **Doğru Arama:** Algoritmanın  $k$  adımında  $p_k$  yönü seçilir ve o yönde arama yapılır. Bu durumda temel mesele **adım büyüklüğü (step length)**  $\alpha$  değerini belirlemektir. Adım büyüklüğü yapay öğrenme camiasında **öğrenme hızı (learning rate)** olarak da bilinir.
- **Güven Bölgesi:** Her  $k$  adımında, asıl fonksiyon  $f(\cdot)$  yerine bir lokal **model fonksiyon**  $m_i(\cdot)$  oluşturulur ve bu fonksiyonun minimumu hesaplanır. Model fonksiyonu lokal olduğu için minimumu sadece  $x_i$  çevresinde belirlenen bir güven bölgesi içinde aranır. Örneğin

$$\min_{p \in \mathbb{R}^n} \{m_i(x_i + p) : \|p\|_2 \leq \Delta\}. \quad (5)$$

Burada  $\Delta > 0$  güven bölgesinin yarıçapıdır. Eğer bu problemin çözümüyle elde edilen sonuç asıl fonksiyonda istenen azaltmayı sağlamazsa yarıçap küçültülür. Genelde  $m_i(\cdot)$  karesel (quadratic) bir fonksiyon olarak alınır:

$$m_i(x_i + p) = f(x_i) + p^T \nabla f(x_i) + \frac{1}{2} p^T B_i p. \quad (6)$$

## Kısıtsız Optimizasyon Algoritmaları (devam)





## Doğru Arama Algoritmaları

Taylor teoremini kullanırsak

$$f(x_i + \alpha p) = f(x_i) + \underbrace{\alpha p^T \nabla f(x_i)}_{<0} + O(\alpha^2)$$

eşitliğini elde ederiz. Buna göre  $p$  ve  $-\nabla f(x_i)$  arasındaki  $\theta$  açısı  $\pi/2$  değerinden küçük olmalı ki

$$p^T \nabla f(x_i) = \|p\| \|\nabla f(x_i)\| \cos \theta < 0 \quad (7)$$

sağlansın. Böyle bir  $p$  vektörüne **iniş yönü (descent direction)** denir. Bu yönde atılacak ufak bir adım ile fonksiyonun değeri düşürülebilir. Doğal olarak en fazla düşüşü verecek  $p$  vektörünü bulmak isteriz:

$$\min_{p \in \mathbb{R}^n} \{p^T \nabla f(x_i) : \|p\| = 1\}.$$

Bu problemin optimal çözümü (7) sayesinde

$$p = \frac{-\nabla f(x_i)}{\|\nabla f(x_i)\|}$$

olarak elde edilir.

## Doğru Arama Algoritmaları (devam)

- En dik iniş (steepest descent) algoritması ya da diğer adıyla **gradyant ya da bayır iniş (gradient descent) algoritması**  $p_i = -\nabla f(x_i)$  yönünü kullanır.
- **Newton algoritması** ise amaç fonksiyonunun ikinci dereceden Taylor serisi açılımını kullanarak karesel bir fonksiyon ile yakınsama yapar:

$$f(x_i + p) \approx f(x_i) + p^T \nabla f(x_i) + \frac{1}{2} p^T \nabla^2 f(x_i) p \equiv m_i(p).$$

Elde edilen  $m_i(\cdot)$  fonksiyonun minimumu basitçe türevi alınarak hesaplanır. Eğer  $\nabla^2 f(x_i)$  matrisinin **tersi varsa**, optimal yön

$$p_i^N = -\nabla^2 f(x_i)^{-1} \nabla f(x_i)$$

olarak bulunur. Eğer  $\nabla^2 f(x_i)$  matrisi **pozitif belirli** ise,  $p_i^N$  vektörü iniş yönü olacaktır:

$$\nabla f(x_i)^T p_i^N = -p_i^N \nabla^2 f(x_i) p_i^N < 0.$$

## Doğru Arama Algoritmaları (devam)

- ▶ Newton yönü iki açıdan başa bela olabilir:
  - ▶  $\nabla^2 f(x_i)$  matrisinin tersi olmayabilir.
  - ▶  $\nabla^2 f(x_i)$  matrisi pozitif belirli olmayabilir.
- ▶ Newton algoritmasının dezavantajı her adımda  $\nabla^2 f(\cdot)$  matrisinin hesaplanmasıdır.

### Not

Her adımda matrisin tersini hesaplamamanın önüne geçmek için **Newton-benzeri (quasi-Newton) algoritmalar** geliştirilmiştir. Bu algoritmalar, geçmiş adımlardaki birinci türevleri kullanarak Hesyan matrisine yakın olan bir  $B_i$  matrisi hesaplarlar. Hesaplama biçimleri nedeniyle de bir sonraki adımdaki matrisin tersi ( $B_{i+1}^{-1}$ ), eldeki matrisin tersi ( $B_i^{-1}$ ) ile kolayca bulunabilir.

Newton-benzeri algoritmalar için Matematik Dünyası'nın son sayısına bakılabilir:

Hesaplamalı Tarifler I: Newton ve Benzeri Metodlar

## Doğru Arama Algoritmaları (devam)

### Gradyant İniş

$$x_{k+1} = x_i - \alpha_i \nabla f(x_i)$$

### Newton-benzeri Algoritmalar

$$x_{k+1} = x_i - \alpha_i B_i^{-1} \nabla f(x_i)$$

### Newton Algoritması

$$x_{k+1} = x_i - \alpha_i \nabla^2 f(x_i)^{-1} \nabla f(x_i)$$

## Güven Bölgesi Algoritmaları

Daha önce (6) ile gösterdiğimiz model fonksiyonunda ikinci terimi almazsak, yani  $B_i = 0$  olursa, çözmemiz gereken problem

$$\min_{p \in \mathbb{R}^n} \{f(x_i) + p^\top \nabla f(x_i) : \|p\|_2 \leq \Delta_i\}$$

haline gelir. Bu problemin minimum değeri kolayca

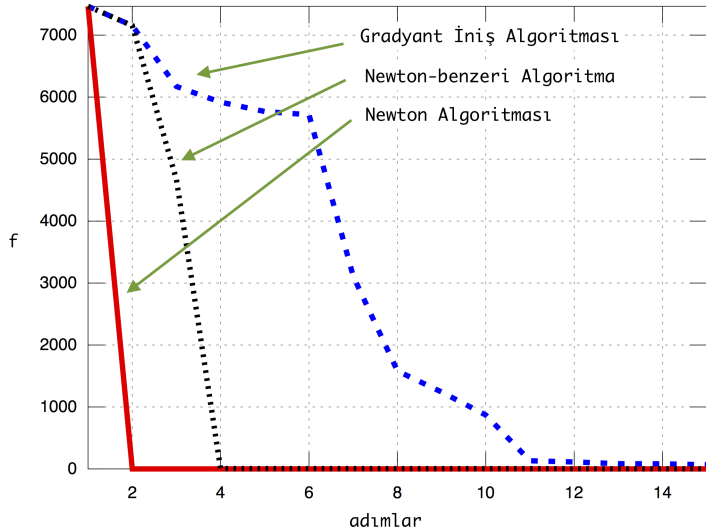
$$p_i = -\Delta_i \frac{\nabla f(x_i)}{\|\nabla f(x_i)\|},$$

olarak bulunur. Hesaplanan bu  $p_i$  vektörü en dik iniş yönü ile aynıdır. Güven bölgesi algoritmaları ile doğru arama algoritmaları arasında bunun gibi benzerlikler halihazırda gösterilmiştir.

**Güven bölgesi Newton algoritması**  $B_i$  matrisi yerine doğrudan  $\nabla^2 f(x_i)$  matrisini kullanır. Burada  $B_i$  matrisinin pozitif belirli olmasına ihtiyaç yoktur.

**Güven bölgesi Newton-benzeri algoritmalar** ise, tahmin edileceği üzere,  $B_i$  matrisi yerine önceki türev bilgisini kullanarak yaklaşık bir Hesyan matrisi hesaplarlar.

## Çözümeye Yakınsama Hızları



## UYGULAMA

Aşağıda verilen iki boyutlu fonksiyonu, gradyant iniş algoritması, Newton algoritması ve Newton-benzeri algoritmalar ile çözelim.

### Rosenbrock Fonksiyonu

$$\min_{x \in \mathbb{R}^2} 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Türevleri:

$$\nabla f(x) = \begin{bmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix},$$

$$\nabla^2 f(x) = \begin{bmatrix} 800x_1^2 - 400(x_2 - x_1^2) + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}.$$