VERİ MADENCİLİĞİ

Farklı Sınıflandırma Yöntemleri

Yrd. Doç. Dr. Şule Gündüz Öğüdücü
http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=195/

Konular

- Sınıflandırma yöntemleri
  - Örnek tabanlı yöntemler
    - k-En Yakın Komşu Yöntemi
  - Genetik Algoritmalar
  - Destek Vektör Makineleri
  - Bulanık Küme Sınıflandırıcılar
  - Öngörü
    - Eğri Uydurma
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Örnek Tabanlı Yöntemler

- Örnek tabanlı sınıflandırma:
  - Öğrenme kümesi saklanır
  - Sınıflandırılacak yeni bir örnek geldiğinde öğrenme kümesi sınıf etiketini öngörmek için kullanılır (tembel (lazy) yöntemler)
- Yöntemler
  - k-en yakın komşu yöntemi

Örnek Tabanlı Yöntemler

Öğrenme Kümesi

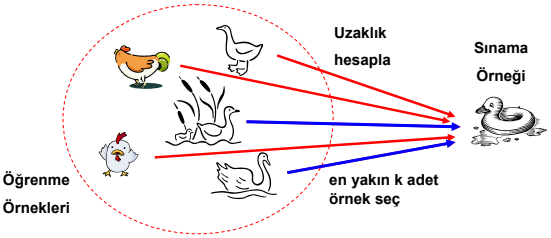
Nit1	.....	NitN	Sınıf
			A
			B
			B
			C
			A
			C
			B

Yeni örnek

Nit1	.....	NitN

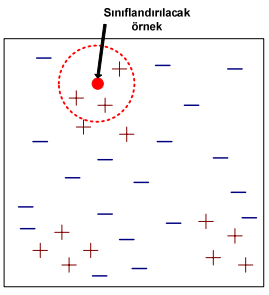
En Yakın Komşu Yöntemi

- Temel yaklaşım: Sınıflandırılmak istenen örneğe en yakın örnekleri bul.
  Örnek: ördek gibi yürüyor, ördek gibi bağırıyor
  => büyük olasılıkla ördek

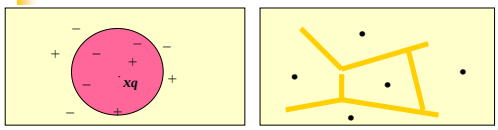


En Yakın Komşu Sınıflandırıcı

- Bütün örnekler n-boyutlu uzayda bir noktaya karşı düşürülür
- Nesneler arasındaki uzaklık (Öklid uzaklığı)
- Öğrenilen fonksiyon ayrık değerli veya gerçel değerli olabilir
- Ayrık değerli fonksiyonlarda k-komşu algoritması Xq örneğine en yakın k öğrenme örneğinde en çok görülen sınıf değerini verir
- Sürekli değerli fonksiyonlarda en yakın k öğrenme örneğinin ortalaması alınır



K-En Yakın Komşu Yöntemi



- *Xq* örneği 1-en yakın komşuya göre pozitif olarak, 5-en yakın komşuya göre negatif olarak sınıflandırılır
- Voronoi diyagramları: Her öğrenme örneğini çevreleyen dışbükey çokgenlerden oluşan karar yüzeyi

K-En Yakın Komşu Yöntemi

- Uzaklık-ağırlıklı k-en yakın komşu algoritması
  - Öğrenme kümesindeki örneklere (x_i), sınıflandırılmak istenen örneğe (x_q) olan uzaklıklarına göre ağırlıklar verilmesi
  - yakın örneklerin ağırlığı daha fazla $w \equiv \frac{1}{d(x_q, x_i)^2}$
- k-en yakın komşunun ortalaması alındığı için gürültülü veriden az etkileniyor
- İlgisiz nitelikler uzaklığı etkileyebilir
  - bu nitelikler uzaklık hesaplarken kullanılmayabilir

Konular

- Sınıflandırma yöntemleri
  - Örnek tabanlı yöntemler
    - k-En Yakın Komşu Yöntemi
  - Genetik Algoritmalar
  - Destek Vektör Makineleri
  - Bulanık Küme Sınıflandırıcılar
  - Öngörü
    - Eğri Uydurma
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Genetik Algoritmalar

- Optimizasyon amaçlı
- Bir başlangıç çözümü öneriyor, tekrarlanan her ara adımda daha iyi çözüm üretmeye çalışıyor.
- Doğal evrime ve en iyi olanın yaşamını sürdürmesine dayanıyor
- Çözümü birey olarak sunuyor.
- Birey: $I=I_1,I_2,\ldots,I_n$ – I_j kullanılan alfabenin bir karakteri
- gen: I_j
- Toplum: Bireylerden oluşan küme

Genetik Algoritmalar

- Genetik Algoritmalar (GA) 5 parçadan oluşuyor:
  - Bireylerden oluşan bir başlangıç kümesi, P
  - Çaprazlama (Crossover): Bir anne babadan yeni bireyler üretmek için yapılan işlem
  - Mutasyon: Bir bireyi rastgele değiştirme
  - Uygunluk (fitness): En iyi bireyleri belirleme
  - Çaprazlama ve mutasyon tekniklerini uygulayan ve uygunluk fonksiyonuna göre toplum içindeki en iyi bireyleri seçen algoritma

Çaprazlama Örnekleri

Parents	Children
000 | 000	000 | 111
111 | 111	111 | 000

a) Single Crossover

Parents	Children
000 | 000 | 00	000 | 111 | 00
111 | 111 | 11	111 | 000 | 11

a) Multiple Crossover

Genetik Algoritma

```
Input:
   P      //Initial Population
Output:
   P'     //Improved Population
Genetic Algorithm:
                //Illustrates Genetic Algorithm
   repeat
      N = | P |;
      P' = ∅;
      repeat
         i1, i2 = select(P);
         o1, o2 = cross(i1, i2);
         o1 = mutate(o1);
         o2 = mutate(o2);
         P' = P' ∪ {o1, o2};
      until | P' |= N;
      P = P';
   until termination criteria satisfied;
```


GA – Avantajlar, Dezavantajlar

- Avantaj
  - Paralel çalışabilir
  - NP karmaşık problem çözümlerine uygun
- Dezavantaj
  - Son kullanıcının modeli anlaması güç
  - Problemi GA ile çözmeye uygun hale getirmek zor
  - Uygunluk fonksiyonunu belirlemek zor
  - Çaprazlama ve mutasyon tekniklerini belirlemek zor

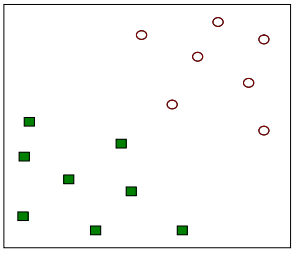
Konular

- Sınıflandırma yöntemleri
  - Örnek tabanlı yöntemler
    - k-En Yakın Komşu Yöntemi
  - Genetik Algoritmalar
  - Destek Vektör Makineleri
  - Bulanık Küme Sınıflandırıcılar
  - Öngörü
    - Eğri Uydurma
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Destek Vektör Makineleri

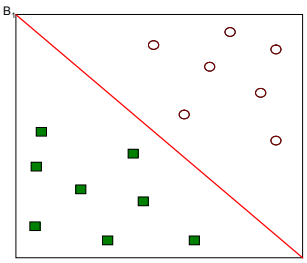
- Hem doğrusal olarak ayırt edilebilen hem de edilemeyen veri kümesini sınıflandırabilir
- Doğrusal olmayan bir eşlem ile n boyutlu veri kümesi m > n olacak şekilde m boyutlu yeni bir veri kümesine dönüştürülür
- Yüksek boyutta doğrusal sınıflandırma işlemi yapılır
- Uygun bir dönüşüm ile her zaman veri bir hiper düzlem ile iki sınıfa ayrılabilir
- Hiper düzleme en yakın öğrenme verileri destek vektörleri olarak adlandırılır

Destek Vektör Makineleri



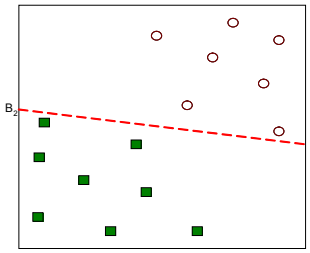
- Destek Vektör Makineleri (Support Vector Machines SVM): Veriyi ayıracak doğrusal bir sınır

Destek Vektör Makineleri



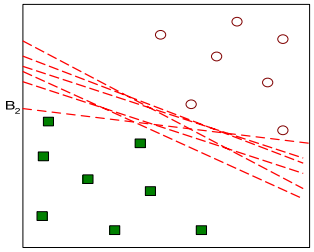
- Bir çözüm

Destek Vektör Makineleri



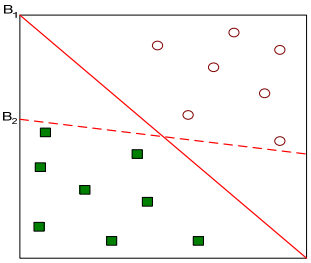
- Başka bir çözüm

Destek Vektör Makineleri



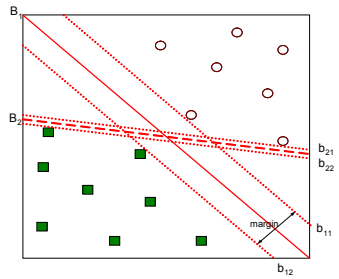
- Diğer çözümler

Destek Vektör Makineleri



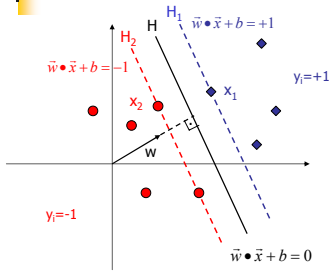
- Hangisi daha iyi? B1 mi, B2 mi?
- Daha iyi nasıl tanımlanır?

Destek Vektör Makineleri



- Farklı sınıftan örnekler arasındaki uzaklığı enbüyüten hiper düzlemi bul => B1, B2'den daha iyi

Destek Vektör Makineleri



$(\mathbf{w}\cdot\mathbf{x}) + b \geq +1$, $y_i=+1$
$(\mathbf{w}\cdot\mathbf{x}) + b \leq -1$, $y_i=-1$
$\Rightarrow y_i(\mathbf{w}\cdot\mathbf{x} + b) \geq +1$

$(\mathbf{w}\cdot\mathbf{x_1})+b=y_i=+1$
$(\mathbf{w}\cdot\mathbf{x_2})+b=y_i=-1$

$\Rightarrow \mathbf{w}\cdot(\mathbf{x_1}-\mathbf{x_2})=2$
$\Rightarrow \frac{\mathbf{w}}{||\mathbf{w}||}\cdot(\mathbf{x_1}-\mathbf{x_2})=\frac{2}{||\mathbf{w}||}$

Destek Vektör Makineleri

- $\frac{2}{||w||^2}$ enbüyük olması isteniyor
  - $y_i(w\cdot x + b) \geq +1$ olacak şekilde $\frac{||w||^2}{2}$ enküçük olmalı
  - kısıtlı eniyileme (constraint optimization) problem

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N}\alpha_i\left[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right]$$

- Problem

$\alpha_1 \ldots \alpha_N$ bulunması
$\Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ en büyük olacak
kısıtlar:
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0, \forall\alpha_i$

Eniyileme Problemi Çözümü

- Çözüm

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^T \mathbf{x_k} \;\forall\; \mathbf{x_k}, \alpha_k \neq 0$$

- Sınıflandırma fonksiyonu

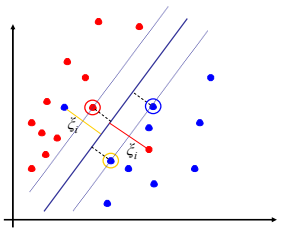
$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x_i}^T \mathbf{x} + b$$

  - f(x) = 1 ise x pozitif olarak, diğer durumlarda negatif olarak sınıflandırılıyor.

Destek Vektör Makineleri

- Öğrenme kümesi doğrusal olarak ayrılamıyor
  - ξ_i değişkenleri ekleniyor

$(\mathbf{w}\cdot\mathbf{x}) + b \geq +1 - \xi_i,\ y_i = +1$
$(\mathbf{w}\cdot\mathbf{x}) + b \leq -1 - \xi_i,\ y_i = -1$
$\Rightarrow y_i(\mathbf{w}\cdot\mathbf{x} + b) \geq +1 - \xi_i$
$\xi_i \geq 0,\ \forall i$



$$Lp = \frac{1}{2}\|w\|^2 - C\left(\sum_{i=1}^{N} \xi_i\right)^k$$

- Problem:
  $\alpha_1 \ldots \alpha_N$ bulunması
  $\Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$ en büyük olacak kısıtlar:
  (1) $\Sigma\alpha_i y_i = 0$
  (2) $0 \leq \alpha_i \leq C,\ \forall \alpha_i$

Eniyileme Problemi Çözümü

- Çözüm

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i}$$
$$b = y_k(1 - \xi_k) - \mathbf{w}^T \mathbf{x_k},\ \mathrm{k} = \mathrm{argmax}\ \alpha_k$$

- Sınıflandırma fonksiyonu

$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x_i}^T \mathbf{x} + b$$

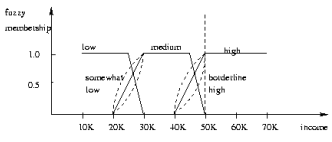
DVM Uygulamaları

- Boser, Guyon ve Vapnik tarafından 1992 yılında önerildi. 1990'ların sonlarına doğru yaygın olarak kullanılmaya başlandı
- DVM için en yaygın eniyileme algoritmaları SMO [Platt '99] ve SVM^{light} [Joachims' 99]

Konular

- Sınıflandırma yöntemleri
  - Örnek tabanlı yöntemler
    - k-En Yakın Komşu Yöntemi
  - Genetik Algoritmalar
  - Destek Vektör Makineleri
  - Bulanık Küme Sınıflandırıcılar
  - Öngörü
    - Eğri Uydurma
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Bulanık Küme Sınıflandırıcılar



- Bulanık mantık 0.0 ve 1.0 arasında gerçel değerler kullanarak üyelik dereceleri hesaplar
- Nitelik değerleri bulanık değerlere dönüştürülür
- Kurallar kümesi oluşturulur
- Yeni bir örneği sınıflandırmak için birden fazla kural kullanılır
- Her kuraldan gelen sonuç toplanır

Konular

- Sınıflandırma yöntemleri
  - Örnek tabanlı yöntemler
    - k-En Yakın Komşu Yöntemi
  - Genetik Algoritmalar
  - Destek Vektör Makineleri
  - Bulanık Küme Sınıflandırıcılar
  - Öngörü
    - Eğri Uydurma
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Öngörü

- Sınıflandırma problemleriyle aynı yaklaşım
  - model oluştur
  - bilinmeyen değeri hesaplamak için modeli kullan
    - eğri uydurma
      - doğrusal
      - doğrusal olmayan
- Sınıflandırma ayrık değerli
- Öngörü sürekli değerli

Eğri Uydurma

- Doğrusal eğri uydurma:
  - en basit eğri uydurma yöntemi
  - veri doğrusal bir eğri ile modellenir.
    - veri kümesindeki niteliklerin doğrusal fonksiyonu

$$y = w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k$$

  - öğrenme kümesindeki y_i sınıfından bir x_i örneği için çıkış

$$y = w_0 x_{i0} + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_k x_{ik} = \sum_{j=0}^{k} w_j x_{ij}$$

  - karesel hatayı enküçültecek ağırlıkları bulma

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k} w_j x_{ij} \right)^2$$

Konular

- Sınıflandırma yöntemleri
- Model Değerlendirme
  - Hata oranı
  - Anma
  - Kesinlik
  - F-ölçütü
  - ROC eğrileri
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Sınıflandırma Modelini Değerlendirme

- Model başarımını değerlendirme ölçütleri nelerdir?
  - Hata oranı
  - Anma
  - Kesinlik
  - F-ölçütü
- Farklı modellerin başarımı nasıl karşılaştırılır?
  - ROC

Sınıflandırma Hatası

- Sınıflandırma yöntemlerinin hatalarını ölçme
  - başarı: örnek doğru sınıfa atandı
  - hata: örnek yanlış sınıfa atandı
  - hata oranı: hata sayısının toplam örnek sayısına bölünmesi
- Hata oranı sınama kümesi kullanılarak hesaplanır

Model Başarımını Değerlendirme

- Model başarımını değerlendirme ölçütleri
  - modelin ne kadar doğru sınıflandırma yaptığını ölçer
    - hız, ölçeklenebilirlik gibi özellikleri değerlendirmez
- Karışıklık matrisi:

		ÖNGÖRÜLEN SINIF	
		Sınıf=1	Sınıf=-1
DOĞRU SINIF	Sınıf =1	a	b
	Sınıf =-1	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Model Başarımını Değerlendirme: Doğruluk

		ÖNGÖRÜLEN SINIF	
		+1	-1
DOĞRU SINIF	+1	a (TP)	b (FN)
	-1	c (FP)	d (TN)

- Modelin başarımı:

$$Dogruluk = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Hata \quad Orani = \frac{b+c}{a+b+c+d} = \frac{FN+FP}{TP+TN+FP+FN}$$

Örnek

Sınıflandırıcı A	
TP=25	FN=25
FP=25	TN=25

Doğruluk=%50

Sınıflandırıcı B	
TP=50	FN=0
FP=25	TN=25

Doğruluk=%75

Sınıflandırıcı C	
TP=25	FN=25
FP=0	TN=50

Doğruluk=%75

- Hangi sınıflandırıcı daha iyi?
  - B ve C, A'dan daha iyi bir sınıflandırıcı
  - B, C'den daha iyi bir sınıflandırıcı mı?

Model Başarımını Değerlendirme: Kesinlik

		ÖNGÖRÜLEN SINIF	
		+1	-1
DOĞRU SINIF	+1	a (TP)	b (FN)
	-1	c (FP)	d (TN)

$$\text{Kesinlik} = \frac{\text{Doğru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif sınıflandırılmış örneklerin sayısı}} = \frac{TP}{TP+FP}$$

Model Başarımını Değerlendirme: Anma

		ÖNGÖRÜLEN SINIF	
		+1	-1
DOĞRU SINIF	+1	a (TP)	b (FN)
	-1	c (FP)	d (TN)

$$\underset{\text{Doğru pozitif oranı}}{\text{Anma}} = \frac{\text{Doğru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif örneklerin sayısı}} = \frac{TP}{TP+FN}$$

Anma / Kesinlik

- A modeli B modelinden daha iyi anma ve kesinlik değerine sahipse A modeli daha iyi bir sınıflandırıcıdır.
- Kesinlik ve anma arasında ters orantı var.

Sınıflandırıcıları Karşılaştırma

- Doğruluk en basit ölçüt
- Kesinlik ve anma daha iyi ölçme sağlıyor
  - Model A'nın kesinliği model B'den daha iyi ancak model B'nin anma değeri model A'dan daha iyi olabilir.

Model Başarımını Değerlendirme: F-ölçütü

- F-ölçütü: Anma ve kesinliğin harmonik ortalamasını alır.

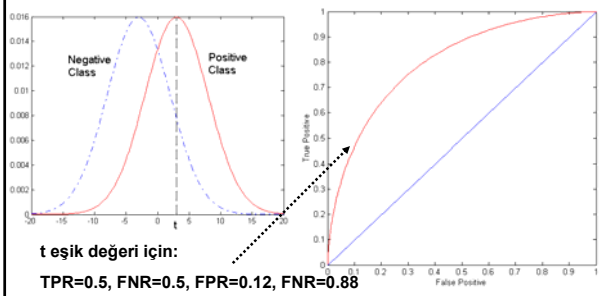
$$\text{F-ölçütü} = \frac{2 * \text{kesinlik} * \text{anma}}{\text{kesinlik} + \text{anma}}$$

ROC (Receiver Operating Characteristic)

- İşaret işlemede bir sezicinin, gürültülü bir kanalda doğru algılama oranının yanlış alarm oranına karşı çizdirilen grafiği (algılayıcı işletim eğrisi)
- Farklı sınıflandırıcıları karşılaştırmak için ROC eğrileri
- Doğru pozitif (TPR - y ekseni) oranının yanlış pozitif (FPR – x ekseni) oranına karşı çizdirilen grafiği
  - TPR = TP / (TP + FN)
  - FPR = FP / (TN + FP)
- ROC üzerindeki her nokta bir sınıflandırıcının oluşturduğu bir modele karşı düşer

ROC Eğrisi

- İki sınıftan oluşan tek boyutlu bir veri kümesi (positive – negative)
- x > t için her örnek pozitif olarak sınıflandırılıyor



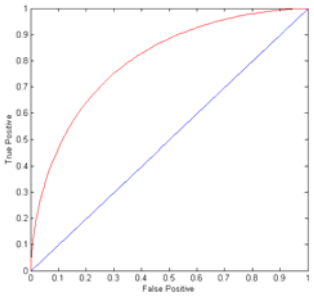
t eşik değeri için:

TPR=0.5, FNR=0.5, FPR=0.12, FNR=0.88

ROC Eğrisi

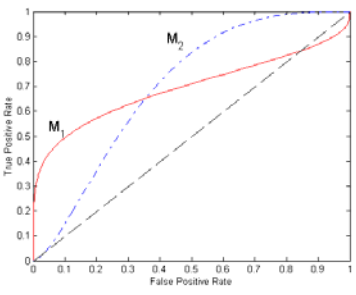
(FPR,TPR)

- (0,0): Bütün örneklerin negatif sınıflandırılması
- (1,1): Bütün örneklerin pozitif sınıflandırılması
- (0,1): ideal durum
- Çapraz çizgi:
  - Rastlantısal tahmin



ROC Eğrilerinin Kullanılması

- Farklı modelleri karşılaştırmak için
- M_1 veya M_2 birbirlerine üstünlük sağlamıyor
  - küçük FPR değerleri için M_1 daha iyi
  - büyük FPR değerleri için M_2 daha iyi
- ROC eğrisi altında kalan alan
  - ideal = 1
  - Rastlantısal tahmin=0.5



ROC Eğrisinin Çizilmesi

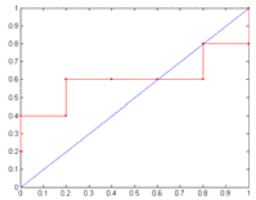
- Her örnek için P(+|A) olasılığı hesaplanır
- P(+|A) değeri azalarak sıralanır
- Her farklı P(+|A) değeri için bir eşik değeri uygulanır
- Her eşik değeri için TP, FP, TN, FN hesaplanır

Örnek	P(+|A)	Sınıf
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

ROC Eğrisinin Çizilmesi

Class	+	-	+	-	-	-	+	-	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

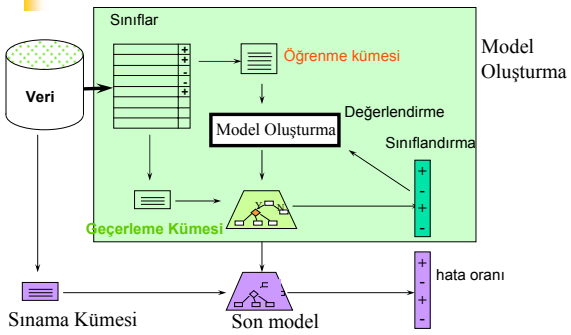
ROC Eğrisi:



Model Parametrelerini Belirleme

- Sınama kümesi sınıflandırıcı oluşturmak için kullanılmaz
- Bazı sınıflandırıcılar modeli iki aşamada oluşturur
  - modeli oluştur
  - parametreleri ayarla
- Sınama kümesi parametreleri ayarlamak için kullanılmaz
- Uygun yöntem üç veri kümesi kullanma: öğrenme, geçerleme, sınama
  - geçerleme kümesi parametre ayarlamaları için kullanılır
  - model oluşturulduktan sonra öğrenme ve geçerleme kümesi son modeli oluşturmak için kullanılabilir

Sınıflandırma: Öğrenme, Geçerleme, Sınama



Model Başarımını Tahmin Etme

- Örnek: Doğruluğu %25 olan bir modelin gerçek başarımı ne kadardır?
  - Sınama kümesinin büyüklüğüne bağlı
- Sınıflandırma (hileli) yazı tura atmaya benziyor
  - tura doğru sınıflandırma (başarı), yazı yanlış sınıflandırma (başarısızlık)
- İstatistikte birbirinden bağımsız olayların başarı ya da başarısızlıkla sonuçlanmaları Bernoulli dağılımı ile modellenir.
- Gerçek başarı oranını belirlemek için istatistikte güven aralıkları tanımlanmıştır.

Güven Aralığı

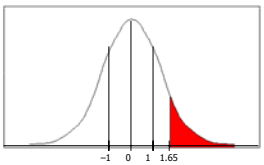
- p belli bir güvenle belli bir aralıkta bulunmaktadır.
- Örnek: N=1000 olayda S=750 başarı sağlanmış.
  - Tahmin edilen başarı oranı: 75%
  - Gerçek başarıya ne kadar yakın
    - %80 güven ile $p \in [73,2 - 76,7]$
- Örnek: N=100 olayda S=75 başarı sağlanmış.
  - Tahmin edilen başarı oranı: 75%
  - Gerçek başarıya ne kadar yakın
    - %80 güven ile $p \in [69,1 - 80,1]$

Ortalama Değer ve Varyans

- Başarı oranı *p* olan bir Bernoulli dağılımının ortalama değeri ve varyansı: *p*, *p* (1–*p*)
- N kere tekrarlanan bir deneyin beklenen başarı oranı *f=S/N*
- Büyük *N* değerleri için, *f* normal dağılım
- *f* için ortalama değer ve varyans: *p*, *p* (1–*p*)/*N*
- Ortalama değeri 0 ve varyansı 1 olan *X* rastlantı değişkeninin *%c* güven aralığı :
  Pr[–*z* ≤ *X* ≤ *z*]=c
- Simetrik bir dağılım için:
  Pr[–*z* ≤ *X* ≤ *z*]=1 – 2*Pr [*X* ≥ *z*]

Güven Sınırları

- Ortalama değeri 0 ve varyansı 1 olan bir normal dağılımın güven sınırları



Pr[*X* ≥ *z*]	*z*
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

- Pr[–*1,65* ≤ *X* ≤ *1,65*]=90%
- *f* in ortalama değerinin 0, varyansının 1 olacak şekilde dönüştürülmesi gerekir.

Dönüşüm

- *f* in ortalama değerinin 0, varyansının 1 olacak şekilde dönüştürülmesi için $\frac{f-p}{\sqrt{p(1-p)/N}}$
- Güven aralığı $\Pr\left[-z \le \frac{f-p}{\sqrt{p(1-p)/N}} \le z\right] = c$
- p'nin değeri $p = \left(f + \frac{z^2}{2N} \pm z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}\right) \Big/ \left(1 + \frac{z^2}{N}\right)$

Örnek

- *f* = 75%, *N* = 1000, *c* = 80% (*z* = 1.28):
  p ∈ [0,732 – 0,767]
- *f* = 75%, *N* = 100, *c* = 80% (*z* = 1.28):
  p ∈ [0,691 – 0,801]

Konular

- Sınıflandırma yöntemleri
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
  - holdout
  - k-kat çapraz geçerleme
  - Bootstrap
- Sınıflandırıcıları birleştirme

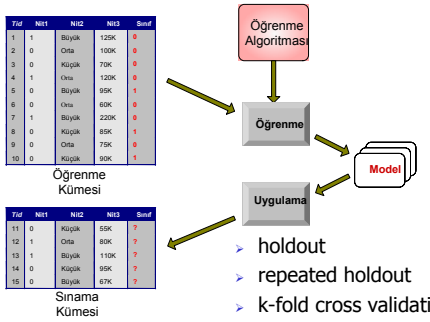
Verinin Dengesiz Dağılımı

- Küçük veya dengesiz veri kümeleri için örnekler tanımlayıcı olmayabilir
- Veri içinde bazı sınıflardan çok az örnek olabilir
  - tıbbi veriler: %90 sağlıklı, %10 hastalık
  - elektronik ticaret: %99 alışveriş yapmamış, %1 alışveriş yapmış
  - güvenlik: %99 sahtekarlık yapmamış, %1 sahtekarlık yapmış
- Örnek: Sınıf1: 9990 örnek, Sınıf2: 10 örnek
  - bütün örnekleri sınıf1'e atayan bir sınıflandırıcının hata oranı: 9990 / 10000= %99,9
    - hata oranı yanıltıcı bir ölçüt olabilir

Dengeli Dağılım Nasıl Sağlanır?

- Veri kümesinde iki sınıf varsa
  - iki sınıfın eşit dağıldığı bir veri kümesi oluştur
  - Az örneği olan sınıftan istenen sayıda rasgele örnekler seç
  - Çok örneği olan sınıftan aynı sayıda örnekleri ekle
- Veri kümesinde iki sınıftan fazla sınıf varsa
  - Öğrenme ve sınama kümesini farklı sınıflardan aynı sayıda örnek olacak şekilde oluştur

Örnek



- holdout
- repeated holdout
- k-fold cross validation
- bootstrapping

Büyük Veri Kümelerinde Değerlendirme

- Veri dağılımı dengeli ise: Veri kümesindeki örnek sayısı ve her sınıfa ait örnek sayısı fazla ise basit bir değerlendirme yeterli
  - *holdout* yöntemi: Belli sayıda örnek sınama için ayrılır, geriye kalan örnekler öğrenme için kullanılır
    - genelde veri kümesinin 2/3'ü öğrenme, 1/3'i sınama kümesi olarak ayrılır
  - öğrenme kümesi kullanılarak model oluşturulur ve sınama kümesi kullanılarak model değerlendirilir

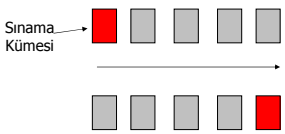
Tekrarlı Holdout Yöntemini

- Veri kümesini farklı altkümelere bölerek holdout yöntemini tekrarlama
  - Her eğitme işleminde veri kümesinin belli bir bölümü öğrenme kümesi olarak rasgele ayrılır
  - Modelin hata oranı, işlemler sonunda elde edilen modellerin hata oranlarının ortalaması
- Problem: Farklı eğitme işlemlerindeki sınama kümeleri örtüşebilir

k-Kat Çapraz Geçerleme

- Veri kümesi eşit boyutta *k* adet farklı gruba ayrılır.

- Bir grup sınama, diğerleri öğrenme için ayrılır.



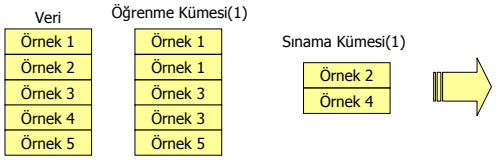
- Her grup bir kere sınama kümesi olacak şekilde deneyler *k* kere tekrarlanır.

Biri Hariç Çapraz Geçerleme

- k-kat çapraz geçerlemenin özel hali
  - k sayısı veri kümesindeki örnek sayısına (N) eşit
- Model *N-1* örnek üzerinde eğitilir, dışarıda bırakılan *1* örnek üzerinde sınanır
- Bu işlem her örnek *1* kez sınama için kullanılacak şekilde tekrarlanır
  - model *N* kez eğitilir
- Model başarımı denemelerin başarımının ortalaması
- Verinin en etkin şekilde kullanımı

Bootstrap Yöntemi

- Veri kümesinden yerine koyma yöntemi ile örnekler seçilerek öğrenme kümesi oluşturulur
  - *N* örnekten oluşan veri kümesinden yerine koyarak *N* örnek seçilir
  - Bu küme öğrenme kümesi olarak kullanılır
  - Öğrenme kümesinde yer almayan örnekler sınama kümesi olarak kullanılır



0.632 bootstrap

- N örnekten oluşan bir veri kümesinde bir örneğin seçilmeme olasılığı: $1-\frac{1}{N}$
- Sınama kümesinde yer alma olasılığı:

$$\left(1-\frac{1}{N}\right)^{N} \approx e^{-1} = 0.368$$

- Öğrenme kümesi veri kümesindeki örneklerin %63,2'sinden oluşuyor

Bootstrap Yönteminde Model Hatasını Belirleme

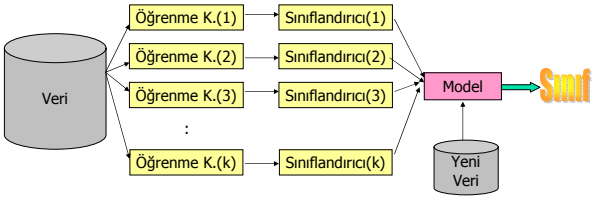
- Model başarımını sadece sınama kümesi kullanarak belirleme kötümser bir yaklaşım
  - model örneklerin sadece ~%63'lük bölümüyle eğitiliyor
- Model başarımı hem öğrenme kümesindeki hem de sınama kümesindeki başarım ile değerlendirilir

  hata = 0,632 $\text{hata}_{(\text{sınama})}$ + 0,368 $\text{hata}_{(\text{öğrenme})}$
- İşlem birkaç kez tekrarlanarak hatanın ortalaması alınır.

Konular

- Sınıflandırma yöntemleri
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme
  - Bagging
  - Boosting

Model Başarımını Artırma



- Bir grup sınıflandırıcı kullanma
  - Bagging
  - Boosting

Bagging

- *N* örnekten oluşan bir veri kümesinde bootstrap yöntemi ile *T* örnek seç
- Bu işlemi *k* öğrenme kümesi oluşturmak üzere tekrarla
- Aynı sınıflandırma algoritmasını *k* öğrenme kümesi üzerinde kullanarak *k* adet sınıflandırıcı oluştur
- Yeni bir örneği sınıflandırmak için her sınıflandırıcının sonucunu öğren
- Yeni örnek en çok hangi sınıfa atanmışsa o sınıfın etiketiyle etiketlendir.

Boosting

- Öğrenme kümesindeki her örneğin bir ağırlığı var
- Her öğrenme işleminden sonra, her sınıflandırıcı için yapılan sınıflandırma hatasına bağlı olarak örneklerin ağırlığı güncelleniyor
- Yeni bir örneği sınıflandırmak için her sınıflandırıcının doğruluğuna bağlı olarak ağırlıklı ortalaması alınıyor.

http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=195/

73