Makine Öğrenmesi

SD413

Benzerlikler: En Yakın Komşu Sınıflandırıcıları

- Birbirine çok benzeyen iki ağaç muhtemelen **aynı tür**dendir.
- Aynı şekilde, benzer semptomlardan şikayet eden hastaların **aynı hastalık**tan muzdarip olmaları beklenir.
- Benzer nesneler genellikle aynı sınıfa aittir.
- **Mantık**: x nesnesinin sınıfını belirlemeniz istendiğinde, ona en çok benzeyen eğitim örneğini bulun ve ardından x'i bu benzer örneğin sınıfıyla etiketleyin.

k-En Yakın Komşu Kuralı

- Belirli bir nesnenin x'e y'den daha fazla benzediğini nasıl tespit ederiz?
- Bunun mümkün olduğundan şüphe duyabilirsiniz. Zürafa, zebradan çok ata benzer diyebilir miyiz? Bunu cevaplamak için çok fazla keyfı ve öznel faktör dikkate alınmalıdır.
- Sınıflandırıcı, gerçek nesnelerden ziyade öznitelik tabanlı açıklamalarını karşılaştırır.

k-En Yakın Komşu Kuralı

Example	Shape	Crust		Filling		Class	# differences
		Size	Shade	Size	Shade		
x	Square	Thick	Gray	Thin	White	?	–
ex_1	Circle	Thick	Gray	Thick	Dark	pos	3
ex_2	Circle	Thick	White	Thick	Dark	pos	4
ex_3	Triangle	Thick	Dark	Thick	Gray	pos	4
ex_4	Circle	Thin	White	Thin	Dark	pos	4
ex_5	Square	Thick	Dark	Thin	White	pos	1
ex_6	Circle	Thick	White	Thin	Dark	pos	3
ex_7	Circle	Thick	Gray	Thick	White	neg	2
ex_8	Square	Thick	White	Thick	Gray	neg	3
ex_9	Triangle	Thin	Gray	Thin	Dark	neg	3
ex_{10}	Circle	Thick	Dark	Thick	White	neg	3
ex_{11}	Square	Thick	White	Thick	Dark	neg	3
ex_{12}	Triangle	Thick	White	Thick	Gray	neg	4

k-En Yakın Komşu Kuralı

- İki pastanın benzerliği, farklı oldukları nitelikleri sayarak belirlenebilir: farklılıklar ne kadar azsa, benzerlik o kadar fazladır.
- En küçük farklılık ex5 örneği için elde edildi. Dolayısıyla x'in ex5 sınıfı olan **pos** ile etiketlenmesi gerektiği sonucuna varıyoruz.
- Tabloda, tüm nitelikler ayrıktır, ancak sürekli niteliklerle uğraşmak da aynı derecede kolaydır. Her örnek, uzayda bir nokta ile temsil edilebileceğinden, **Öklid mesafesini** veya başka bir geometrik formülü kullanabiliriz.
- Örnek uzayında x'e en küçük mesafeye sahip eğitim örneği, geometrik olarak konuşursak, x'in **en yakın komşusudur**.

Tek Komşudan k Komşuya

- Gürültülü domain'lerde, en yakın **tek** komşuya güvenilemez.
- Gürültü nedeniyle sınıf etiketi yanlışsa ne yapacağız?
- Daha sağlam bir yaklaşım, bir değil birkaç en yakın komşuyu belirlemek ve onların oy kullanmalarına izin vermektir.
- Bu, **k-NN** sınıflandırıcısının özüdür; burada **k**, oy veren komşuların sayısıdır. k, genellikle kullanıcı tarafından belirlenen bir parametredir.

Tek Komşudan k Komşuya

- Diyelim ki bir 4-NN sınıflandırıcı, 2-sınıflı bir domain'e uygulandığında, iki komşunun pozitif ve iki komşunun negatif olduğu bir durumla sonuçlanıyor.
- Bu problem nasıl çözülebilir?
- Peki ikiden fazla sınıf varsa?

Sorular

- Tüm özniteliklerin ayrık olduğu alanlarda ve sürekli oldukları alanlarda örnekten örneğe benzerliği nasıl ölçebiliriz?
- Hangi koşullar altında k-NN sınıflandırıcısı (k > 1 ile) 1-NN sınıflandırıcıdan daha iyi performans gösterecek ve neden?
- 2 sınıflı alanlarda, k-NN sınıflandırıcısındaki k'nin neden tek bir sayı olması gerektiğini açıklayın. Bu neden çok sınıflı alanlarda önemsizdir?
- Öznitelik gürültüsü sınırda örneklerin sınıflandırılmasını nasıl etkiler? Sınıf etiketi gürültüsünün etkisi nedir?

Benzerlik Ölçüsü

- Daha önce belirtildiği gibi, bazı x'lerin en yakın komşusunu belirlemenin doğal bir yolu, eğitim örneklerinden x'in **geometrik uzaklıklarını** kullanmaktır.

- İki nitelik varken mesafeler, iki boyutlu bir alan üzerinde bir cetvel yardımıyla kolayca ölçülebilir. Ancak üç veya daha fazla nitelik olduğunda bu pratik değildir.

Öklid Uzaklığı

- İki boyutlu bir uzayda, bir düzlemde, iki nokta arasındaki geometrik uzaklık, x = (x1, x2) ve y = (y1, y2), Pisagor teoremi ile ölçülür.

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\Sigma_{i=1}^{n}(x_i - y_i)^2}$$

Öklid Uzaklığı

- Örnek:

Ex1	1	3	1	Pos
Ex2	3	5	2	Pos
Ex3	3	2	2	Neg
Ex4	5	2	3	Neg
x	**2**	**4**	**2**	**?**

Hamming Uzaklığı

- **Eşit** uzunluktaki iki dize arasındaki, karşılık gelen sembollerin farklı olduğu konumların sayısıdır.

- Başka bir deyişle, bir diziyi diğerine dönüştürmek için gereken minimum değişim sayısını veya bir diziyi diğerine dönüştürebilecek minimum hata sayısını ölçer.

- "karolin" ve "kathrin" >> 3
- 0000 ve 1111 >> 4

Karma Veri

x = (2, 1.5, summer)
y = (1, 0.5, winter)

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(2-1)^2 + (1.5-0.5)^2 + 1} = \sqrt{3}$$

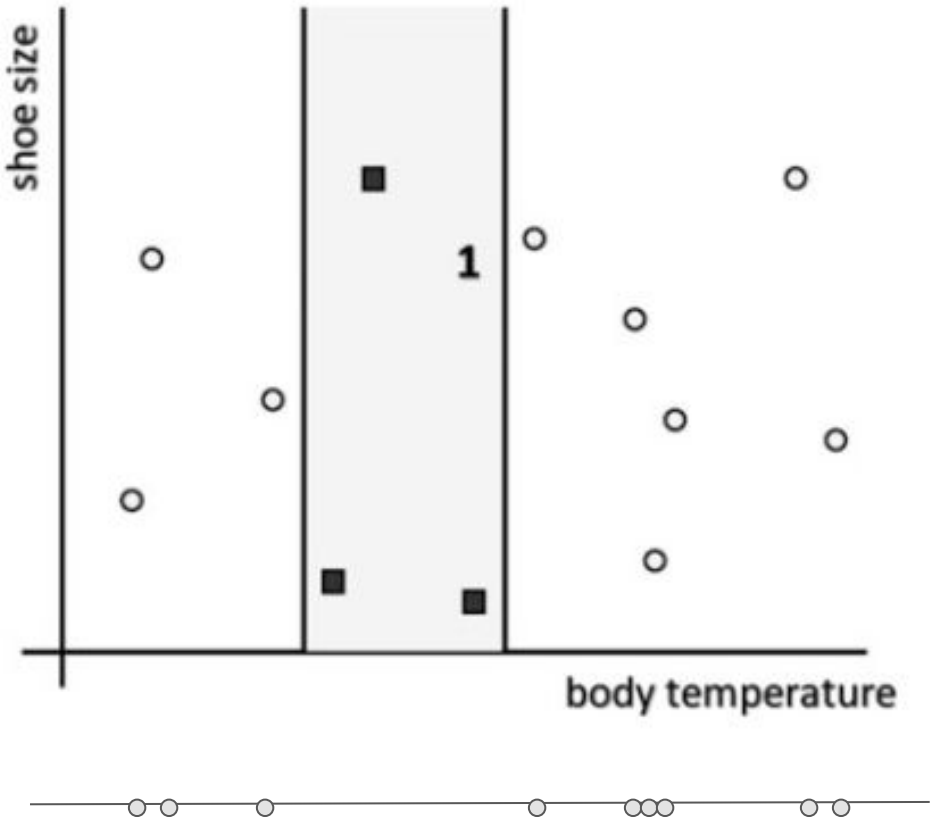
Uzaklık Ölçüleri

Herhangi bir uzaklık ölçüsü aşağıdaki koşulları karşılamalıdır:

1. Uzaklık asla **negatif** olamaz.
1. İki özdeş vektör arasındaki uzaklık, x = y, **sıfırdır**.
1. x'ten y'ye olan uzaklık, y'den x'e olan uzaklık ile **aynıdır**.
1. Uzaklık **üçgen eşitsizliğini** sağlamalıdır: $d(x, y) + d(y, z) \geq d(x, z)$

Alakasız Nitelikler

Bazı nitelikler ala sındaki geometrik mesafeyi etkiler.



Alakasız Nitelikler

Alakasız özelliklerin ne kadar hasara yol açabileceği, bunlardan kaçının kullanıldığına bağlıdır.

Sadece bir tanesinin alakasız olduğu yüzlerce özelliğin bulunduğu bir alanda sorun yoktur.

Bununla birlikte, alakasız özelliklerin yüzdesi artarsa işler daha da kötüleşir. Niteliklerin büyük çoğunluğunun tanımak istediğimiz sınıfla hiçbir ilgisi yoksa, geometrik mesafe neredeyse anlamsız hale gelir.

Niteliklerin Ölçekleri

x = (t, 0.2, 254)
y = (f, 0.1, 194)

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(1-0)^2 + (0.2-0.1)^2 + (254-194)^2}$$

Niteliklerin Ölçekleri

x = (t, 0.2, 254)
y = (f, 0.1, 194)

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(1-0)^2 + (0.2-0.1)^2 + (254-194)^2}$$

Bu ifadeyi incelediğimizde, üçüncü terimin tamamen baskın olduğunu ve diğer ikisini **önemsiz** hale getirdiğini görüyoruz. Değerlerini aralıkları içinde nasıl değiştirirsek değiştirelim, toplam mesafe, dM(x, y), pek etkilenmeyecektir.

Nitelikleri Normalleştirme

- Ölçekle ilgili sorunlardan kurtulmanın bir yolu, öznitelikleri normalleştirmektir: tüm değerleri aynı birim aralığına, [0, 1] düşecek şekilde yeniden ölçeklendirmek.

- Bunu yapmanın en basit yolu, verilen öznitelik için maksimum (MAX) ve minimum (MIN) değerlerini belirlemek ve ardından bu özniteliğin her bir değerini, x'i aşağıdaki formülü kullanarak değiştirmektir:

$$x = \frac{x - MIN}{MAX - MIN}$$

Nitelikleri Normalleştirme

- Örnek:

[7, 4, 25, –5, 10]

MIN = –5 MAX = 25

x–MIN = [12, 9, 30, 0, 15]
MAX – MIN = 30

$$x = \frac{x - MIN}{MAX - MIN}$$ .3, 1, 0, 0.5]

Sorular

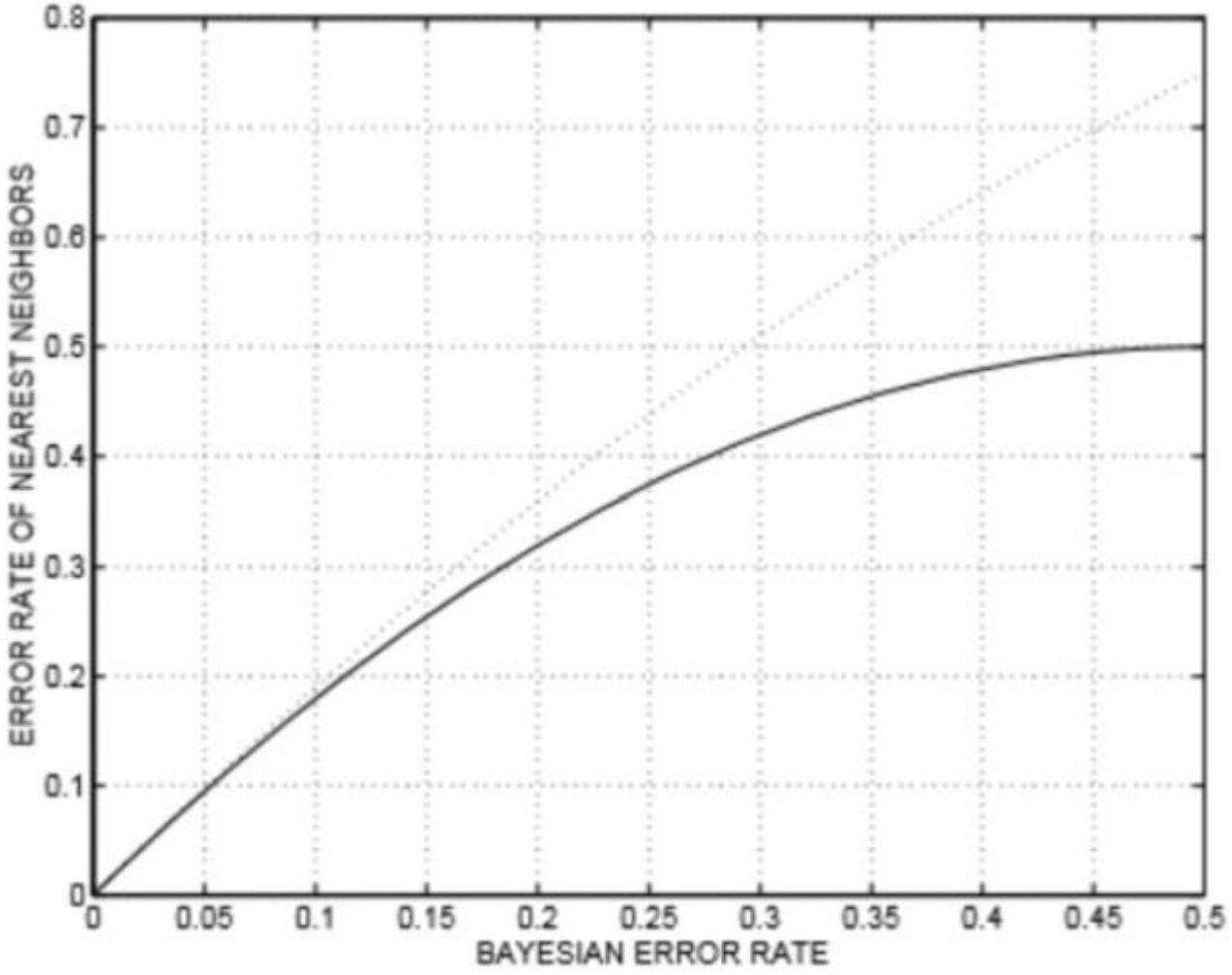
- Alakasız nitelikler neden k-NN sınıflandırıcının performansını bozar?
- Öznitelik ölçeklemeyle ilgili temel sorunlar nelerdir?
- Normalleştirme nedir? Neden gereklidir?
- Normalleşme hangi koşullarda yanıltıcı olur?

1-NN vs Bayes

- Herhangi bir sınıflandırıcının başarısını değerlendirmek için nihai ölçüt Bayes formülüdür.

- Bayes sınıflandırıcısında kullanılan olasılıklar ve pdf'ler mutlak doğrulukla biliniyorsa, o zaman bu sınıflandırıcı –buna İdeal Bayes diyelim– verilen (gürültülü) veriler üzerinde teorik olarak elde edilebilecek en düşük hata oranını sergiler.

1-NN vs Bayes


ERROR RATE OF NEAREST NEIGHBORS
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
BAYESIAN ERROR RATE

Komşuların Sayısını Artırmak

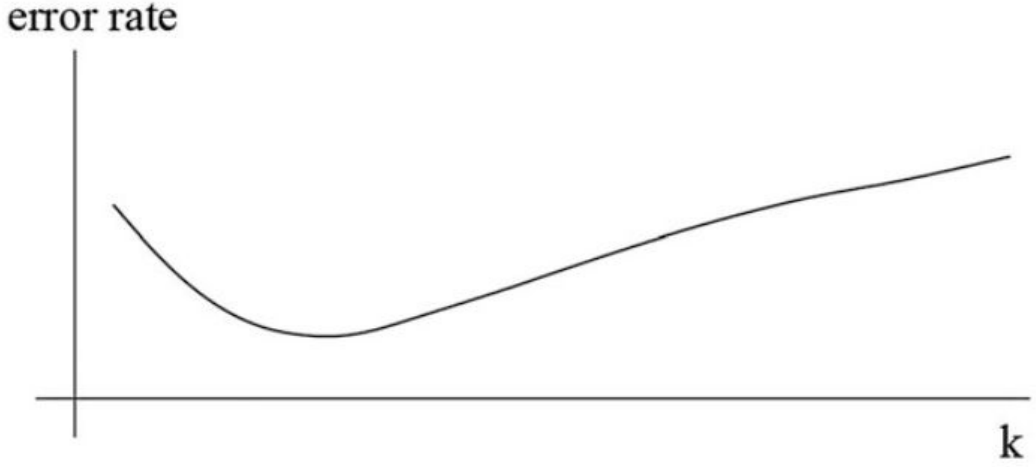
- k büyüdükçe hata oranının azaldığı ve $k \rightarrow \infty$ için İdeal Bayes'e yakınsadığı kanıtlanmıştır.

- En azından teoride, en yakın komşu sınıflandırıcının performansı maksimuma ulaşabilir.

Komşuların Sayısını Artırmak

- Gerçekçi bir uygulamada, eğitim örnekleri örnek uzayını seyrek olarak dolduracaktır ve oy veren **komşuların sayısını artırmak** ters etki yapabilir.

- Uç noktayı düşünün: eğitim seti 25 eğitim örneği içeriyorsa, 25-NN sınıflandırıcı, herhangi bir nesneyi eğitim verilerinde en yaygın sınıfa sahip olacak şekilde etiketler.



Çok Boyutluluğun Laneti

- Artık en yakın komşulardan bazılarının bir oy hak etmek için x'e yeterince benzemeyebileceğini anlıyoruz.
- Bu genellikle **çok sayıda özelliğe sahip domain'lerde** olur.
- Her özniteliğin değerlerinin [0, 1] birim uzunluk aralığıyla sınırlı olduğunu varsayalım. Pisagor teoremini kullanarak, bu nitelikler tarafından tanımlanan n-boyutlu uzayda maksimum Öklid mesafesinin **dMAX = √n** olduğunu kolayca gösterebiliriz.
- Özniteliklerin sayısını artırdıkça, örnek uzayını yeterli yoğunlukla doldurmak için gereken eğitim örneklerinin sayısı çok hızlı, belki de en yakın komşu paradigmasını kullanışsız kılacak kadar hızlı büyür.

Sonuç

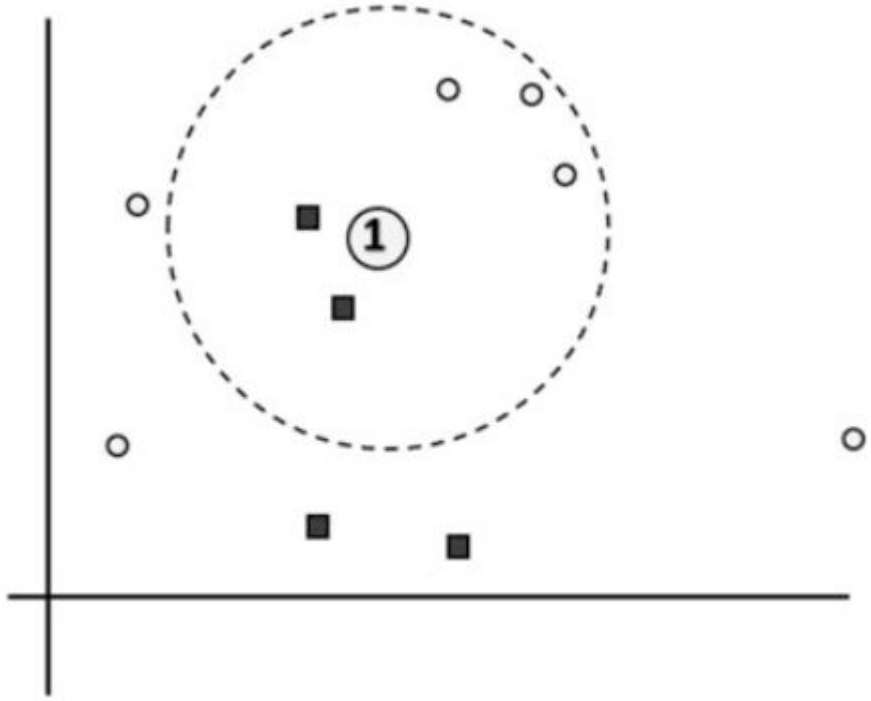
- İdeal k-NN sınıflandırıcı, İdeal Bayes'in performansına ulaşma yeteneğine sahip olsa da, mühendis her iki yaklaşımın pratik sınırlamalarının farkında olmalıdır.
- İdeal Bayes, olasılıklar ve pdf'lerin mükemmel bilgisinin yokluğunda **gerçekçi değildir**.
- Öte yandan, k-NN, **seyrek veri, alakasız nitelikler ve uygun olmayan nitelik ölçeklendirmesinden muzdariptir**.
- Somut seçim, verilen uygulamanın özel gereksinimlerine bağlıdır.

Sorular

- k-NN'nin performansı, İdeal Bayes'in performansıyla nasıl karşılaştırılır? Bunu k = 1 ve k > 1 için ayrı ayrı özetleyin. İki paradigma hangi teorik varsayımlara dayanıyor?
- k-NN sınıflandırıcısının performansı k değeri arttıkça nasıl değişir? Teori ve pratik arasındaki fark nedir?
- "Çok Boyutluluğun Laneti"nden ne anlıyorsunuz?

Ağırlıklandırılmış NN

- Şimdiye kadar, oylama mekanizması, komşular arasında en çok gözlenen sınıfın kazanması anlamında demokratiktir.
- Ancak **demokrasi azaltılırsa** sınıflandırma performansı genellikle iyileşir.
- <u>Komşu ne kadar yakınsa, etkisi o kadar büyük olur.</u>

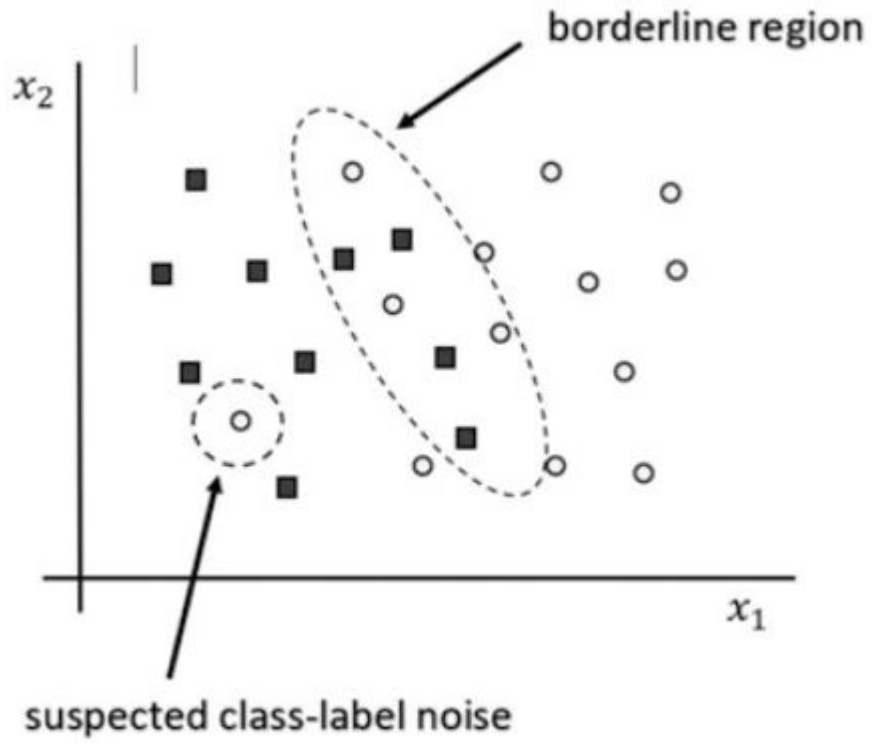


$$w_i = \begin{cases} \frac{d_k - d_i}{d_k - d_1}, & d_k \neq d_1 \\ 1 & d_k = d_1 \end{cases}$$

Tehlikeli Örnekleri Silmek

- Her eğitim örneğinin değeri farklı olabilir. Bazıları temsil ettikleri sınıfların tipik özelliği iken, bazıları da sınıfı daha az temsil edebilir.
- Bu nedenle, eğitim setini önceden işlemek, yararlı olmadığından şüphelenilen örnekleri kaldırmak genellikle iyi bir şeydir.
- İlk olarak, bir sınıfla etiketlenmiş ancak başka bir sınıfın örnekleriyle çevrelenmiş bir örnek, sınıf etiketi gürültüsünü gösterebilir. İkincisi, iki sınıfı ayıran sınır bölgesinden örnekler güvenilir değildir: öznitelik değerlerindeki küçük miktardaki gürültü bile konumlarını yanlış yönlere kaydırabilir ve böylece sınıflandırmayı etkileyebilir.
- **Ön işleme**, bu iki tür örneği eğitim setinden çıkarmayı amaçlar.

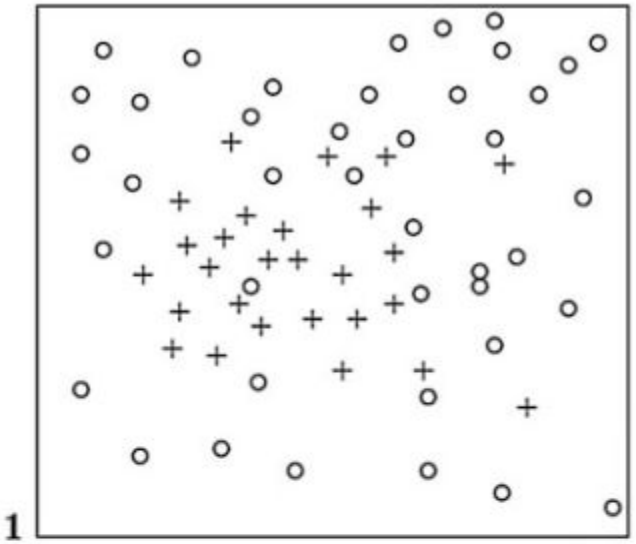
Tehlikeli Örnekleri Silmek

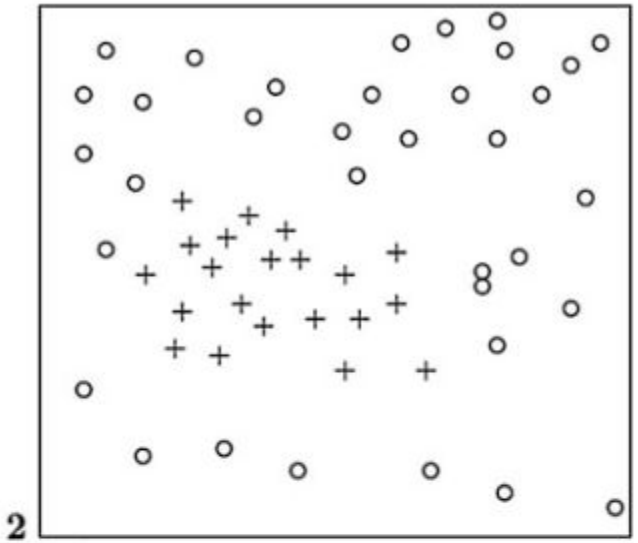


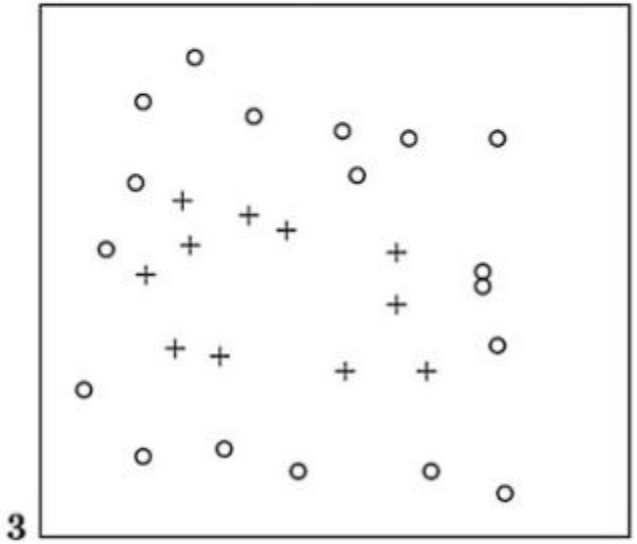
Tekrarlı Örnekleri Silmek

- Makine öğrenmesi uygulamasında, yaklaşık 10^4 öznitelik tarafından tanımlanan 10^6 eğitim örneğine sahip domain'lerle karşılaşabiliriz.
- Ayrıca, binlerce nesneyi olabildiğince çabuk sınıflandırmak gerekebilir. Tek bir nesnenin en yakın komşusunu belirlemek için, Öklid mesafesine dayanan en yakın sınıflandırıcının $10^6 \times 10^4 = 10^{10}$ aritmetik işlemi yapması gerekir.
- Bunu binlerce nesne için tekrarlamak, $10^{10} \times 10^3 = 10^{13}$ aritmetik işlemle sonuçlanır. Bu pratik olmayabilir.
- Neyse ki, k-NN sınıflandırıcısının davranışı birçok eğitim örneğinin silinmesinden etkilenmez.

Tekrarlı Örnekleri Silmek







1: original training set

2: borderline and noisy examples removed

3: redundant examples removed

Nitelik-Vektör Benzerliğinin Sınırlılıkları

- Herhangi bir çocuk size bir kangurunun karnındaki cep yardımıyla kolayca tanınabileceğini söyleyecektir. Örnekleri açıklayan tüm nitelikler arasında, "cebin" varlığı veya yokluğu hakkındaki Boole bilgisi en belirgin olanıdır ve öneminin, kalan tüm niteliklerin toplamından daha büyük olduğunu iddia etmek abartı olmaz. Zürafada, sivrisinek veya solucanda yoktur.

- Nitelikleri ilgili, alakasız ve gereksiz olarak ayırmak çok kabadır. "Kanguru" deneyimi bize, ilgili olanlar arasında bazılarının diğerlerinden daha önemli olduğunu göstermektedir.

Nitelik-Vektör Benzerliğinin Sınırlılıkları

- Kangurularda açıkça gözlemlenen bir diğer özellik ise ön bacaklarının arka ayaklarından çok daha kısa olmasıdır. Ancak bu özellik, öznitelik vektörleri arasındaki geometrik uzaklıklardan türetilen benzerlikler tarafından hemen yansıtılmaz. Tipik olarak, hayvan örnekleri, bir ön bacağın uzunluğu ve bir arka bacağın uzunluğu (diğerleri arasında) gibi niteliklerle tarif edilecektir, ancak farklı uzunluklar arasındaki ilişki yalnızca örtüktür.

- Sınıflandırma orijinal nitelikler yerine a1/a2 gibi bireysel nitelikler arasındaki ilişkilere daha çok bağımlı olabilir. İki veya daha fazla özelliğin karmaşık bir işlevi, bireysel niteliklerden daha bilgilendirici olacaktır.

Bilgisayar Ödevi

1. Girdisi eğitim seti, kullanıcı tarafından belirlenen bir k değeri ve bir nesne, x olan bir program yazın. Çıktı, x'in sınıf etiketidir.

1. Üstte tanımlanan programı UCI deposundaki (https://archive.ics.uci.edu/ml/index.php) bazı karşılaştırmalı alanlara uygulayın. Örneklerin her zaman %40'ını alın ve kalan %60'ta çalışan 1-NN sınıflandırıcı ile yeniden sınıflandırın.

Bilgisayar Ödevi

3. Her biri [0,1] aralığındaki bir çift öznitelikle açıklanan 1000 örnekten oluşan yapay bir alan oluşturun. [0, 1] × [0, 1] bu öznitelik değerlerinin tanımladığı karede, kendi seçtiğiniz bir geometrik şekli tanımlayın ve içindeki tüm örnekleri pozitif, dışındaki tüm örnekleri negatif olarak etiketleyin. Bu ilk gürültüsüz veri setinden, her biri sınıf etiketlerinin yüzde p'sini değiştirerek elde edilen 5 dosya oluşturun, $p \in \{5, 10, 15, 20, 25\}$ (böylece farklı sınıf etiketi gürültüsü seviyeleri elde edilir). Bu veri dosyalarının her birini iki bölüme ayırın, ikincisi k-NN sınıflandırıcısı tarafından yeniden sınıflandırılacak şekilde birinci bölümden gelen verileri çalıştırın. Farklı sınıf etiketi gürültüsü seviyeleri altında farklı k değerlerinin nasıl farklı davranışlara yol açtığını gözlemleyin.

- Ders Sonu.