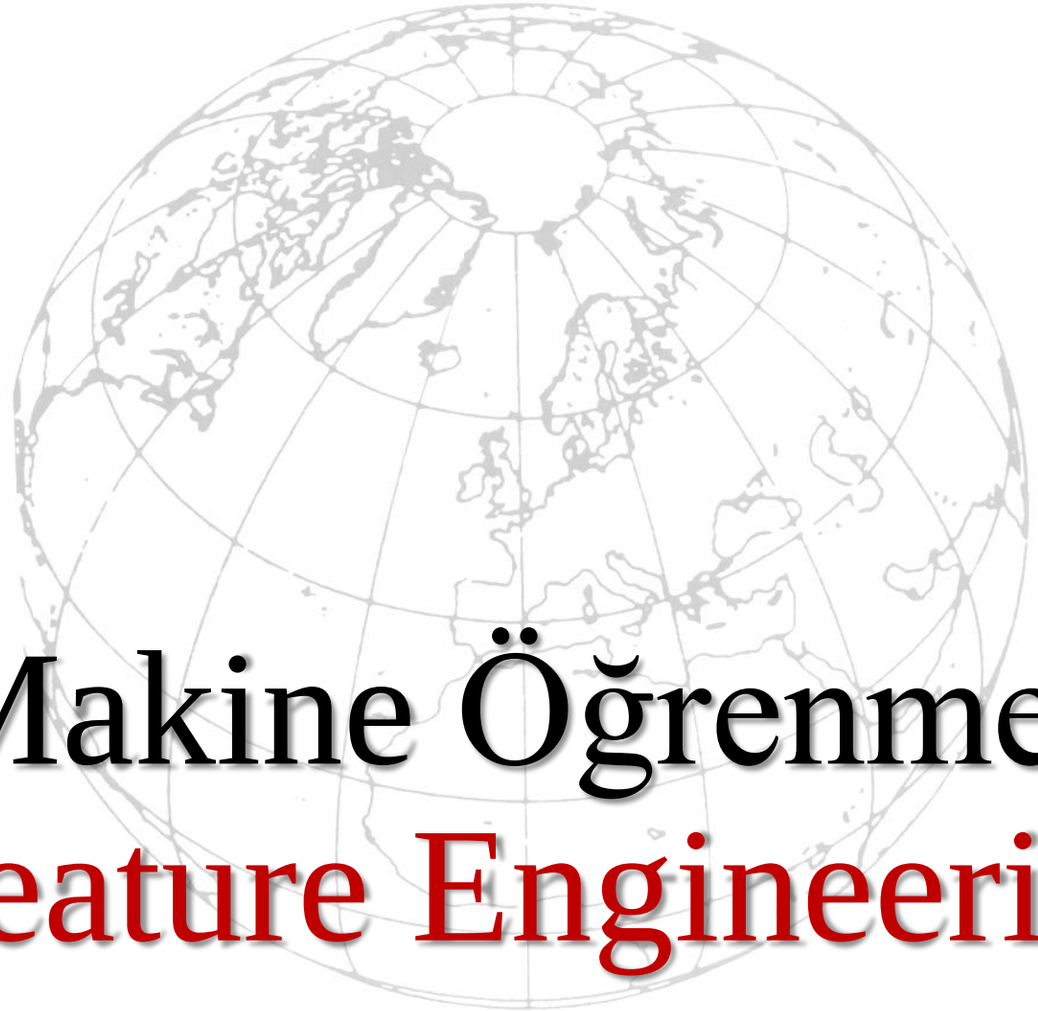


Makine Öğrenmesi
Feature Engineering

Dr. Cahit Karakuş

Makine Öğreniminde Özellik Vektörleri

Öğrenme Algoritmaları

- Makine öğreniminin başarısı algoritmalara ve matematiksel modellere bağlıdır.
- Algoritmalar, bilgi yapılarını bulmak ve inşa etmek için aramayı kontrol eder.
- Öğrenme algoritmaları, eğitim örneklerinden faydalı bilgiler çıkarmalıdır.

Girdi Vektörü

- Girdi vektörü çeşitli adlarla adlandırılır. Bunlardan bazıları şunlardır: girdi vektörü, model vektörü, özellik vektörü.
- Giriş vektörünün bileşenleri, xi, çeşitli şekillerde özellikler, nitelikler, giriş değişkenleri ve bileşenler olarak adlandırılır.
- Bileşenlerin değerleri üç ana tipte olabilir: gerçek değerli sayılar, ayrık değerli sayılar veya kategorik değerler olabilir.
- Kategorik değerleri gösteren bir örnek olarak, bir öğrenci hakkındaki bilgiler, sınıf, ana dal, cinsiyet, danışman niteliklerinin değerleri ile temsil edilebilir.
- Belirli bir öğrenci daha sonra aşağıdaki gibi bir vektörle temsil edilir: (ikinci sınıf, tarih, erkek, higgins).
- Ek olarak, kategorik değerler sıralı (küçük, orta, büyük gibi) veya sırasız (az önce verilen örnekte olduğu gibi) olabilir. Tabii ki, tüm bu tür değerlerin karışımları mümkündür.
- Her durumda, özniteliklerin adlarını değerleriyle birlikte listeleyerek girdiyi sırasız biçimde göstermek mümkündür.
- Vektör formu, niteliklerin bir form tarafından örtük olarak sıralandığını ve verildiğini varsayar.
- Nitelik-değer temsiline bir örnek olarak şunları alabiliriz: (majör: tarih, cinsiyet: erkek, sınıf: ikinci sınıf, danışman: higgins, yaş: 19).
- Yalnızca vektör formunu kullanacağız. Önemli bir uzmanlık, ayrık sayıların (1,0) veya kategorik değişkenlerin (Doğru, Yanlış) özel bir durumu olarak kabul edilebilecek Boole değerlerini kullanır.

Çıktı

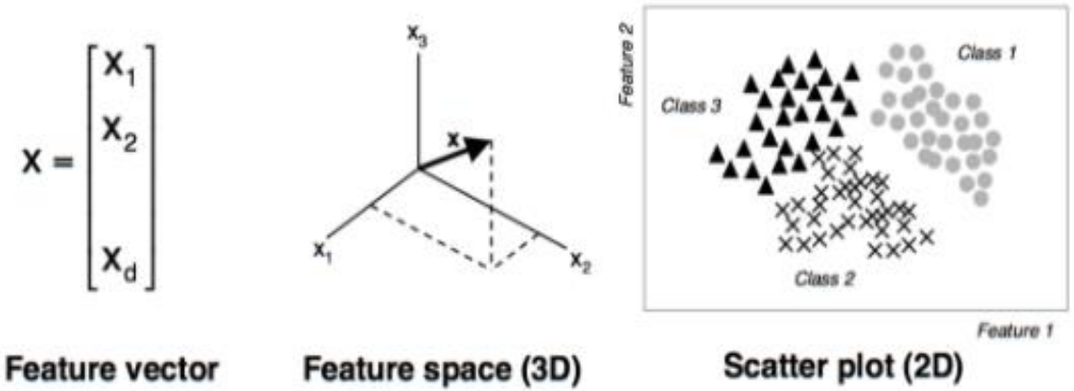
- Çıktı gerçek bir sayı olabilir, bu durumda h fonksiyonunu içeren sürece fonksiyon tahmincisi denir ve çıktıya çıktı değeri veya tahmini denir.
- Alternatif olarak, çıktı kategorik bir değer olabilir, bu durumda h'yi içeren sürece çeşitli şekillerde sınıflandırıcı, tanıyıcı veya kategorizatör adı verilir ve çıktının kendisine etiket, sınıf, kategori veya karar adı verilir.
- Sınıflandırıcıların, örneğin elle basılmış karakterlerin tanınması gibi bir dizi tanıma probleminde uygulamaları vardır. Bu durumda girdi, yazdırılan karakterin uygun bir temsilidir ve sınıflandırıcı bu girdiyi, diyelim ki 64 kategoriden birine eşler.
- Bileşenlerin gerçek sayılar veya kategorik değerler olmasıyla vektör değerli çıktılar da mümkündür.
- Önemli bir özel durum, Boolean çıkış değerleridir. Bu durumda, 1 değerine sahip bir eğitim modeline pozitif bir örnek, 0 değerine sahip bir eğitim örneğine ise bir negatif örnek denir. Girdi aynı zamanda Boolean olduğunda, sınıflandırıcı bir Boolean işlevi uygular. Basitleştirilmiş bir ortamda önemli genel noktalara değinmemize izin verdiği için Boole durumunu biraz ayrıntılı olarak inceliyoruz. Bir Boole fonksiyonunu öğrenmeye bazen kavram öğrenme denir ve fonksiyona kavram denir.

Makine öğreniminde, özellik vektörleri

- Makine öğreniminde, özellik vektörleri, bir nesnenin özellikleri olarak adlandırılan sayısal veya sembolik özelliklerini matematiksel, kolay analiz edilebilir bir şekilde temsil etmek için kullanılır.
- Makine öğrenimi ve örüntü işlemenin birçok farklı alanı için önemlidirler. Makine öğrenimi algoritmaları, algoritmaların işleme ve istatistiksel analiz yapabilmesi için tipik olarak nesnelerin sayısal bir temsilini gerektirir.
- Özellik vektörleri, doğrusal regresyon gibi istatistiksel prosedürlerde kullanılan açıklayıcı değişkenlerin vektörlerinin eşdeğeridir.
- Aşina olabileceğiniz bir özellik vektörü örneği, RGB (kırmızı-yeşil-mavi) renk açıklamalarıdır. Bir renk, içinde ne kadar kırmızı, mavi ve yeşil olduğu ile tanımlanabilir. Bunun için bir özellik vektörü color = [R, G, B] olacaktır.

Giriş

- Bir vektör, bir sütunlu ancak birden çok satırlı bir matris gibi, genellikle uzamsal olarak temsil edilebilen bir dizi sayıdır.
- Bir özellik, bir nesnenin bir yönünün sayısal veya sembolik bir özelliğidir.
- Özellik vektörü, bir nesne hakkında birden çok öğe içeren bir vektördür.
- Nesneler için özellik vektörlerini bir araya getirmek bir özellik uzayı oluşturabilir.
- Özellikler, bir bütün olarak, yalnızca bir pikseli veya tüm bir görüntüyü temsil edebilir.
- Ayrıntı düzeyi, birinin nesne hakkında ne öğrenmeye veya temsil etmeye çalıştığına bağlıdır. Yüksekliğini, genişliğini, derinliğini vb. gösteren bir özellik vektörüyle 3 boyutlu bir şekli tanımlayabilirsiniz.



Feature vector Feature space (3D) Scatter plot (2D)

Özellik Vektörlerinin Kullanımları

- Özellik vektörleri, birçok türde analize yardımcı olmak için nesneleri sayısal bir şekilde temsil etmenin etkinliği ve pratikliği nedeniyle makine öğreniminde yaygın olarak kullanılmaktadır.
- Analiz için iyidirler çünkü özellik vektörlerini karşılaştırmak için birçok teknik vardır. İki nesnenin öznitelik vektörlerini karşılaştırmanın basit bir yolu, Öklid mesafesini almaktır.
- Görüntü işlemede özellikler, gradyan büyüklüğü, renk, gri tonlama yoğunluğu, kenarlar, alanlar ve daha fazlası olabilir.
- Özellik vektörleri, listelenen örnekler gibi bir görüntüyle ilgili özniteliklerin, öznitelik vektörlerine yerleştirildikten sonra sayısal olarak karşılaştırılabilmesi nedeniyle, görüntü işlemedeki analizler için özellikle popülerdir.
- Konuşma tanımada özellikler ses uzunlukları, gürültü seviyesi, gürültü oranları ve daha fazlası olabilir.
- İstenmeyen postayla mücadele girişimlerinde, özellikler bol miktarda bulunur. IP konumu, metin yapısı, belirli kelimelerin sıklığı veya belirli e-posta başlıkları olabilir.
- Özellik vektörleri, sınıflandırma problemlerinde, yapay sinir ağlarında ve makine öğreniminde k-en yakın komşu algoritmalarında kullanılır.

Tanımlar

- **Özellik:** sayıların bir listesidir, örneğin: yaş, ad, boy, kilo vb., bu, her sütunun ilişkisel tablodaki bir özellik olduğu anlamına gelir.
- **Özellik Vektörü**, belirli bir satırın ilişkisel tablodaki temsilidir. Her satır bir özellik vektörüdür, 'n' satırı 'n'inci örnek için bir özellik vektörüdür.
- **Özellik Seti:** Çıktı değişkenini tahmin etmeye yardımcı olur.

ID	First Name	Last Name	Email	Year of Birth
1	Peter	Lee	plee@university.edu	1992
2	Jonathan	Edwards	jedwards@university.edu	1994
3	Marilyn	Johnson	mjohnson@university.edu	1993
6	Joe	Kim	jkim@university.edu	1992
12	Haley	Martinez	hmartinez@university.edu	1993
14	John	Mfume	jmfume@university.edu	1991
15	David	Letty	dletty@university.edu	1995

Feature Vector

Table: Students

Tanımlar

- **Çıkarma:** "Ham" verilerden özellikleri çıkarma
- **Dönüşüm:** Ölçekleme, dönüştürme veya özellikleri değiştirme
- **Seçim:** Daha büyük bir özellik kümesinden bir alt küme seçme
- **Yerelliğe Duyarlı Karma (LSH):** Bu algoritma sınıfı, özellik dönüşümünün özelliklerini diğer algoritmalarla birleştirir.

- Özellik vektörü, makine öğreniminde örüntü tanımada bazı nesneleri tanımlayan sayısal özelliklerin n boyutlu bir vektörüdür.
- Birçok makine öğrenimi algoritması, işlemeyi ve istatistiksel analizi kolaylaştırdıkları için nesnelerin sayısal temsillerine dayanır.
- Bir vektör, sayısal değerler listesinden başka bir şey değildir. Bir vektörün yalnızca bir özelliğin hesaplanmış değerlerinin bir listesi olduğu açıktır.
- Bir veri kümesi genellikle her biri kendi özellik kümesine sahip birkaç örneğe bölünür. Her örnek, o örnek nesne için tüm sayısal değerleri içeren tek bir özellik vektörüne karşılık gelir.
- Tüm özellik vektörleri normalde bir tasarım matrisinde istiflenir, her satır bir örnek için bir vektörü temsil eder ve her sütun o özellik için tüm örneklerin değerlerini temsil eder.
- Verilerinizi, özelliklerinizi temsil eden sütunlar ve çeşitli örneklerinizi temsil eden satırlarla birlikte bir elektronik tabloda düzenlediğinizi varsayalım. Şunu düşünün: Üç kişiye cinsiyetlerini ve yaşlarını sorarsanız, elinizde üç satır (3 kişi) ve iki sütun (boy, kilo) olan bir elektronik tablo elde edersiniz.Her satır artık tek bir fonksiyon vektörü olarak yorumlanabilir.
- Örneğimizdeki fonksiyon vektörünün iki boyutu olacaktır (yükseklik, ağırlık). Boyutlar farklı alanlardan geldiğinden, fiziğin yokluğunda fonksiyon vektörünün büyüklüğünün bizim için doğrudan bir uygulaması olmayabilir (aksine, bir hız vektörünü karşılaştırın). Bununla birlikte, büyüklüğü (normalizasyondan sonra) hesaplayabildik.
- Öznitelik vektörünün yönü ise öznitelik değerlerini yansıttığı için önemlidir.

Örnek

- If $\vec{a}$ and $\vec{b}$ are two vectors, what is the value of $(\vec{a}+\vec{b})\times(\vec{a}-\vec{b})$?
- If a and b are two vectors, what is the value of (a+b) x (a-b)?
- (a +b) x (a -b) = a x a - a x b + b x a - b x b

- but a x a = b x b =0

- Also b x a = - a x b

- therefor (a + b) x (a -b) = -2 a x b

Özellik – Özellik vektörü

- Bir özellik, bir nesnenin tek bir yönüdür. Bir kişinin bir özelliği, örneğin kilolarıdır. Bir diğeri ise boylarıdır. Bir diğeri ise bel ölçüleridir.
- Bir dizi özellik seçip bunları bir demet halinde birleştirdiğimizde, burada bir özellik vektörü elde ederiz (m-yükseklik, kilo-ağırlık, m-bel).

Lineer cebirde vektör uzayı nedir?

- Basitçe söylemek gerekirse, bir vektör uzayı, vektörler olarak adlandırılan (fizikte kullandığımız tipik vektörler değil) bir araya getirilebilen ve skalerlerle çarpılabilen bir matematiksel nesneler topluluğudur.
- Vektör uzayı terimi, sayılarla çarpılabilen ve bir araya toplanabilen her tür matematiksel nesneyi ifade etmek için icat edildi.
- Buradaki toplama vektör toplamadır (İki vektör arasındaki toplama) ve buradaki çarpma skaler çarpmadır (gruptaki bir vektörü bir skaler ile çarpma).
- Vektör uzayı olarak adlandırılabilmesi için vektör toplama ve skaler çarpmanın belirli kuralları sağlaması gerekir. Vektörler vektör uzayından gelir ve skalerler alandan gelir.

Vektör toplama kuralları

- Vektör uzayının bir parçası olan iki u ve v varlığı için, u + v de vektör uzayının bir parçası olmalıdır. Bu nedenle, işlem ikili bir işlemdir.
- Bir kümedeki ikili işlemler nelerdir?
- İki varlık u ve v için, u + v = v + u (Değişmeli özellik)
- İki varlık u ve v için, (u + v) + w = u + (v + w). (İlişkisel özellik)
- u + 0 = 0 + u = u kimliğinin varlığı.
- Buradaki 0 normal sıfır değildir. Sadece ekleme ile ilgili olarak kimlik öğesinin bir temsilidir. Toplama ile ilgili kimlik elemanı, vektör uzayında 0 değerini etkilemeyecek herhangi bir giriştir. Mevcut tersi u + v = 0. Muhtemelen toplama için, v = -u.

Skaler çarpma kuralları

- v, c'ye ait olan u için. v ayrıca v'nin bir parçasıdır
- c(u + v) = c u + c v . Vektörlerin toplamı ile çarpılan bir skaler için dağılım yasası. (c + d) u = c u + d u . Y
- ine, bir vektör için dağılım yasası, bir skaler toplamı ile çarpılır.c (d.v) = (c.d) . v (İlişkisel özellik)
- Kimlik öğesinin varlığı u . 1 = 1 . u = u
- Burada c ve d'nin tümü skalerdir ve alanın bir parçası olmalıdır. Alanın elemanlarına skaler, vektör uzayının elemanlarına vektör denir. Lütfen buradaki vektörlerin fizikte okunan normal vektörler olmadığını unutmayın. Bunlar sadece vektör uzayının bileşenleridir.

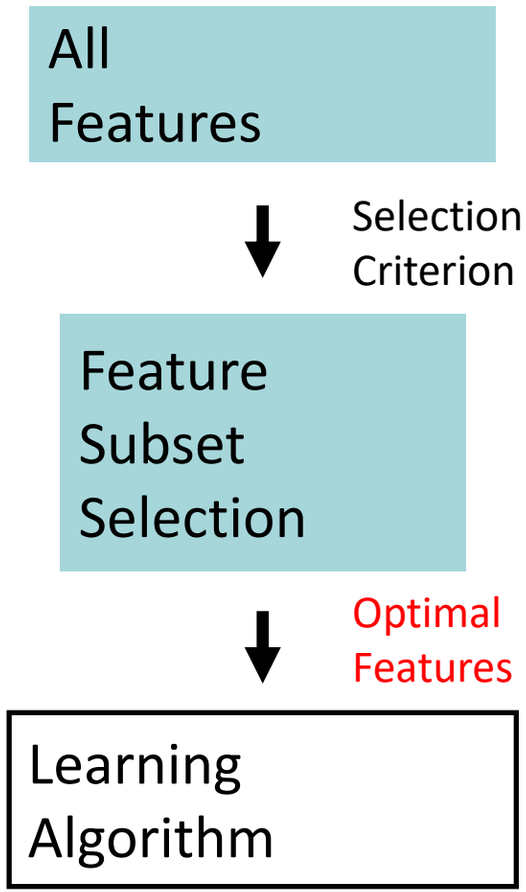
Makine Öğrenmesi
Özellik Seçimi

Özellik Seçiminin Faydaları

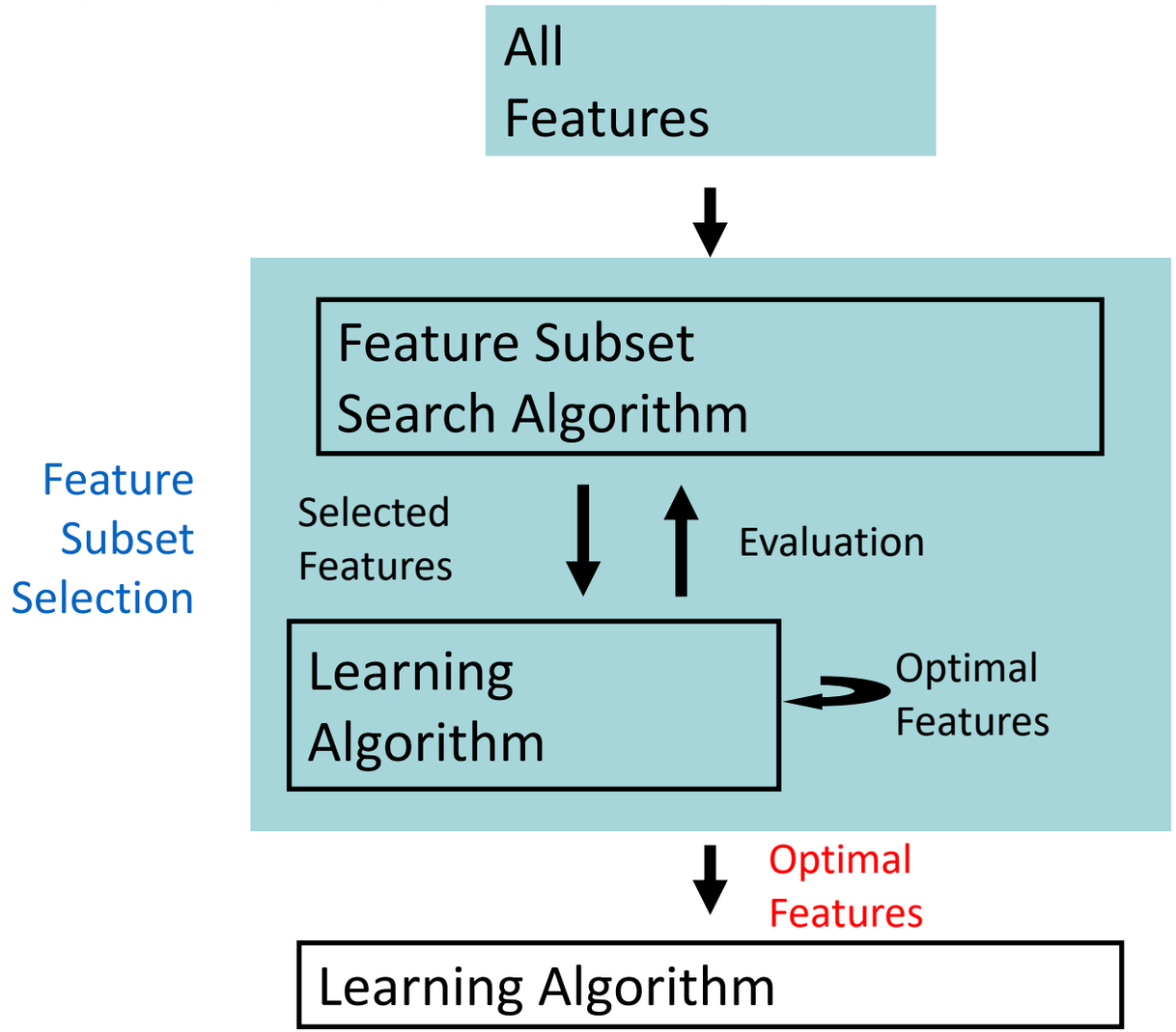
- Küçük bir özellik alt kümesini kullanılarak iyi ve genellikle daha iyi sınıflandırma performansı elde edilebilir.
  - Verilerde daha az gürültü
- Daha uygun maliyetli sınıflandırıcılar sağlanır
  - Dikkate alınması gereken daha az özellik
    - daha küçük veri kümeleri
    - daha hızlı sınıflandırıcılar
- Verilen problem için (biyolojik olarak) ilgili özelliklerin detaylı tanımlanması

Özellik Seçimi

Filter approach



Wrapper (kapsayıcı özellik) approach



Filter Approach

- Sınıflandırma modelinden bağımsız
- Her özellik için bir alaka ölçüsü hesaplanır
- Seçilen eşikler aralığı dışındaki değere sahip özellikler kaldırılır.

Example: **Feature-class entropy**

- Özelliği gözlemlerken sınıf hakkındaki “belirsizliği” ölçer.

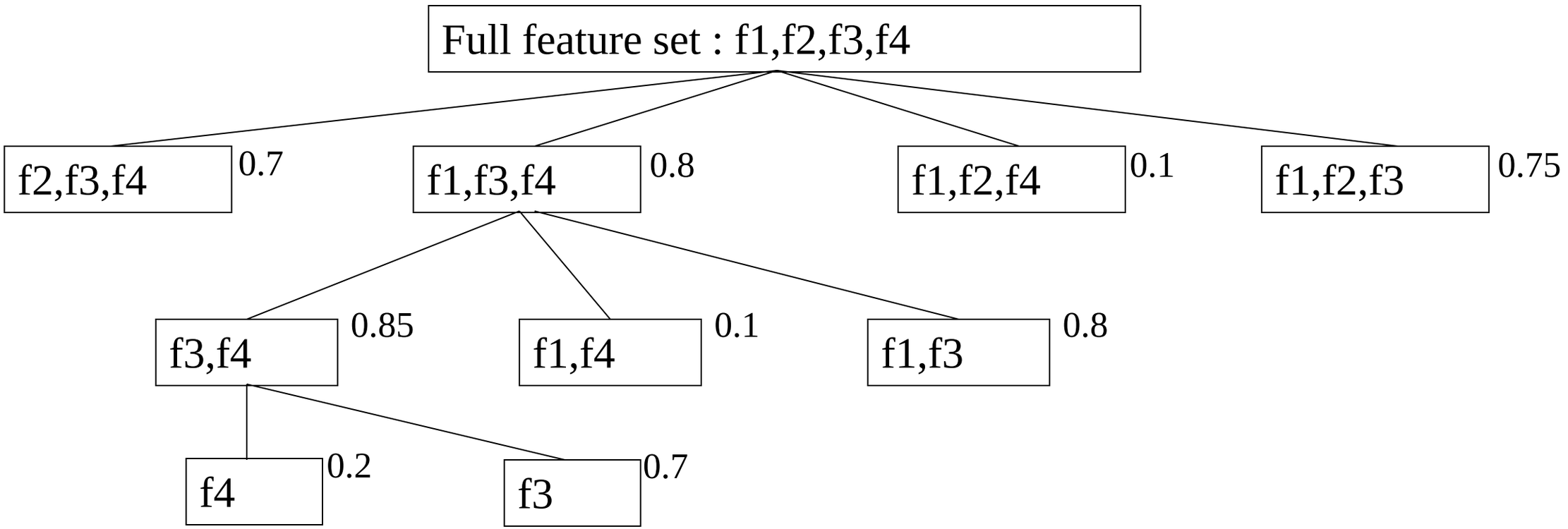
f1	f2	f3	f4	class	f1	f2	f3	f4	class
1	0	1	1	1	1	0	0	0	0
0	1	1	0	1	0	0	1	0	0
1	0	1	0	1	1	1	0	1	0
0	1	0	1	1	0	1	0	1	0

Wrapper approach

- Bir sınıflandırma algoritmasına özgü iyi bir özellik alt kümesi aranması, bir arama algoritması tarafından yönlendirilir.
- Algoritma, iyi özellik alt kümelerini bulmak için sınıflandırıcının değerlendirmesini bir kılavuz olarak kullanır.
- Arama algoritması örnekleri: sıralı ileri veya geri arama, genetik algoritmalar

- Sıralı geriye doğru eleme
  - Tüm özellikler kümesiyle başlar
  - en iyi sınıflandırma performansıyla sonuçlanan özelliğinin kaldırılmasını yinelemeli olarak atar.

Wrapper approach



Makine Öğrenmesinde Öznitelikler

Öznitelik

- Makine öğrenmesi ve örüntü tanıma alanlarında, gözlemlenen bir olgunun ölçülebilir bir niteliğine özellik ya da öznitelik denir.
- Anlaşılır, ayırt edici ve bağımsız özellikler seçmek etkili örüntü tanıma, sınıflandırma ve regresyon algoritmaları için kritik bir adımdır. Özellikler genellikle sayısaldır ancak sentaktik örüntü analizinde kelimeler ve çizgeler de kullanılır.

- İşlenmemiş öznitelikler kümesi gereksiz öğeler içerebilir ve büyüklüğünden ötürü yönetilmesi zor olabilir. Bu yüzden, makine öğrenmesi ve örüntü tanıma uygulamalarından çoğu özniteliklerin bir alt kümesinin seçilmesini ya da yeni ve indirgenmiş bir öznitelikler kümesinin oluşturulmasını içerir. Kullanılacak özniteliklerin öğrenmeyi kolaylaştırması, genelliği ve yorumlanabilirliği artırması amaçlanır.

- Özniteliklerin çıkarılması ya da seçilmesi öznitelik mühendisliği olarak adlandırılır. Birçok farklı ihtimalin deneylenmesi ve hazır yöntemler ile bir alan uzmanının önsezilerinin bir araya getirilmesini gerektirir.
- Bir sayısal öznitelikler kümesinin tanımlanması için öznitelik vektörü kullanılabilir. Bir öznitelik vektörü kullanılarak iki ihtimalli sınıflandırma yapılması öznitelik vektörü ve bir ağırlıklar vektörünün skaler çarpımının alınması ve çarpım sonucunun bir eşik değeri ile karşılaştırılması ile mümkün olur.
- Bir öznitelikler vektörü kullanılarak yapılan sınıflandırma algoritmalarından bazıları en yakın komşu sınıflandırması, yapay sinir ağları ve Bayes yaklaşımlarıdır.

Öznitelik Seçim Algoritmaları

- Makine öğrenmesi algoritmaları sınıflandırma yaparken en önemli noktalardan biri hangi değişkenlerin etkili olduğunu öğrenmektir.
- Sınıflandırma algoritmalarında kullanılan değişken sayısının olabildiğince azaltılması ve sınıflandırma sonuçlarında bunu yaparken önemli bir düşüş olmaması son derece önemlidir.
- Veri setindeki bazı özniteliklerde önemli bilgi bulunmaz.
- Öznitelik seçimi, veri setindeki özniteliklerden yeni bir öznitelik alt kümesi oluşturarak düzenlenir.
- Öznitelik seçimi yapılmasındaki önemli noktalardan biri de, örnek sayısının düşük, nitelik sayısının ise fazla olduğu durumlarda makine öğrenmesi algoritmalarının işlem yapmasının zor olmasıdır.
- Öznitelik seçimi, filtre modelli, sarmal modelli ve gömülü modelli algoritmalardır.
- Filtre modelli algoritmanın olumlu yönü sınıflandırma algoritmasından bağımsız ve hızlı olarak işlem yapmasıdır. Olumsuz yönü ise diğer yöntemlere göre başarı oranının daha düşük seviyede olmasıdır.
- Sarmal modelli algoritmanın olumlu yönü başarı oranını yüksek tutması ve uygulamasının kolay olmasıdır. İşlem yükünün fazlalığından dolayı yavaş çalışması ise olumsuz yönüdür.
- Gömülü modelli algoritmanın olumlu yönü, başarı oranını yüksek tutarak işlem yükünü düşürmesidir. Olumsuz yönü ise karar ağaçlarından bağımsız olarak kullanılamamalarıdır.

Öznitelik (makine öğrenmesi)

- Makine öğrenmesi ve örüntü tanıma alanlarında, gözlemlenen bir olgunun ölçülebilir bir niteliğine özellik (ya da öznitelik) denir. Veri yığınından anlaşılır, ayırt edici ve bağımsız özellikler seçmek etkili örüntü tanıma, sınıflandırma ve regresyon algoritmaları için kritik bir adımdır. Özellikler genellikle sayısaldır ancak sentaktik örüntü analizinde kelimeler ve çizgeler de kullanılır.

- İşlenmemiş öznitelikler kümesi gereksiz öğeler içerebilir ve büyüklüğünden ötürü yönetilmesi zor olabilir. Bu yüzden, makine öğrenmesi ve örüntü tanıma uygulamalarından çoğu özniteliklerin bir alt kümesinin seçilmesini ya da yeni ve indirgenmiş bir öznitelikler kümesinin oluşturulmasını içerir. Kullanılacak özniteliklerin öğrenmeyi kolaylaştırması, genelliği ve yorumlanabilirliği artırması amaçlanır.

- Özniteliklerin çıkarılması ya da seçilmesi öznitelik mühendisliği olarak adlandırılır. Birçok farklı ihtimalin deneylenmesi ve hazır yöntemler ile bir alan uzmanının önsezilerinin bir araya getirilmesini gerektirir.

Öznitelik (makine öğrenmesi)

- Bir sayısal öznitelikler kümesinin tanımlanması için öznitelik vektörü kullanılabilir. Bir öznitelik vektörü kullanılarak iki ihtimalli sınıflandırma yapılması öznitelik vektörü ve bir ağırlıklar vektörünün skaler çarpımının alınması ve çarpım sonucunun bir eşik değeri ile karşılaştırılması ile mümkün olur.
- Bir öznitelikler vektörü kullanılarak yapılan sınıflandırma algoritmalarından bazıları en yakın komşu sınıflandırması, yapay sinir ağları ve Bayes yaklaşımlarıdır.
- Makine öğreniminde ve örüntü tanımada bir özellik, gözlemlenen bir olgunun bireysel ölçülebilir bir özelliği veya karakteristiğidir. Bilgilendirici, ayırt edici ve bağımsız özellikler seçmek, örüntü tanıma, sınıflandırma ve regresyonda etkili algoritmalar için çok önemli bir adımdır. Özellikler genellikle sayısaldır, ancak dizeler ve grafikler gibi yapısal özellikler sözdizimsel örüntü tanımada kullanılır. "Özellik" kavramı, doğrusal regresyon gibi istatistiksel tekniklerde kullanılan açıklayıcı değişkenle ilgilidir.

Öznitelik (makine öğrenmesi)

- Genelde gerçek dünya problemlerinde veriler ön işlem, öznitelik çıkarma, öznitelik şeçme veya azaltma ve sınıflandırma işlemlerinden sırasıyla geçer.
- Öznitelik çıkarma işlemi genelde veriye ait olan alanla ilgili uzman kişiler tarafından çıkartılır.
- Böyle bir uzman yok ise klasik makina öğrenmesinde PCA, LDA, CCA ve otomatik kodlayıcılar gibi yöntemler uygulanır. Bu da Feature Extraction kısmının uzman kişiler tarafından yapılması manasına geliyor.
- Derin öğrenme yöntemleri denince akla ilk gelen CNN de ise öz nitelik şeçme ve öznitelik çıkarma işlemi için konvolüsyon katmaları kullanılır.
- Makina öğrenmesinde öz nitelik çıkarımı için matematiksel ve istatistiki bir sürü yöntem kullanırken derin öğrenme konvolüsyon ilginç ve başarılı bir şekilde öznitelik şeçme ve çıkarma işlemini kendisi yapmış olur.
- Elimizde resimlerden insan yüzlerini tespit eden bir problem olsun. Eğer biz bu problemi makine öğrenimi ile çözecek olsaydık insan yüzünün tüm belirgin özelliklerini sisteme tanımlamamız gerekecekti. Derin öğrenmede ise yüzleri belirleyen öznitelikleri algoritma kendisi bulacaktır. Derin öğrenmenin güçü işte tamda burdan geliyor.

Öznitelik (makine öğrenmesi)

- Girdi verileri kullanılarak yeni türetilmiş değerler çıkarılma işlemidir.
- Bir sınıflandırıcının başarısı özniteliklerin doğru çıkartımı ve seçilimi ile doğru orantılıdır.
- Modele verilecek öznitelik sayısı yani boyut fazlalığı oluşmuş ise bu durum modelin işini zorlaştırır.
- İşe yaramayan öznitelikler ayıklanmalıdır.
- Ne kadar çok öznitelik var ise o kadar çok eğitim süresi ve işlem yükü oluşturur. Bu nedenle öznitelik seçme ve azaltma yöntemleri kullanılmalıdır.

Öznitelikler

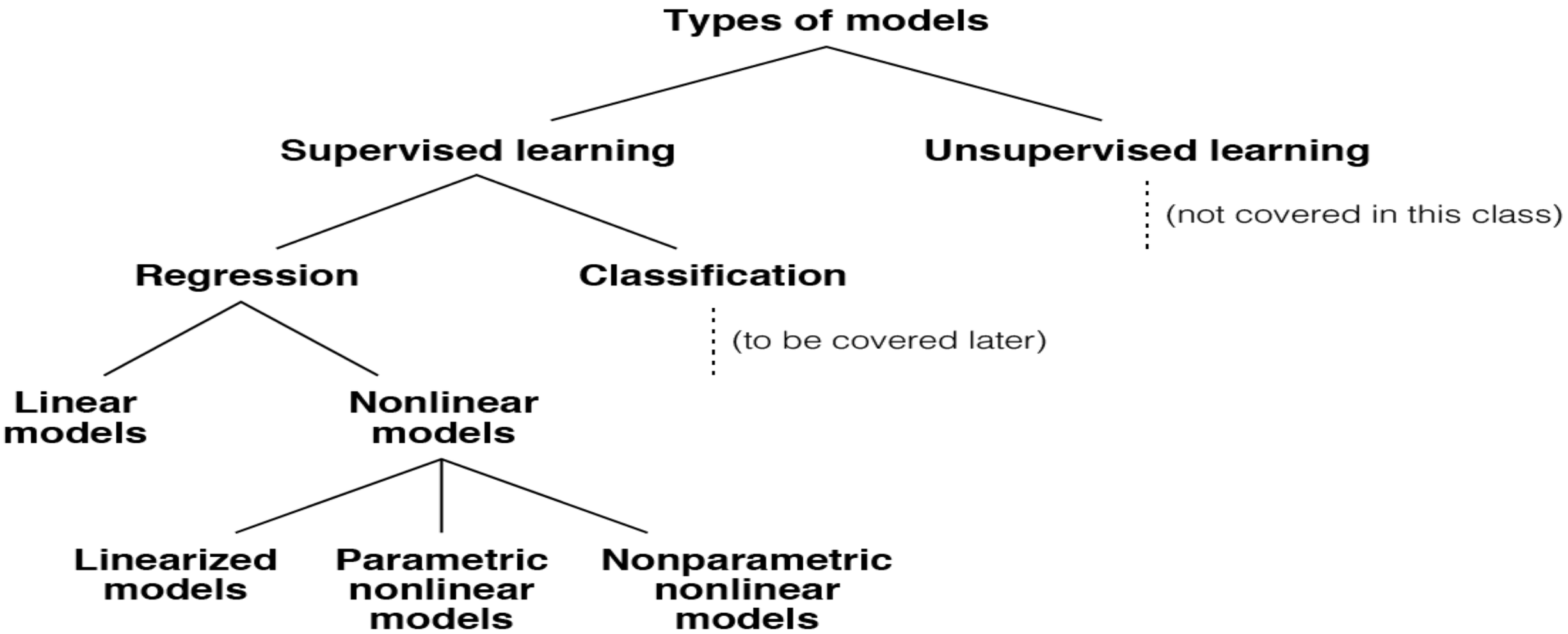
- Ortalama ve Toplam
- Varyans ve Standart Sapma
- Etkinlik
- Basıklık (Kurtosis)
- Çarpıklık (Skewness)
- Polinom Uydurma
- Willison Genliği
- Sıfır Geçiş Sayısı
- Min. ve Max. Noktalar
- Markov Katsayıları
- Entropi
- Varyans
- Frekans Bandları
- Genlik Değerleri
- Filtreleme Yöntemleri

Öznitelik Seçim Algoritmaları

- Cfs Subset Öznitelik Seçim Algoritması
- Correlation Attribute Öznitelik Seçim Algoritması
- One-R Öznitelik Seçim Algoritması
- Gain Ratio Öznitelik Seçim Algoritması
- Info Gain Öznitelik Seçim Algoritması
- ReliefF Öznitelik Seçim Algoritması

Model Specification

Model specification



Issues in model building

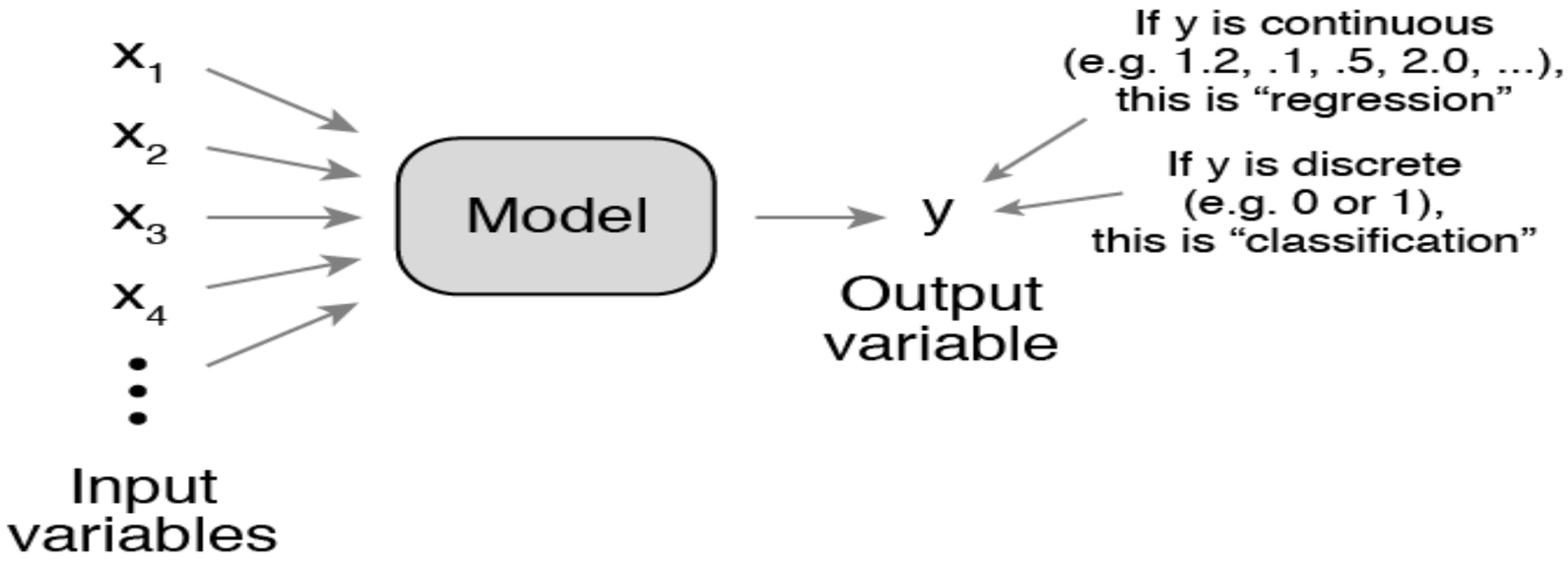
Model specification
(what type of model to use?)

Model fitting
(how do we estimate the model parameters?)

Model accuracy
(how well does the model describe the data?)

Model reliability
(how stable are the parameter estimates?)

Supervised learning

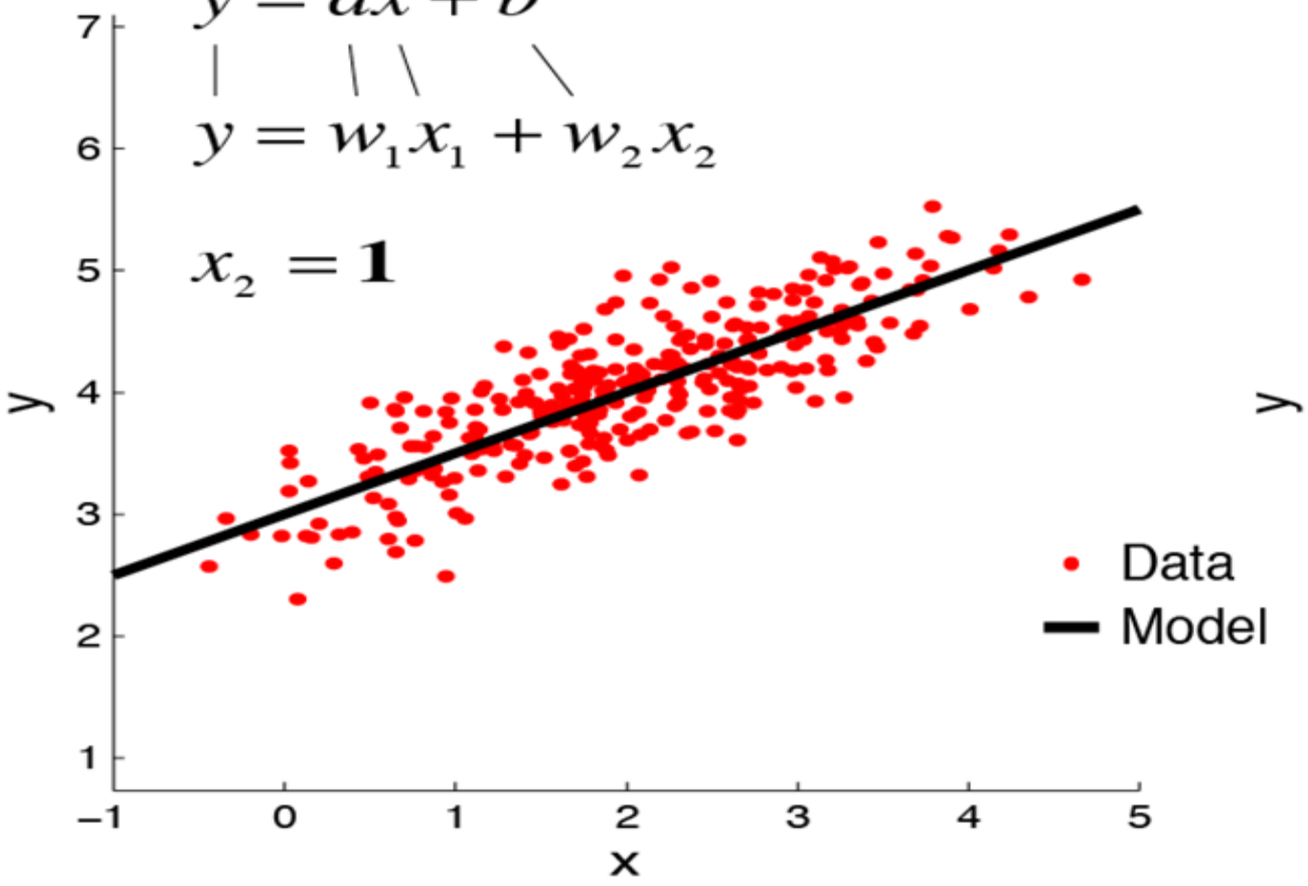


Linear model

$$y = ax + b$$

$$y = w_1 x_1 + w_2 x_2$$

$$x_2 = 1$$

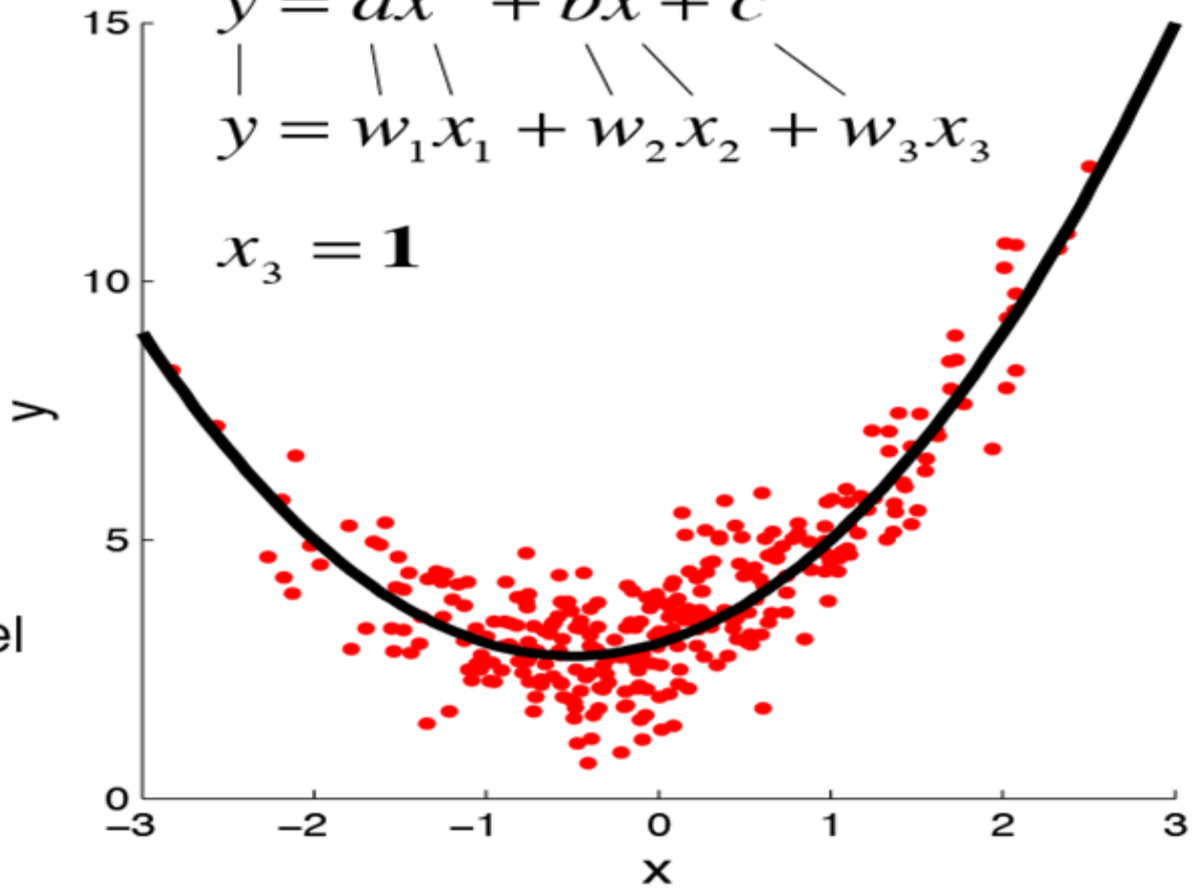


Linearized model

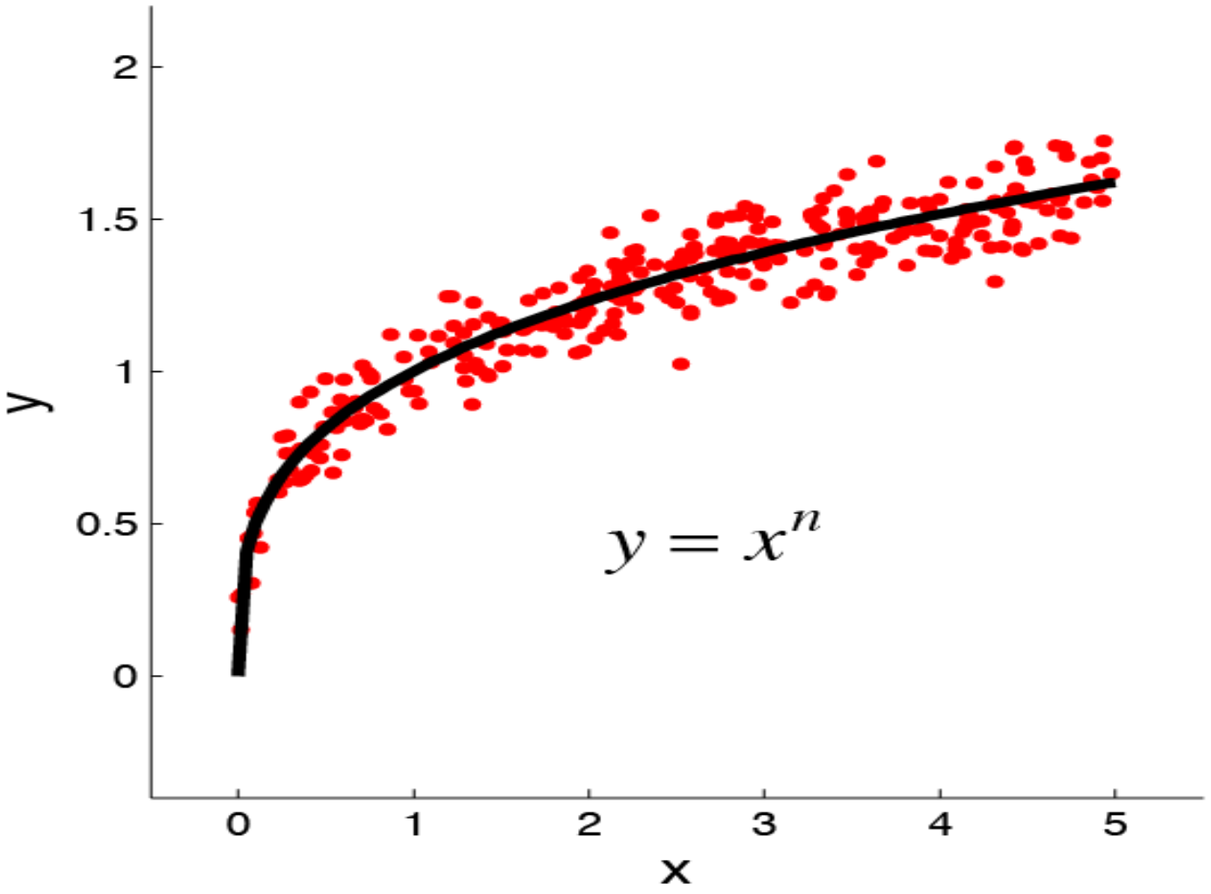
$$y = ax^2 + bx + c$$

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3$$

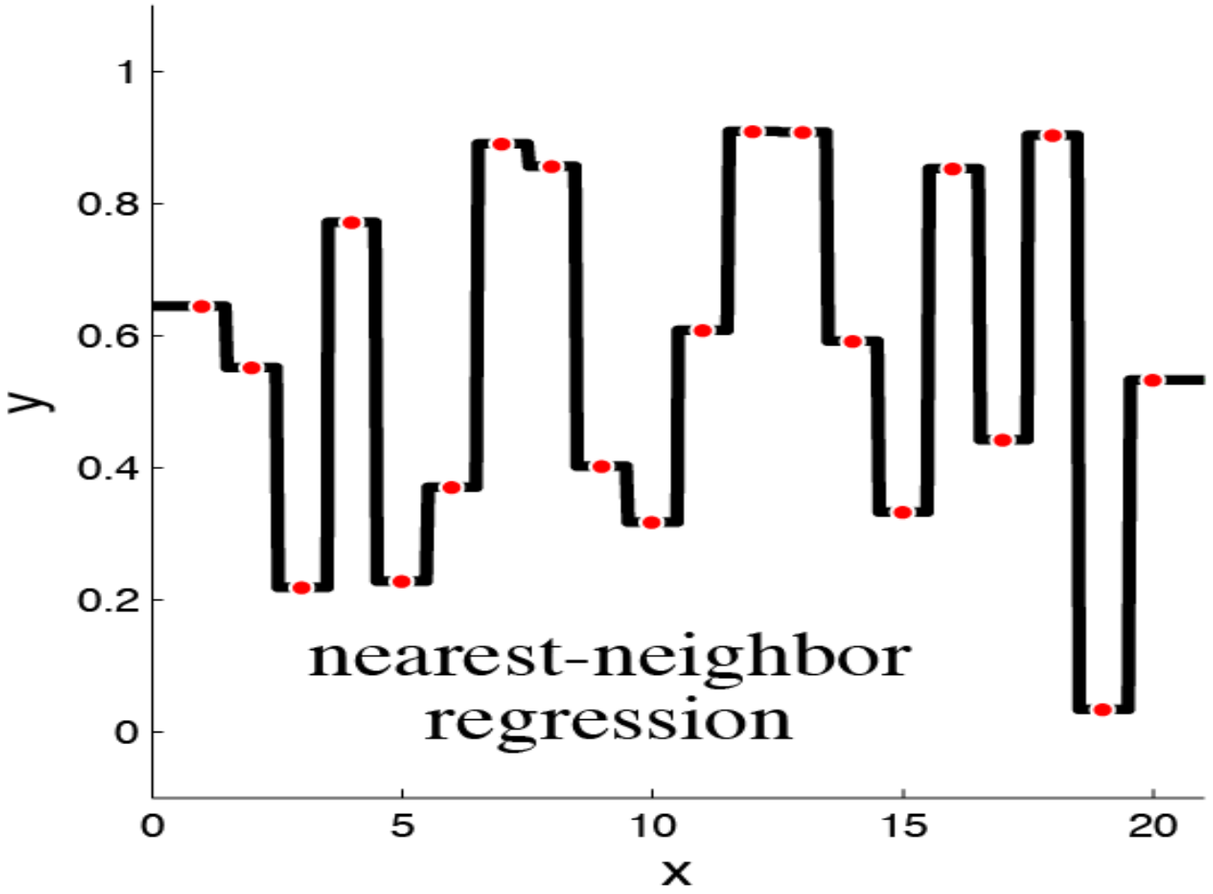
$$x_3 = 1$$



Parametric nonlinear model



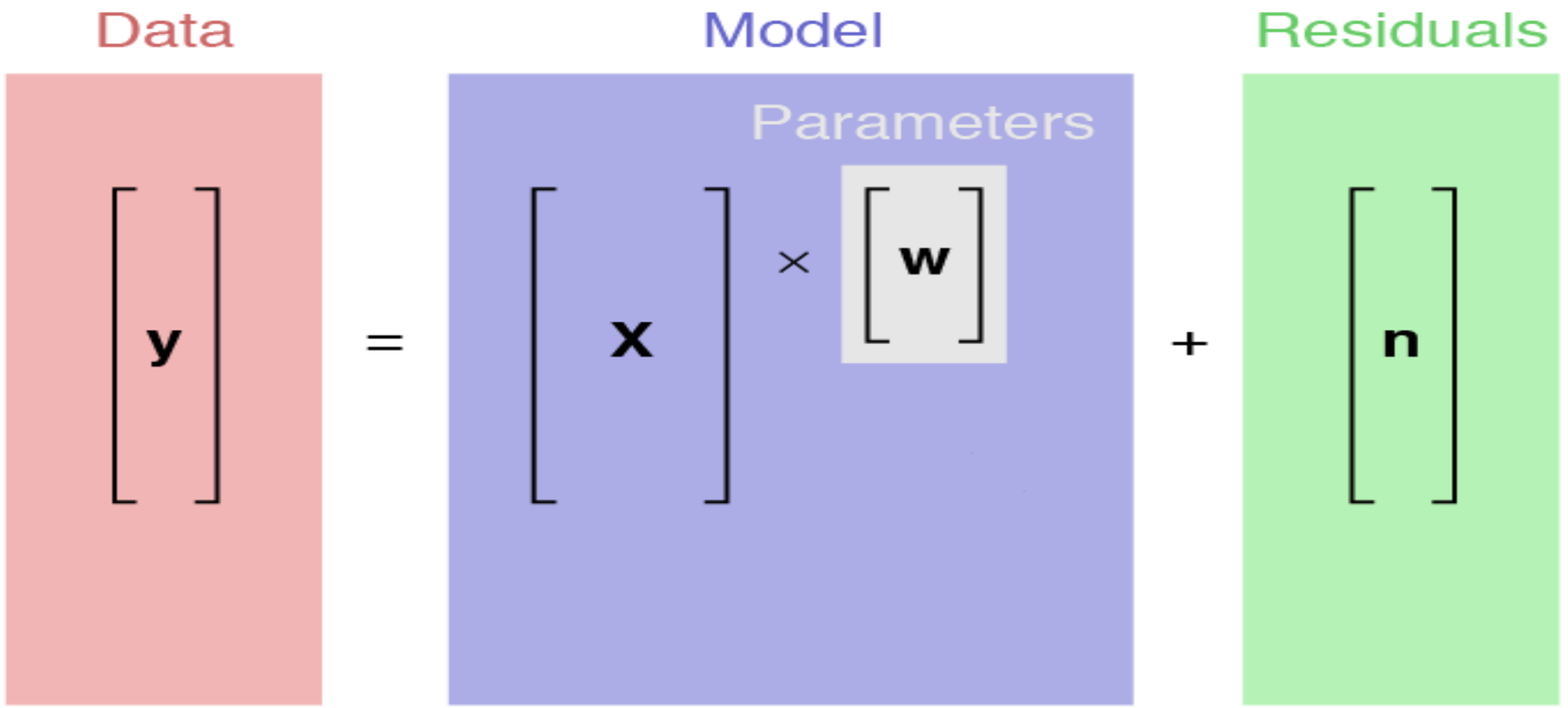
Nonparametric nonlinear model



Characteristics of different types of models

	Linear?	*Parametric?*	*Linear in parameters?*
Linear models	yes	yes	yes
Linearized models	no	yes	yes
Parametric nonlinear models	no	yes	no
Nonparametric nonlinear models	no	no	sometimes

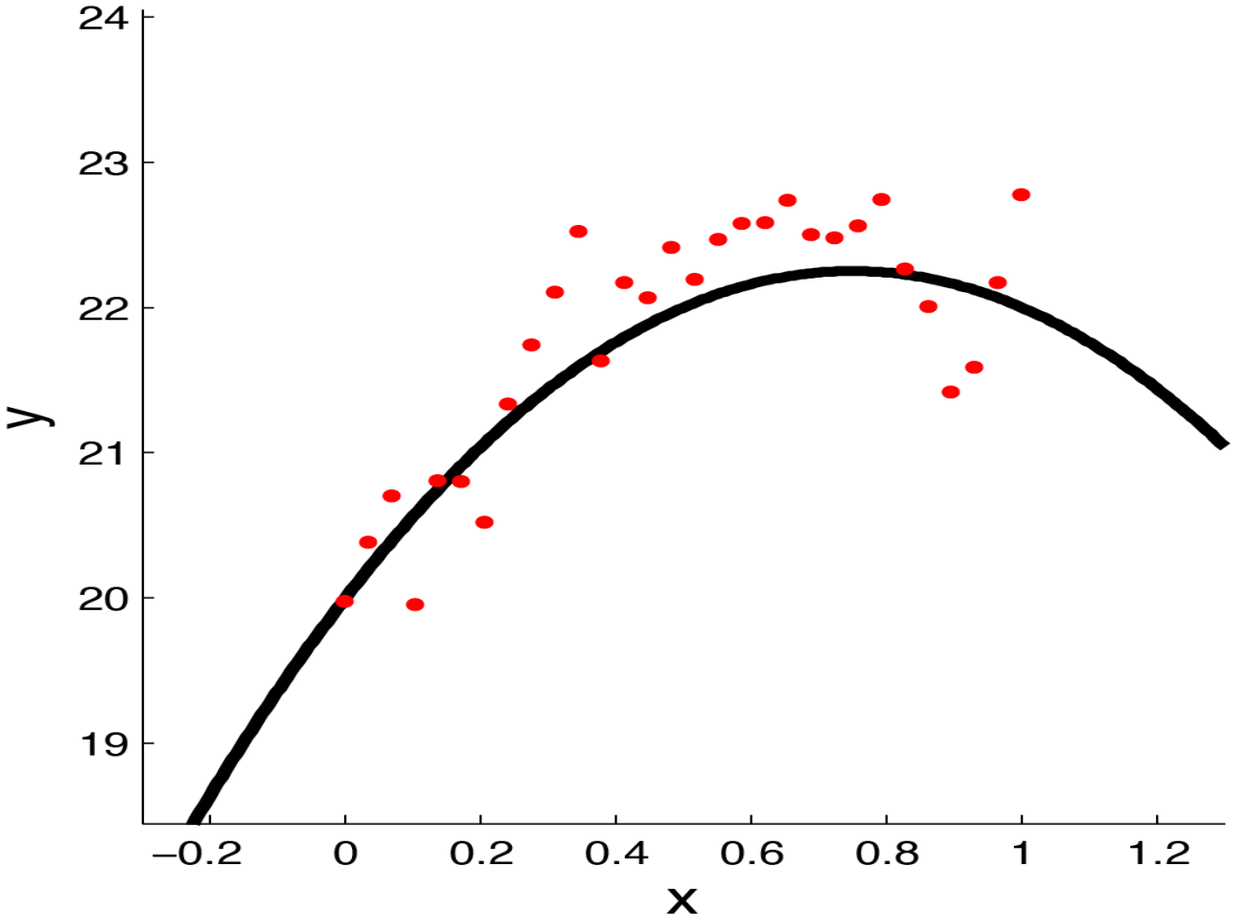
Matrix representation of linear models



$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{n}$$

Model accuracy

Quantifying model accuracy



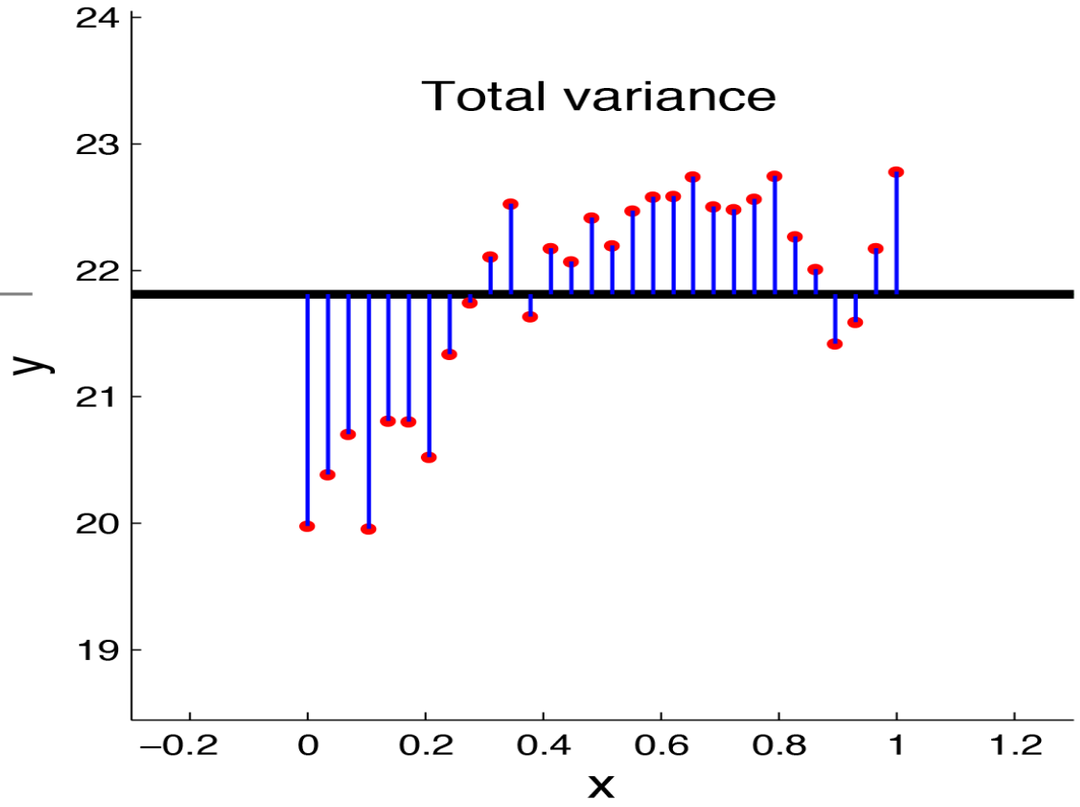
Squared error = 5.4
(dependent on units, hard to interpret)

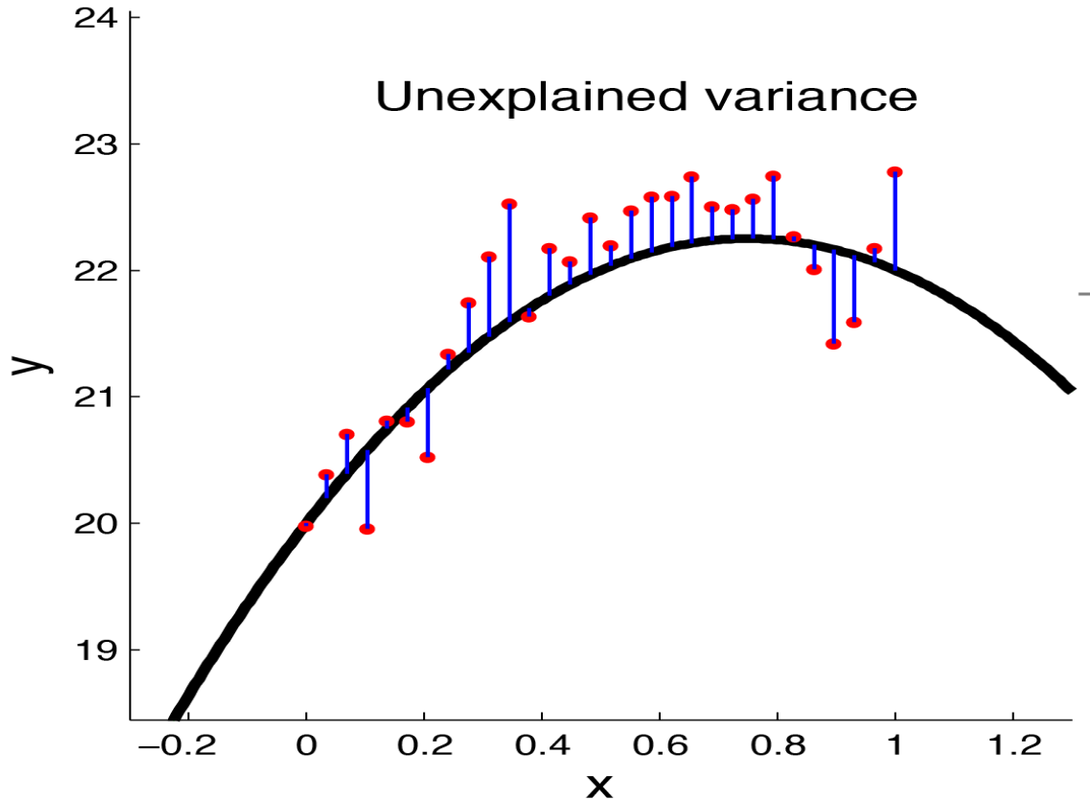
$R^2 = 75\%$
(independent of units, easy to interpret)

Variance

$$\text{variance} = \frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n - 1}$$

Coefficient of determination (R^2)





$$R^2 = 100 \times \left(1 - \frac{\text{unexplained variance}}{\text{total variance}}\right)$$

Coefficient of determination (R^2)

$$R^2 = \text{percent explained variance}$$

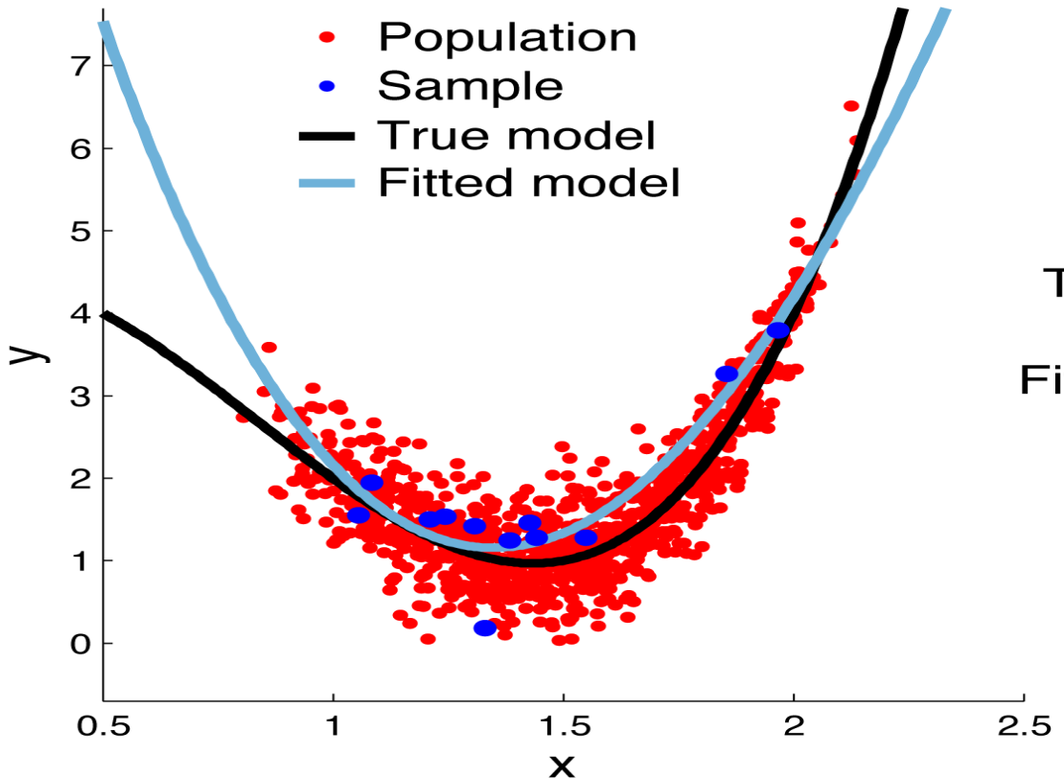
$$R^2 = 100 \times (\text{fraction explained variance})$$

$$R^2 = 100 \times \left(1 - \frac{\text{unexplained variance}}{\text{total variance}}\right)$$

$$R^2 = 100 \times \left(1 - \frac{\frac{\sum_{i=1}^{n}(d_i - m_i)^2}{n-1}}{\frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1}}\right)$$

$$R^2 = 100 \times \left(1 - \frac{\sum_{i=1}^{n}(d_i - m_i)^2}{\sum_{i=1}^{n}(d_i - \bar{d})^2}\right)$$

Direct calculation of R^2 overestimates model accuracy



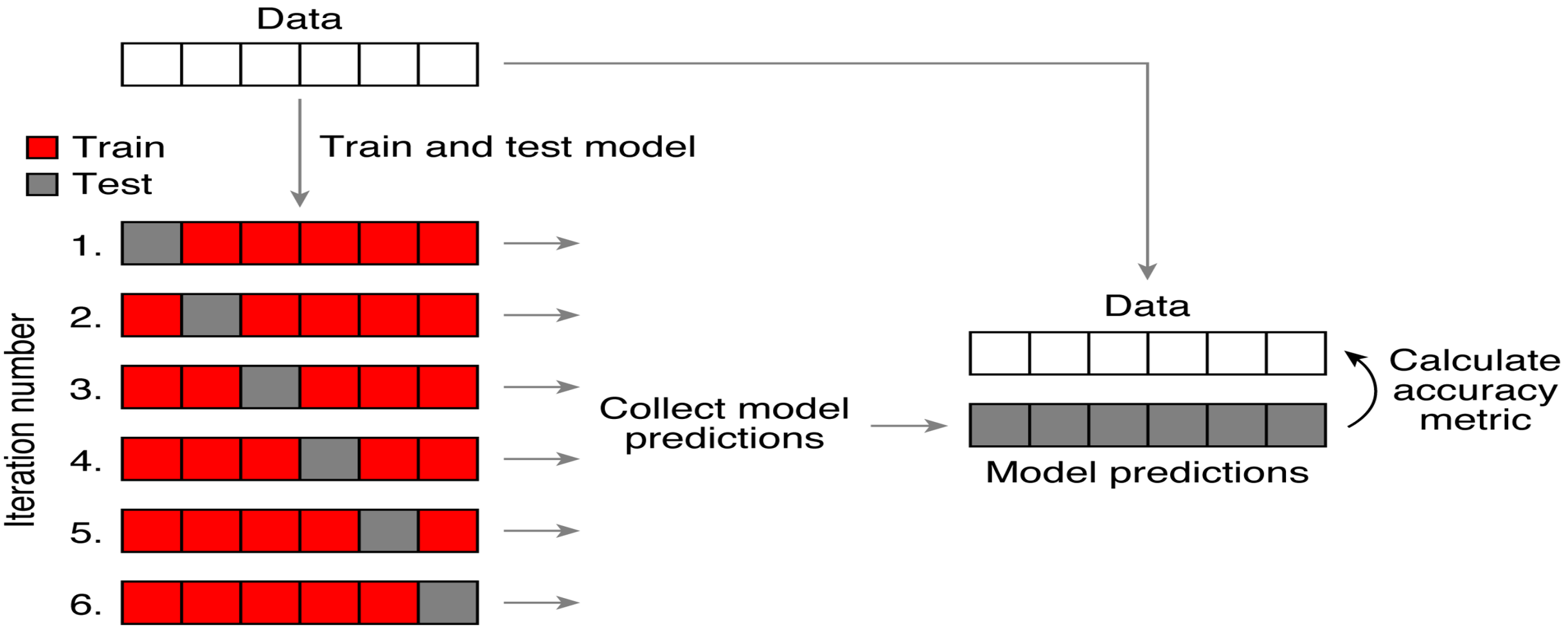
	Population	Sample
True model	high (80%)	high (79%)
Fitted model	low (69%)	very high (85%)

Accuracy of fitted model on sample overestimates true accuracy of fitted model

Cross-validation

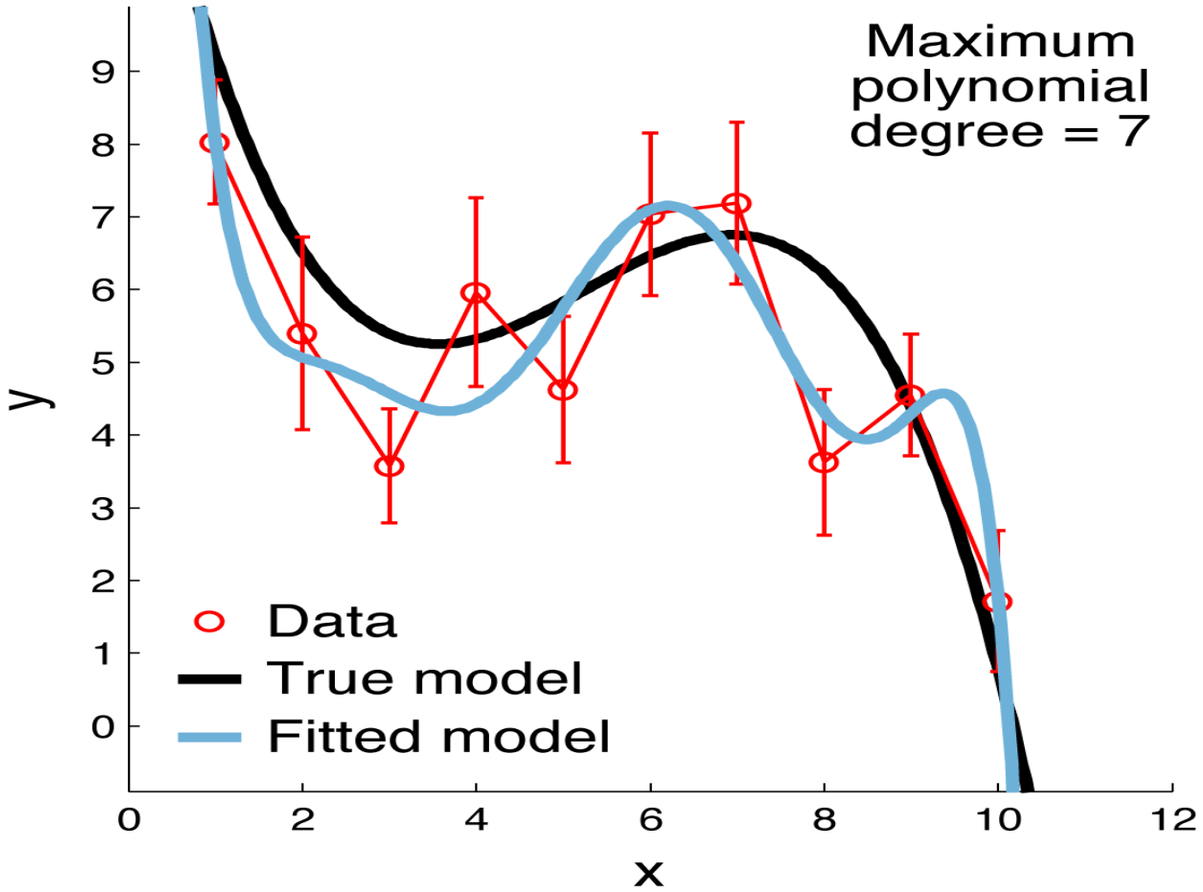
- Goal: estimate true accuracy of a model
- Approach:
  Leave some data out
  Fit model
  Evaluate model on left-out data

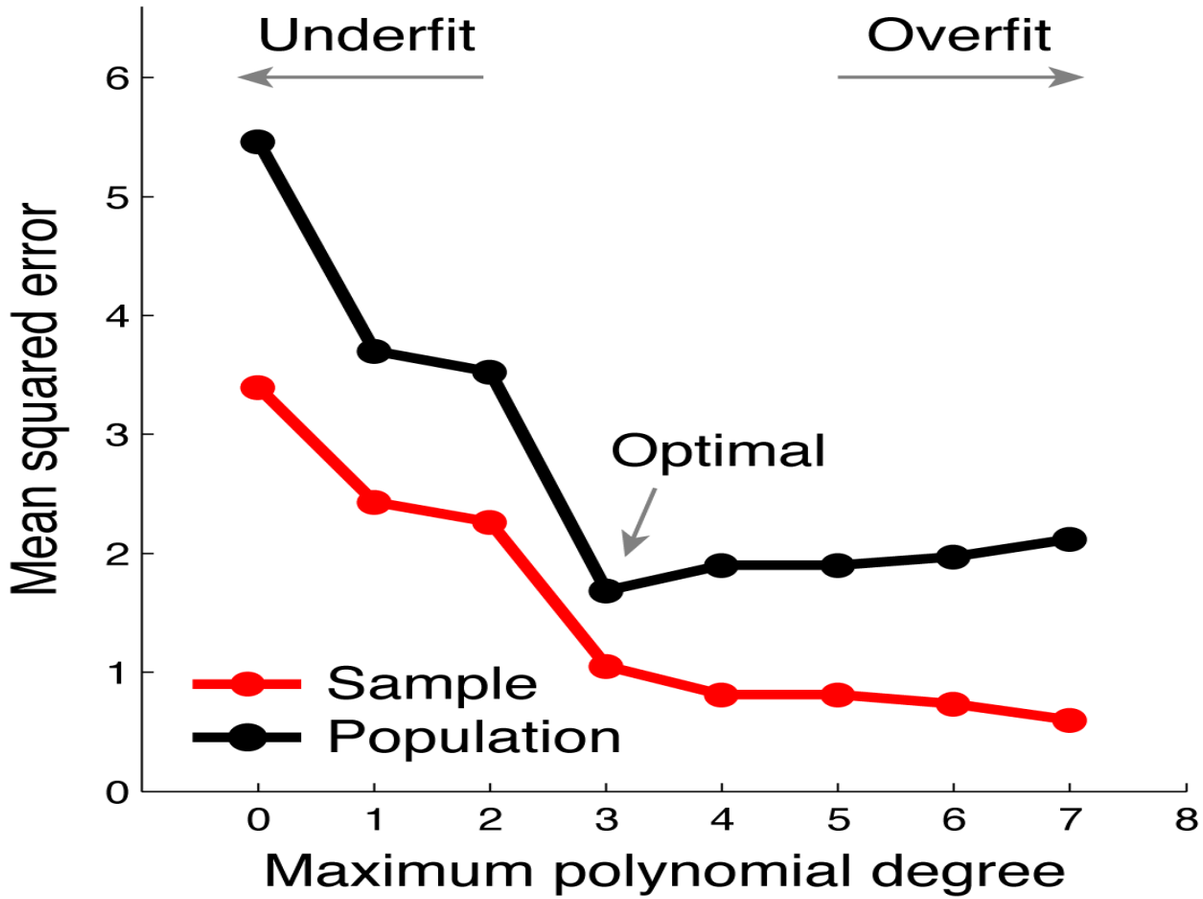
Leave-one-out cross-validation



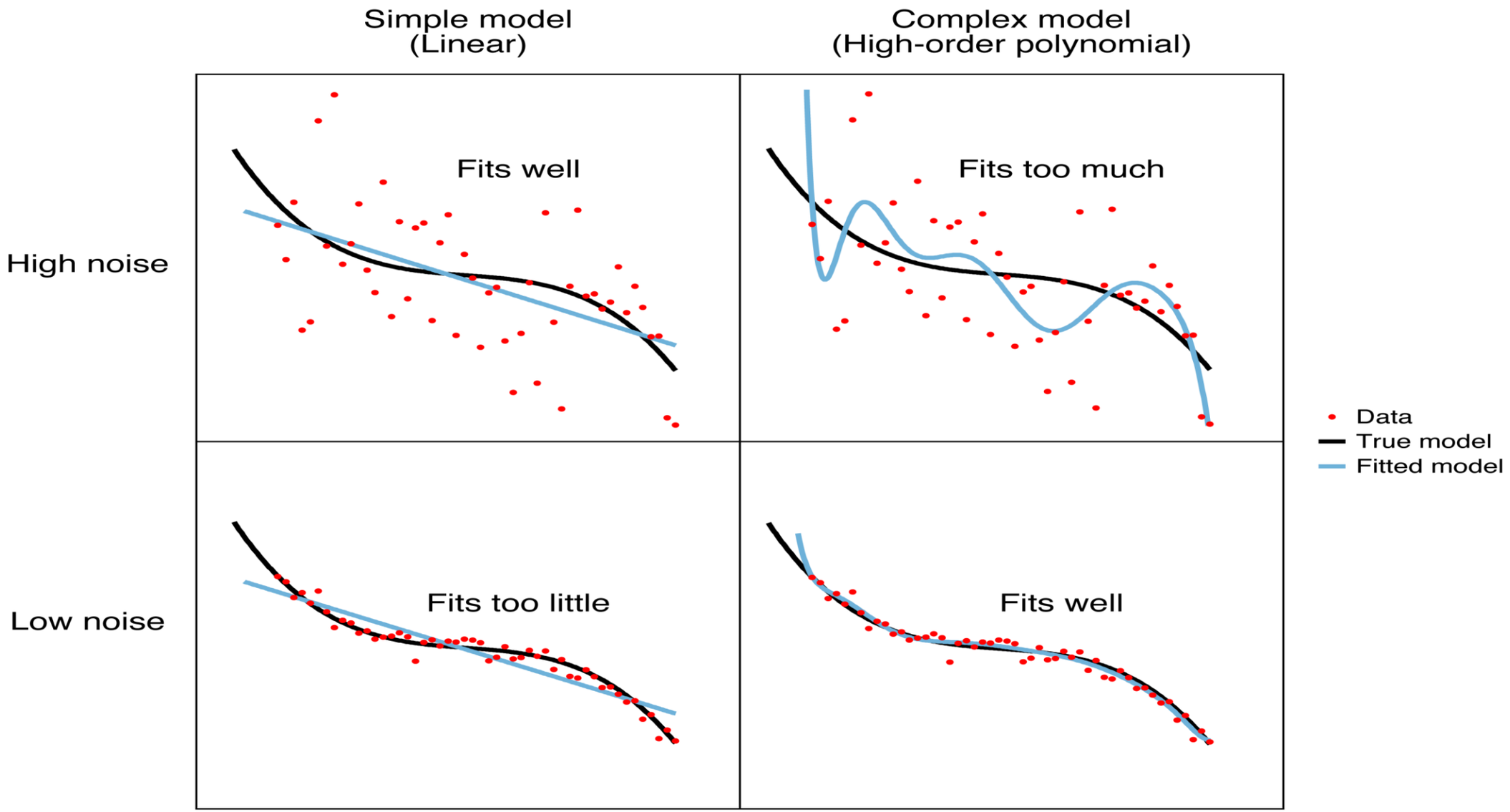
Overfitting

$$y = ax^7 + bx^6 + cx^5 + dx^4 + ex^3 + fx^2 + gx + h$$





Simple models vs. complex models

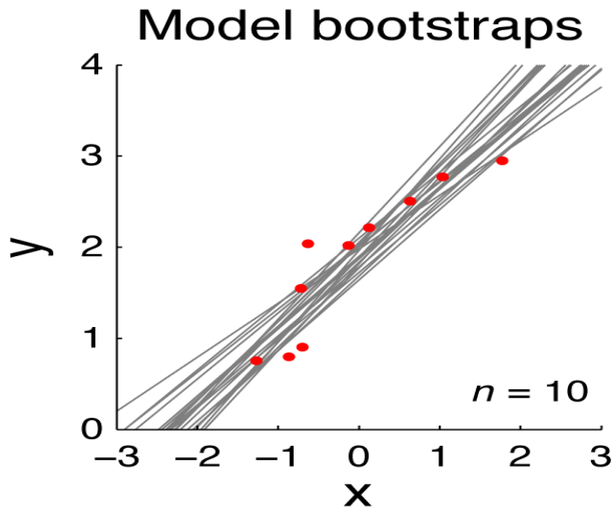


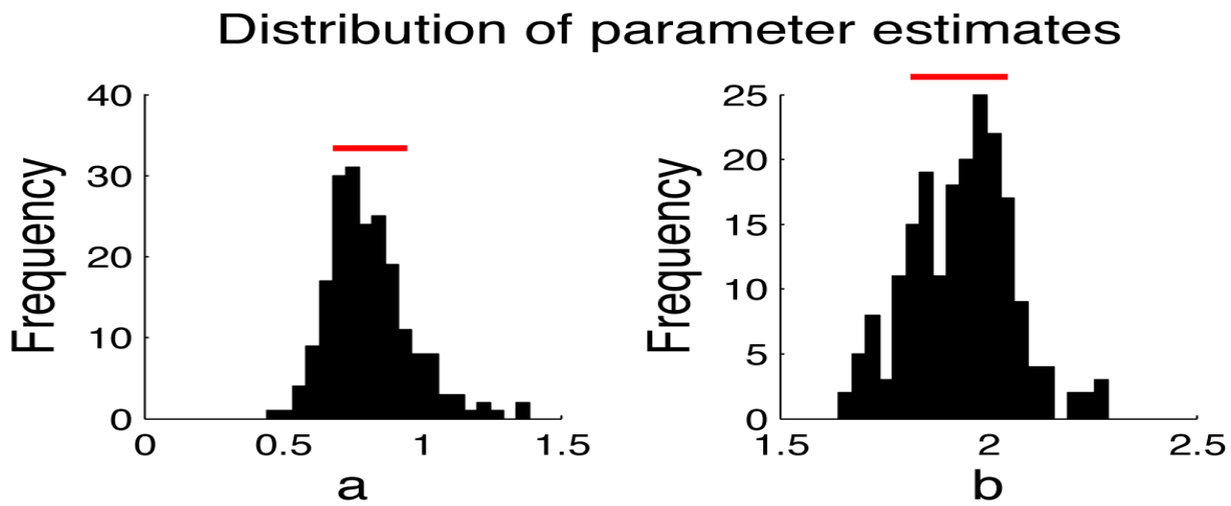
Model reliability

Error bars on model parameters via bootstrapping

True model
$y = 0.5x + 2$

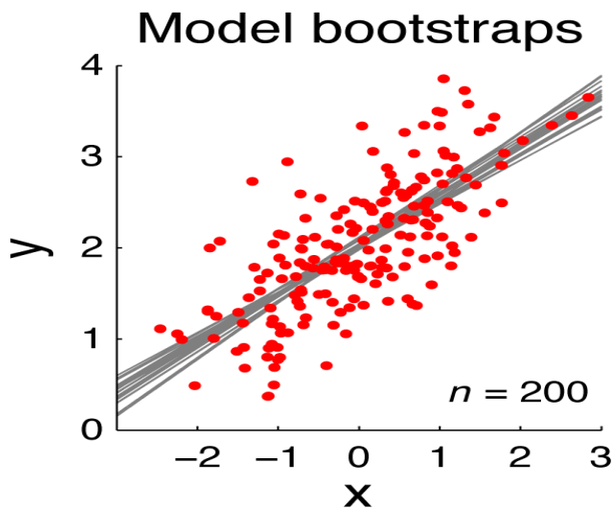
Model to be fitted
$y = ax + b$

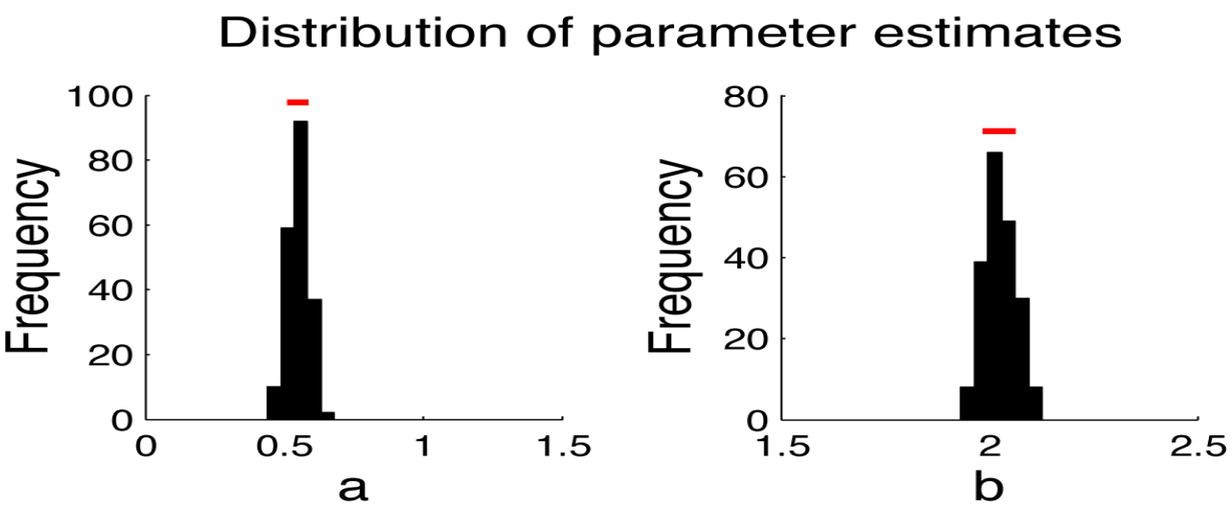




True model
$y = 0.5x + 2$

Model to be fitted
$y = ax + b$

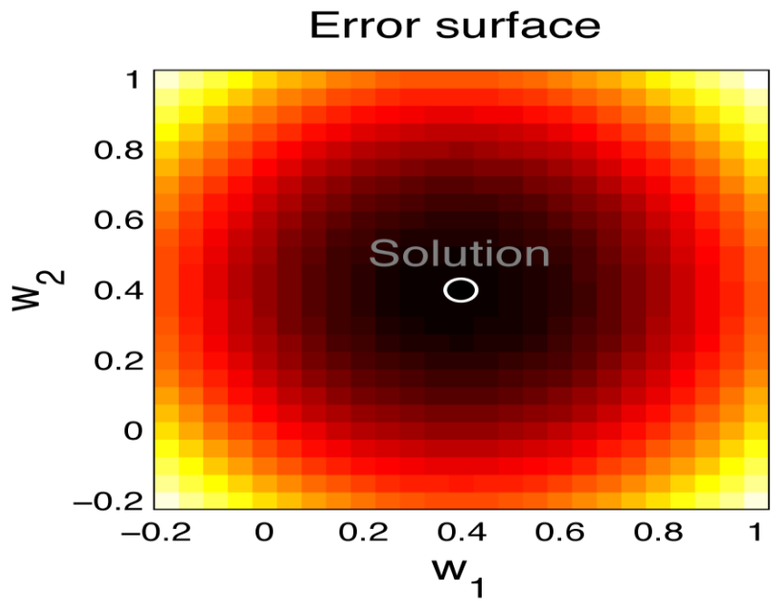


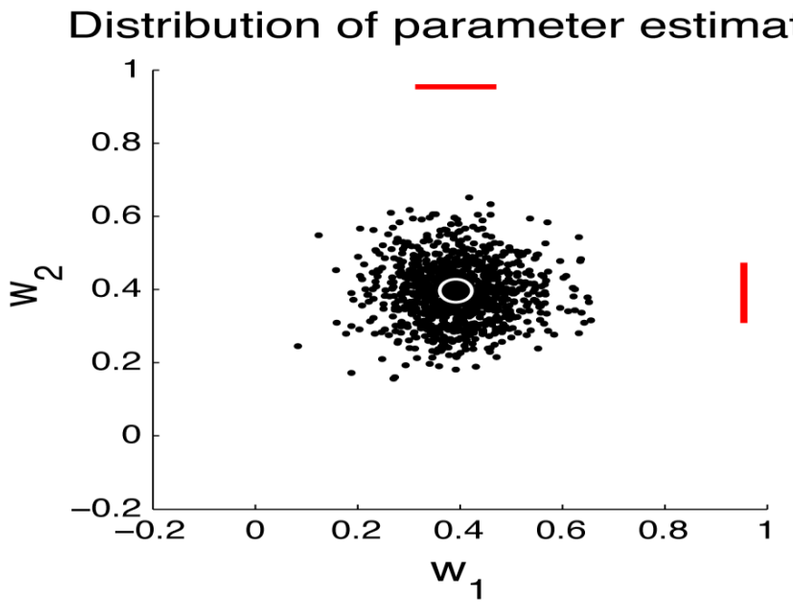


Correlated regressors yield correlated errors

Model
$y = w_1x_1 + w_2x_2$

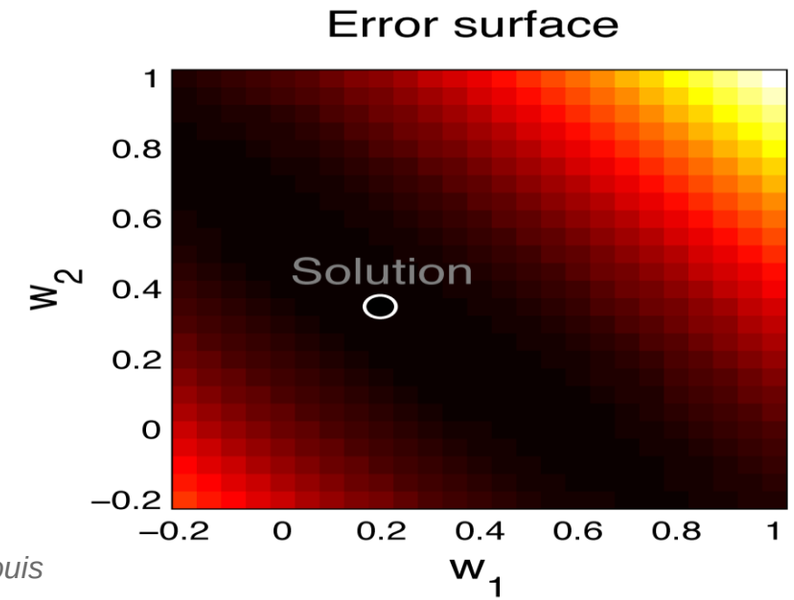
x_1 and x_2
uncorrelated

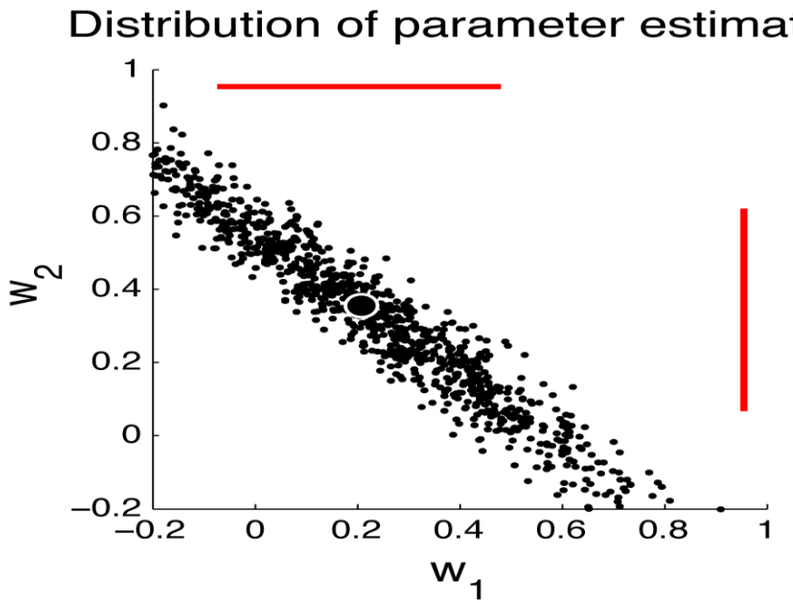




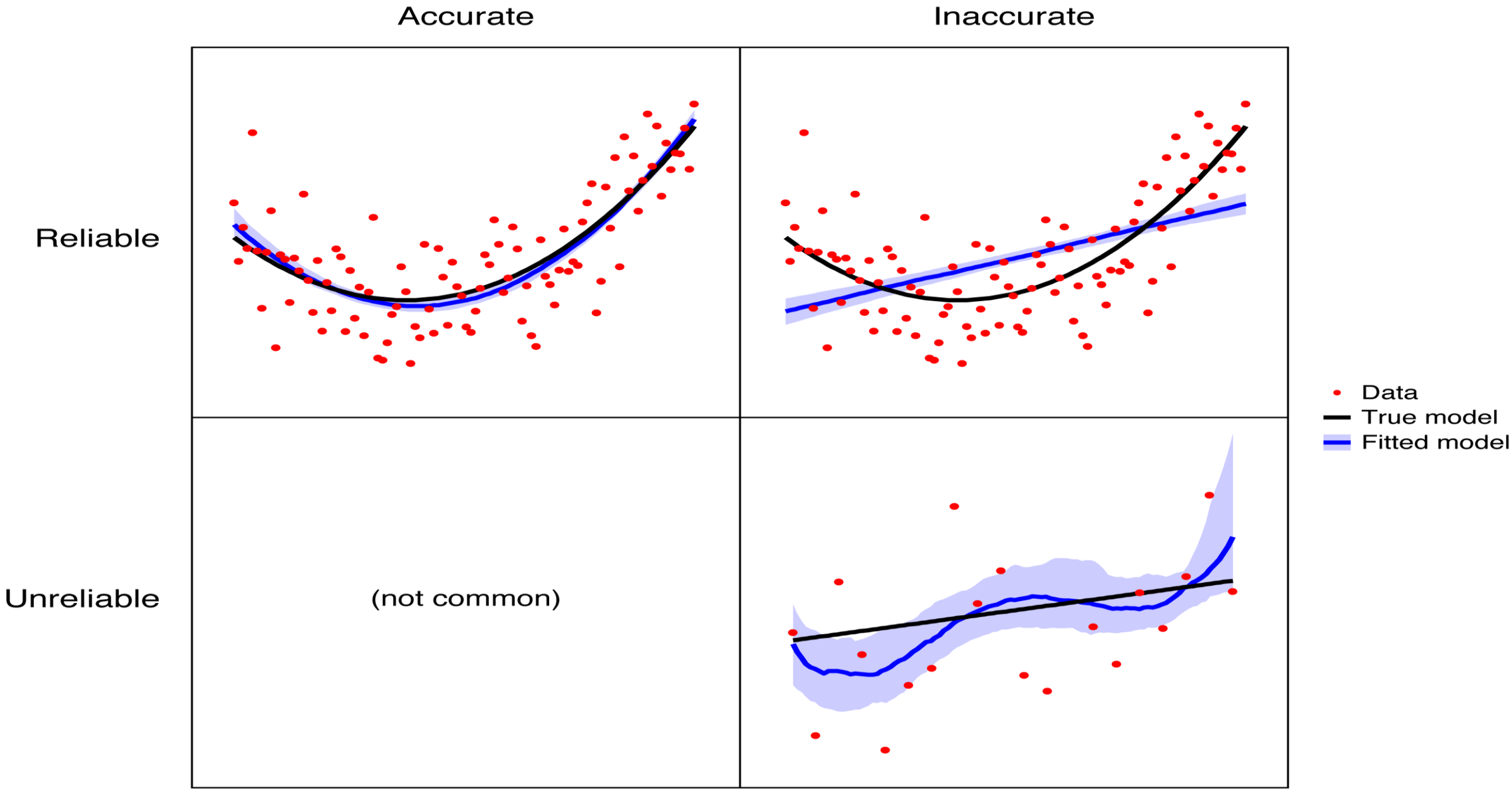
Model
$y = w_1x_1 + w_2x_2$

x_1 and x_2
correlated

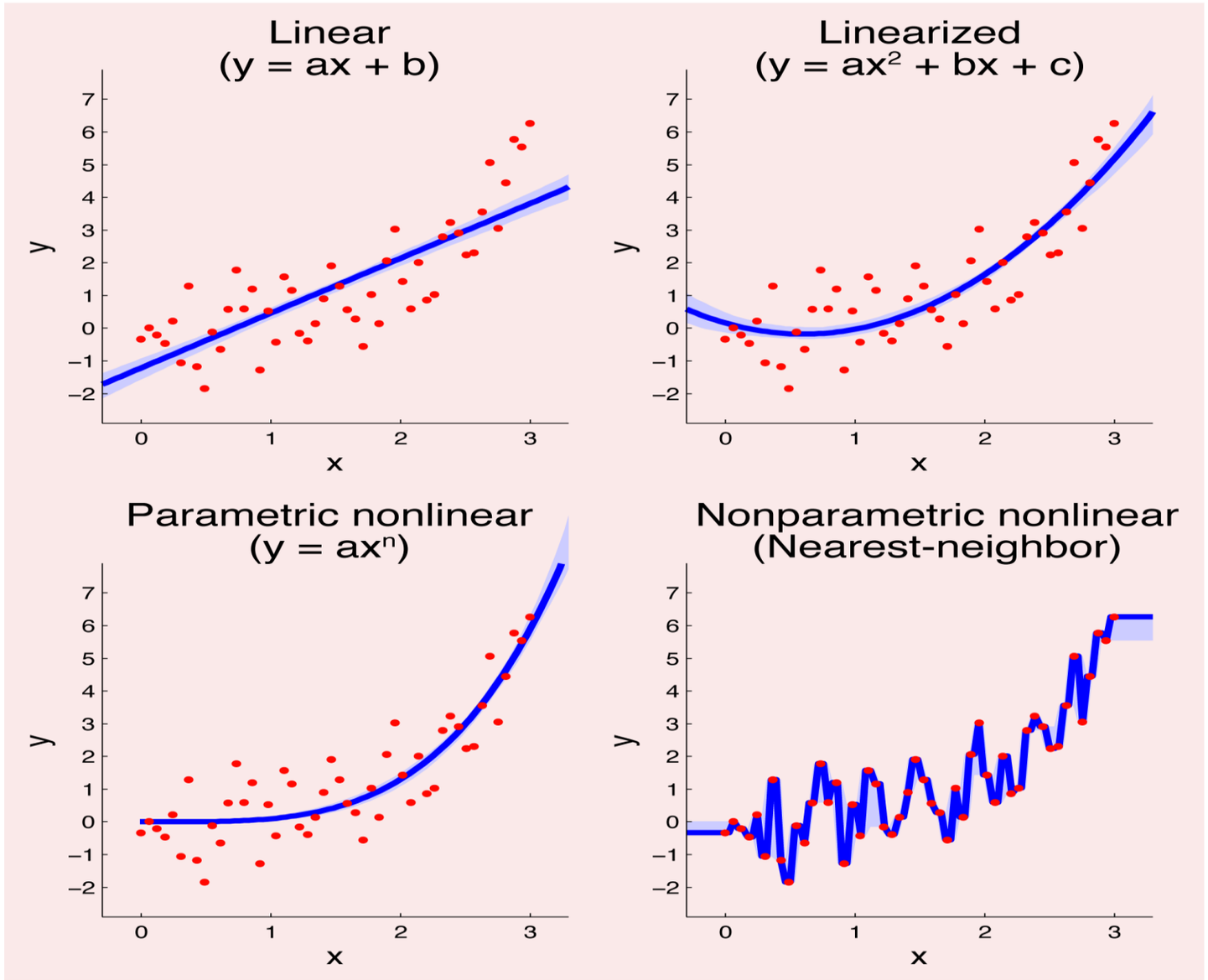




Distinction between accuracy and reliability

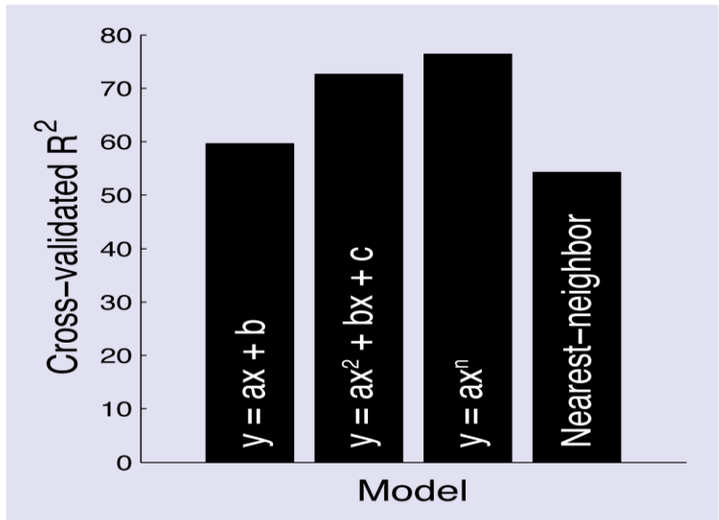


Bootstrapping and cross-validation applies to all models



Use bootstrapping to estimate model reliability

Use cross-validation to estimate model accuracy

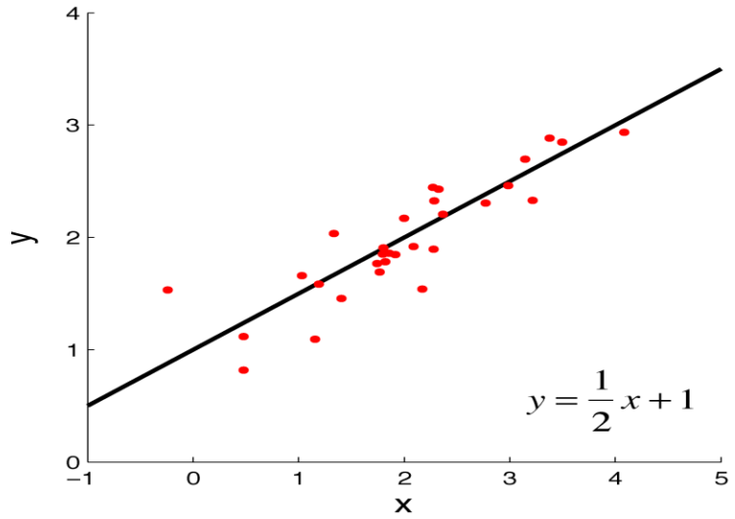
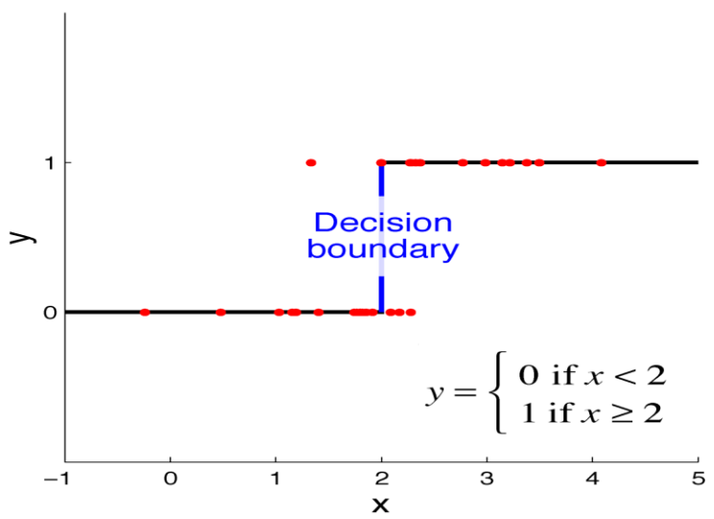
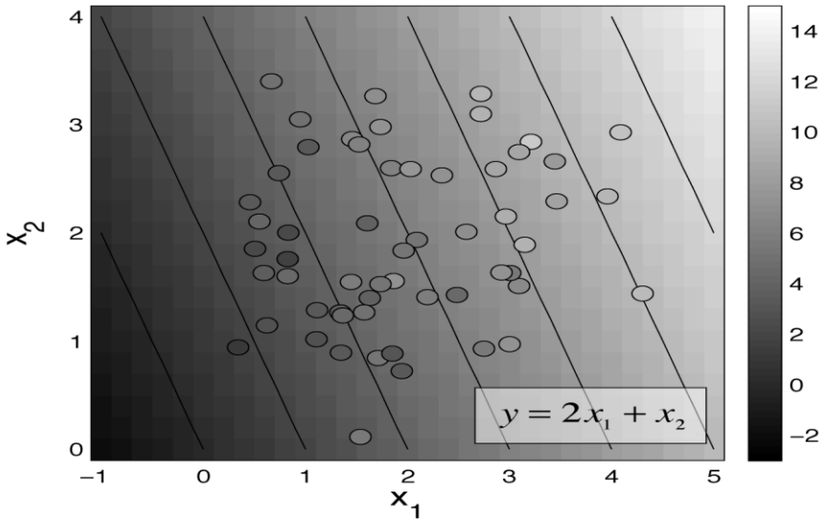
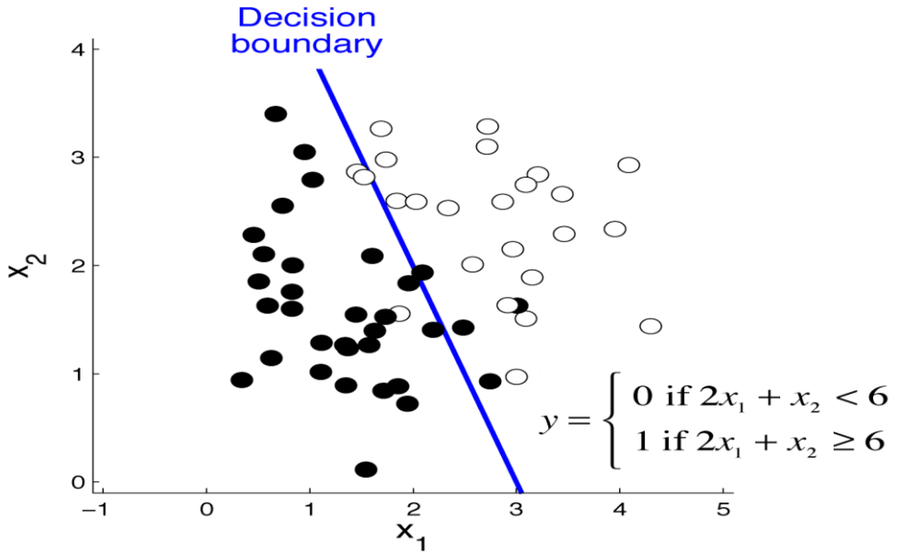


Classification

Linear classification model

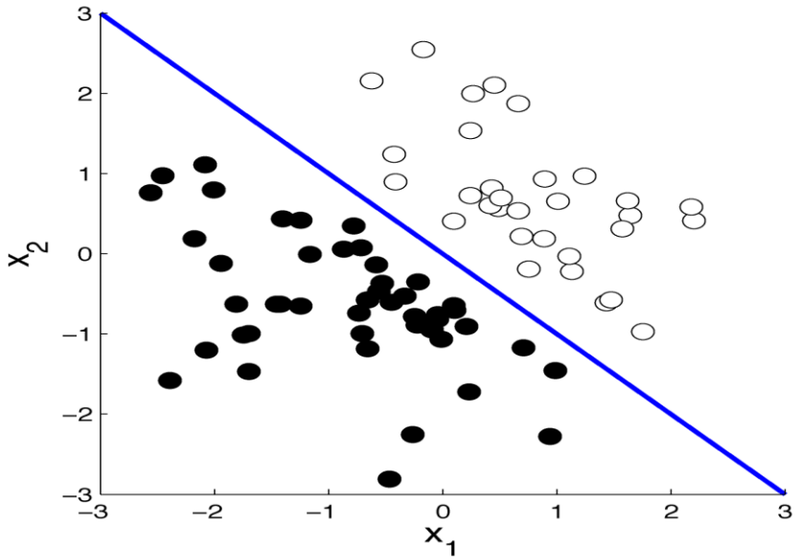
$$y = \begin{cases} 0 \text{ if } \sum_{i=1}^{n} w_i x_i < c \\ 1 \text{ if } \sum_{i=1}^{n} w_i x_i \geq c \end{cases}$$

Comparing regression and classification

	Regression	Classification
1-dimensional input space		
2-dimensional input space		

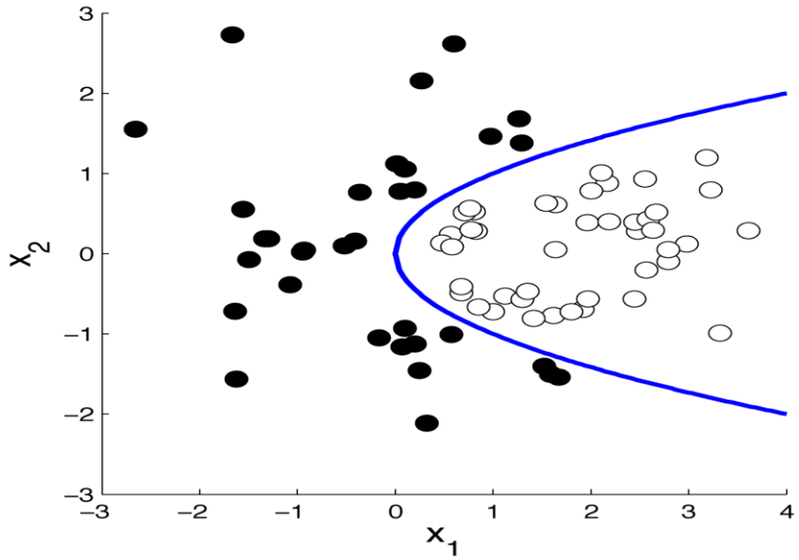
Nonlinear decision boundaries through input space expansion

Linear decision boundary



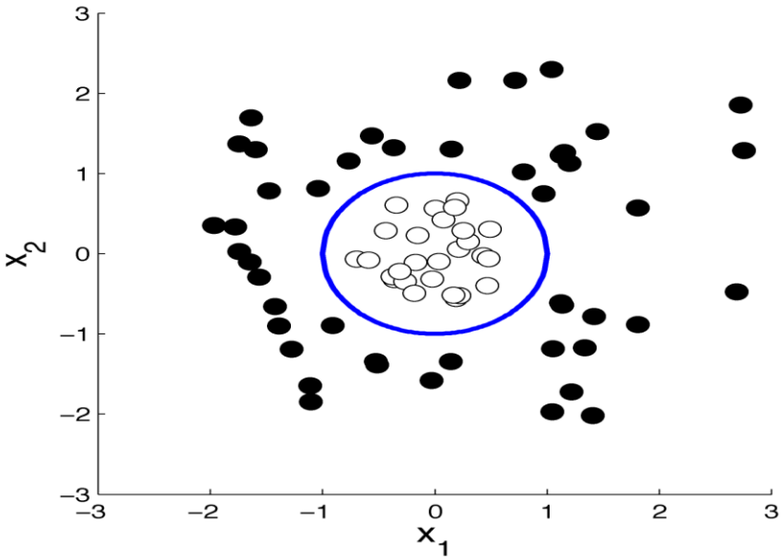
$$y = \begin{cases} 0 \text{ if } x_1 + x_2 < 0 \\ 1 \text{ if } x_1 + x_2 \geq 0 \end{cases}$$

Nonlinear decision boundary



$$y = \begin{cases} 0 \text{ if } x_1 - x_2^2 < 0 \\ 1 \text{ if } x_1 - x_2^2 \geq 0 \end{cases}$$

Nonlinear decision boundary

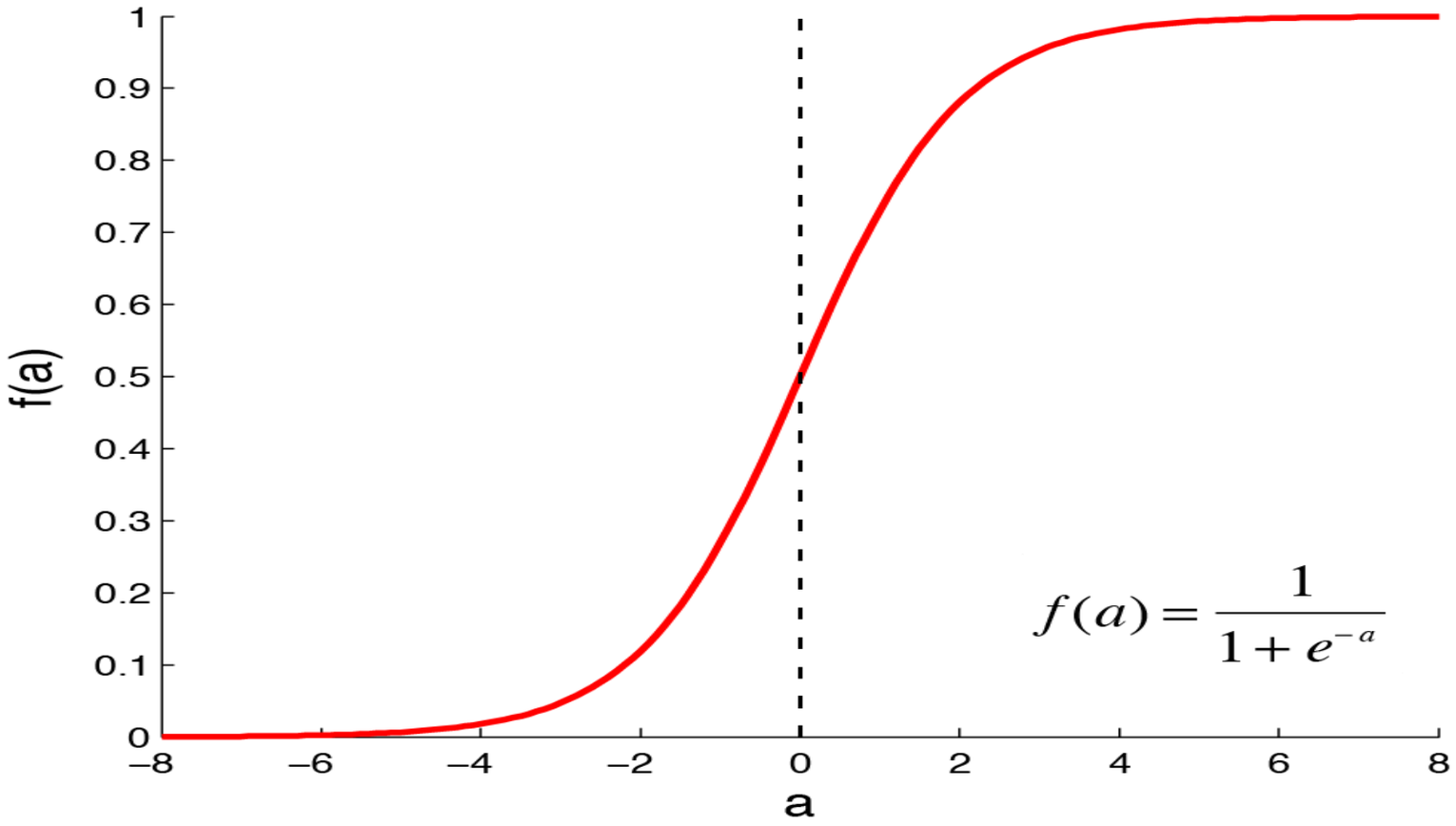


$$y = \begin{cases} 0 \text{ if } x_1^2 + x_2^2 < 1 \\ 1 \text{ if } x_1^2 + x_2^2 \geq 1 \end{cases}$$

Decision boundaries are linear in expanded input space:

$$\{x_1, x_2, x_1^2, x_2^2, x_1 x_2\}$$

Logistic function



Logistic regression

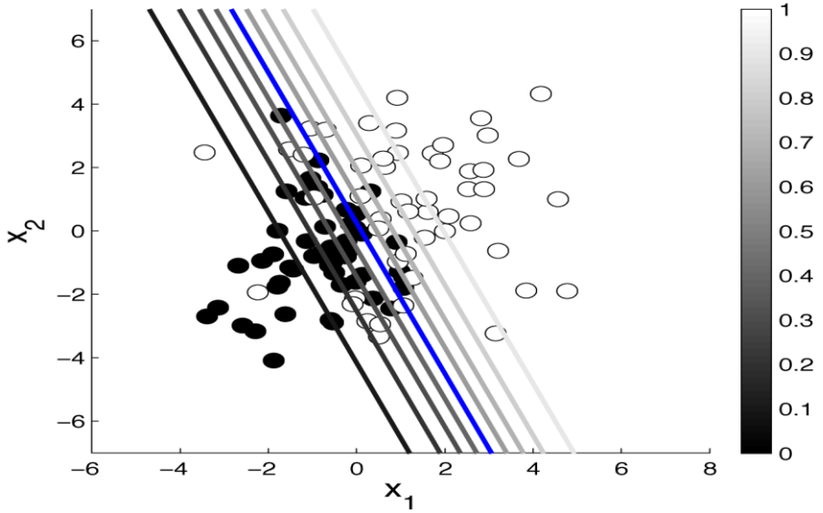
$$y = f\left(\sum_{i=1}^{n} w_i x_i\right) = \frac{1}{1 + e^{\left(-\sum_{i=1}^{n} w_i x_i\right)}}$$

$$\text{likelihood}(d \mid m) = \prod_{j=1}^{m} p(d_j) = \prod_{j=1}^{m} \left(y_j^{d_j} \left(1 - y_j\right)^{1-d_j}\right)$$

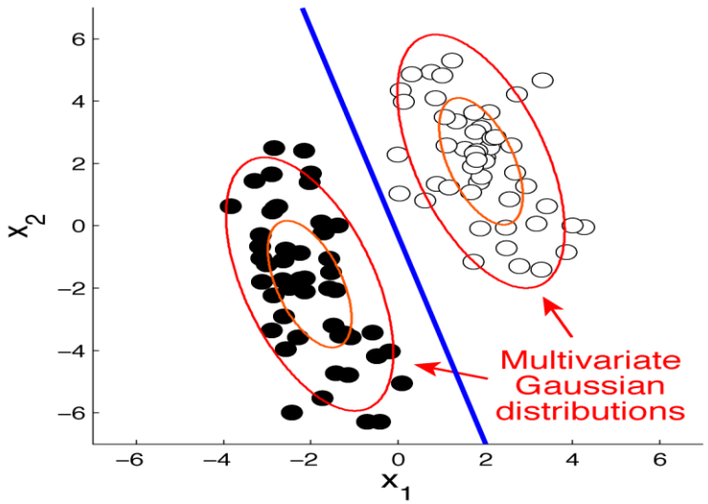
$$\text{negative-log-likelihood}(d \mid m) = -\sum_{j=1}^{m} \left(d_j \log(y_j) + (1 - d_j) \log(1 - y_j)\right)$$

Some classification techniques

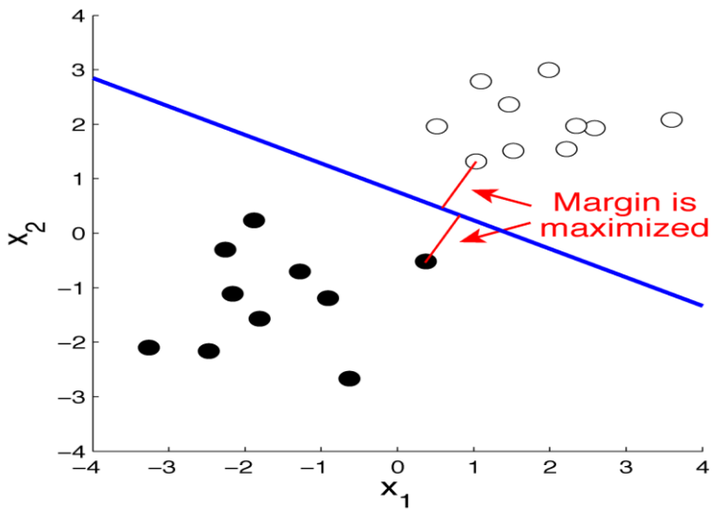
Logistic regression



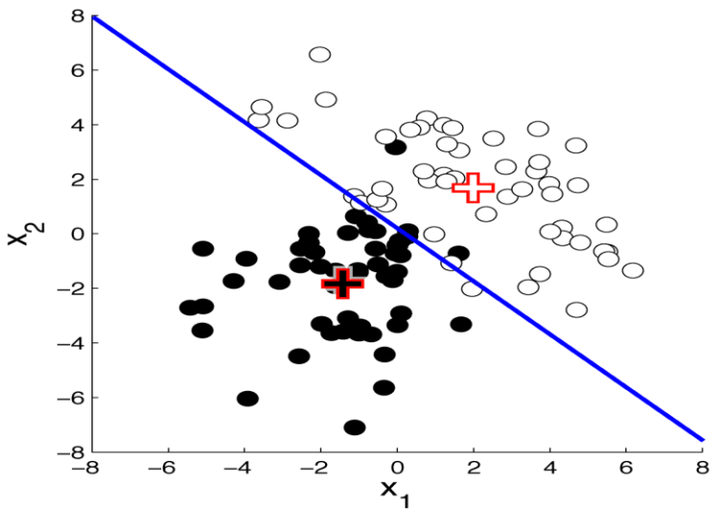
Linear discriminant analysis (LDA)



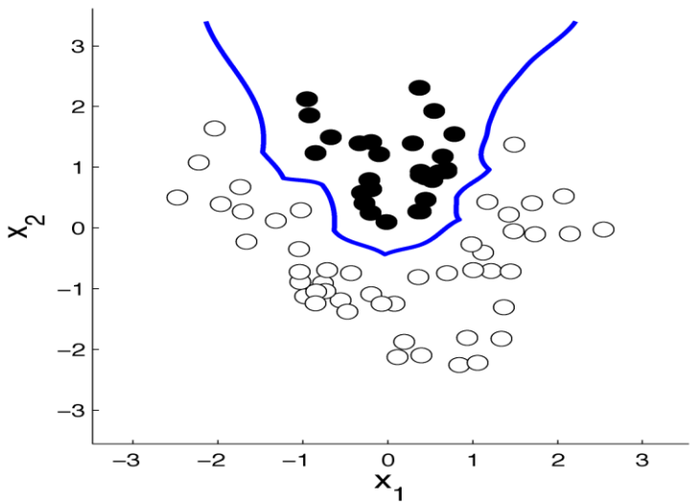
Support vector machines (SVM)



Nearest-prototype classification



Nearest-neighbor classification

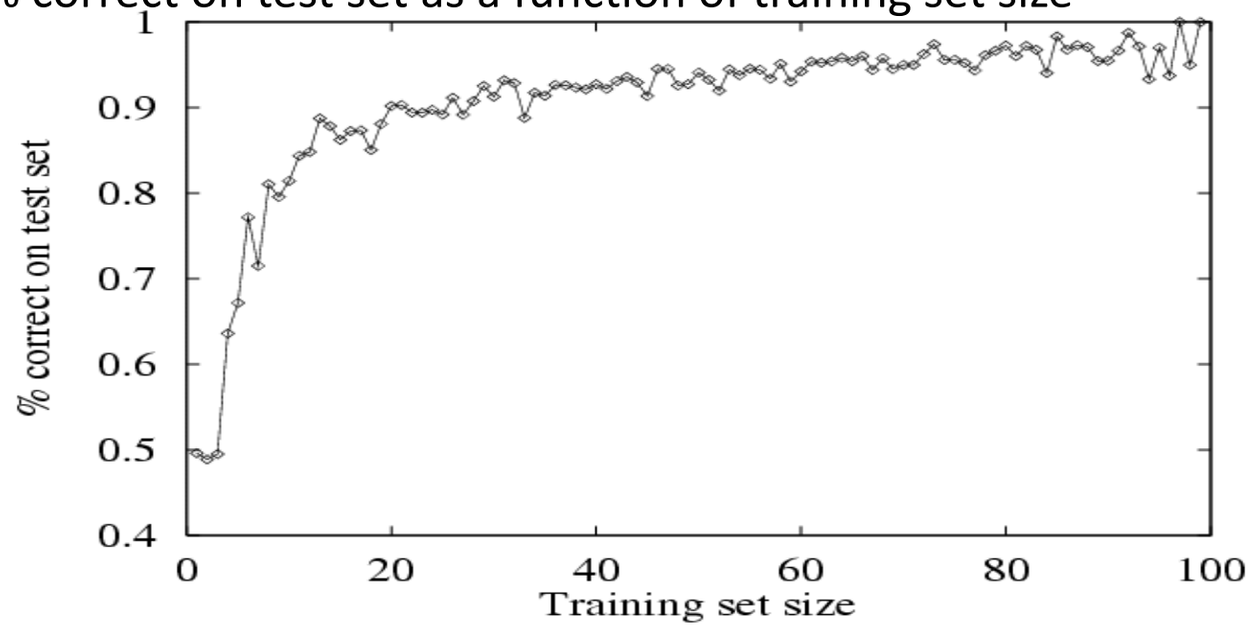


Performance measurement

Performance measurement

- How do we know that $h \approx f$?
  1. Use theorems of computational/statistical learning theory
  2. Try h on a new test set of examples
     (use same distribution over example space as training set)

Learning curve = % correct on test set as a function of training set size



Sınıflandırma algoritmaları için performans ölçüm kriterleri

Sınıflandırma algoritmalarınıdan en etkili olanı belirlemek kullanılan kriter kriterler;

- doğru pozitif (DP) oranı,
- yanlış pozitif (YP) oranı,
- doğru negatif (DN) oranı,
- Yanlış negatif (YN) oranı,
- hassasiyet (precision),
- f-ölçütü (f-measure),
- alıcı işlem karakteristiği eğrisi (ROC: Receiver Operating Characteristic),
- kappa (κ) istatistiği,
- ortalama mutlak hata (MAE: Mean Absolute Error),
- kök ortalama kare hata (RMSE: Root Mean Square Error),
- Matthews korelasyon katsayısı (MCC: Matthews Correlation Coefficient) olarak sıralanır.

Sınıflandırma Algoritmaları İçin Performans Ölçüm Kriterleri

- Doğru Pozitif Oranı (Duyarlılık)
- Yanlış Pozitif Oranı
- Precision (Hassasiyet)
- F-Ölçütü
- Alıcı İşlem Karakteristiği Eğrisi (ROC: Receiver Operating Characteristic)
- Kappa (κ) İstatistiği
- Ortalama Mutlak Hata (MAE: Mean Absolute Error)
- Kök Ortalama Kare Hata (RMSE: Root Mean Square Error)
- Matthews Korelasyon Katsayısı (MCC: Matthews Correlation Coefficient)

Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,
Dr. Cahit Karakuş
cahitkarakus@gmail.com