İKT484 Makine Öğrenmesi

1. Makine Öğrenmesi Kavramları (3/3)

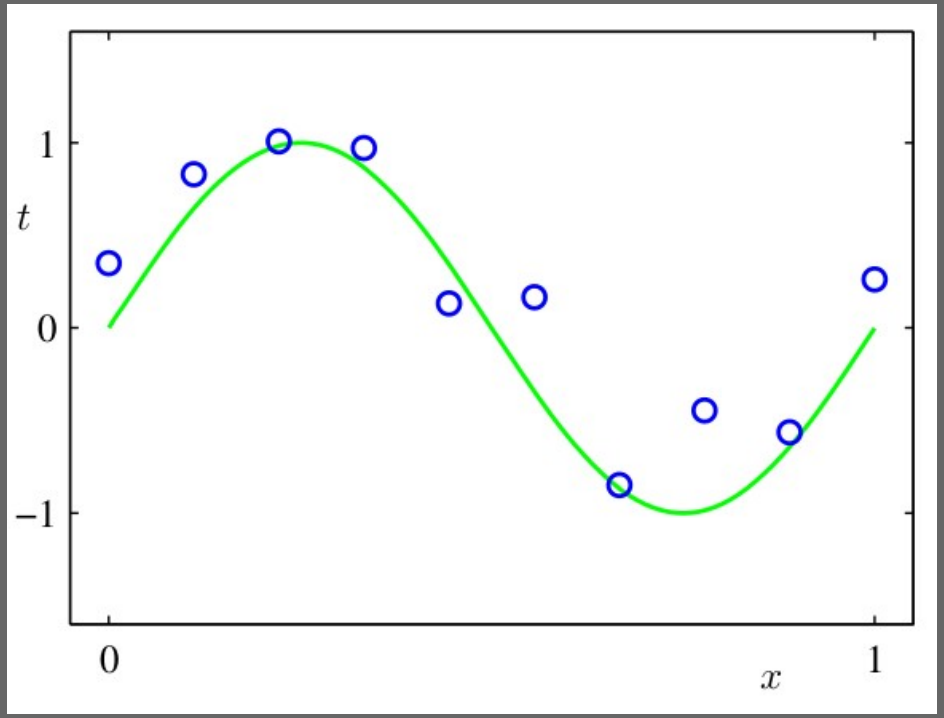
Öğretim Görevlisi:
Bora GÜNGÖREN
bora.gungoren@atilim.edu.tr

Uygulamalı Örnek

- Polinom eğri uydurmanın temel problemine bakacağız (Bishop 2006'dan uyarlanmıştır).
  - Bu, parametrelerin (polinom katsayılarının) optimize edilmesini içerecek ve bir tahmin problemi olarak sınıflandırılacaktır.

Uygulamalı Örnek

- Gerçek fonksiyonumuz bir sinüs dalgasıdır $t=\sin(2\pi x)$.
  - Bize bir eğitim seti verildi $X=(x_1, x_2, \ldots, x_N)^T$
  - Veri seti, veri ve biraz (Gauss) gürültü üretti.



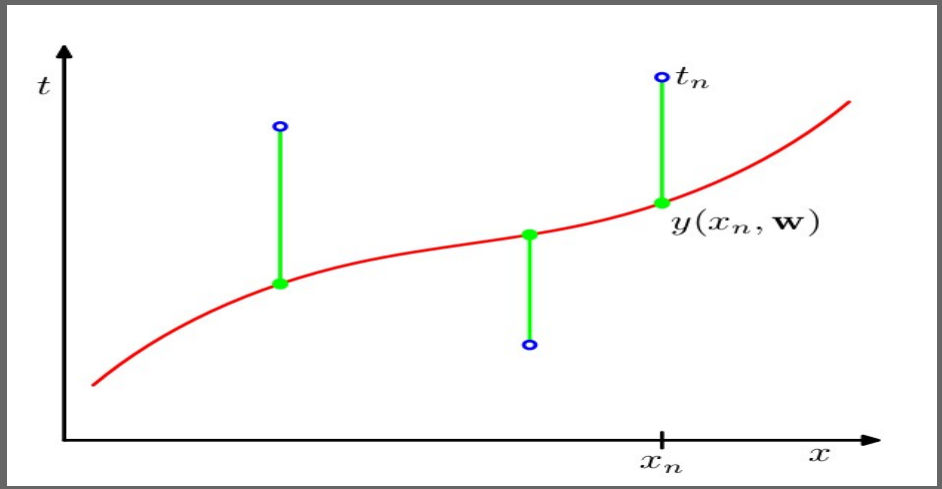
Uygulamalı Örnek

- Bir modelimiz ve model parametrelerimiz olmalı.
  - $w=(w_0, w_1,w_2,...w_M)$ model parametreleridir, burada M polinomun boyutudur.

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \qquad (1.1)$$

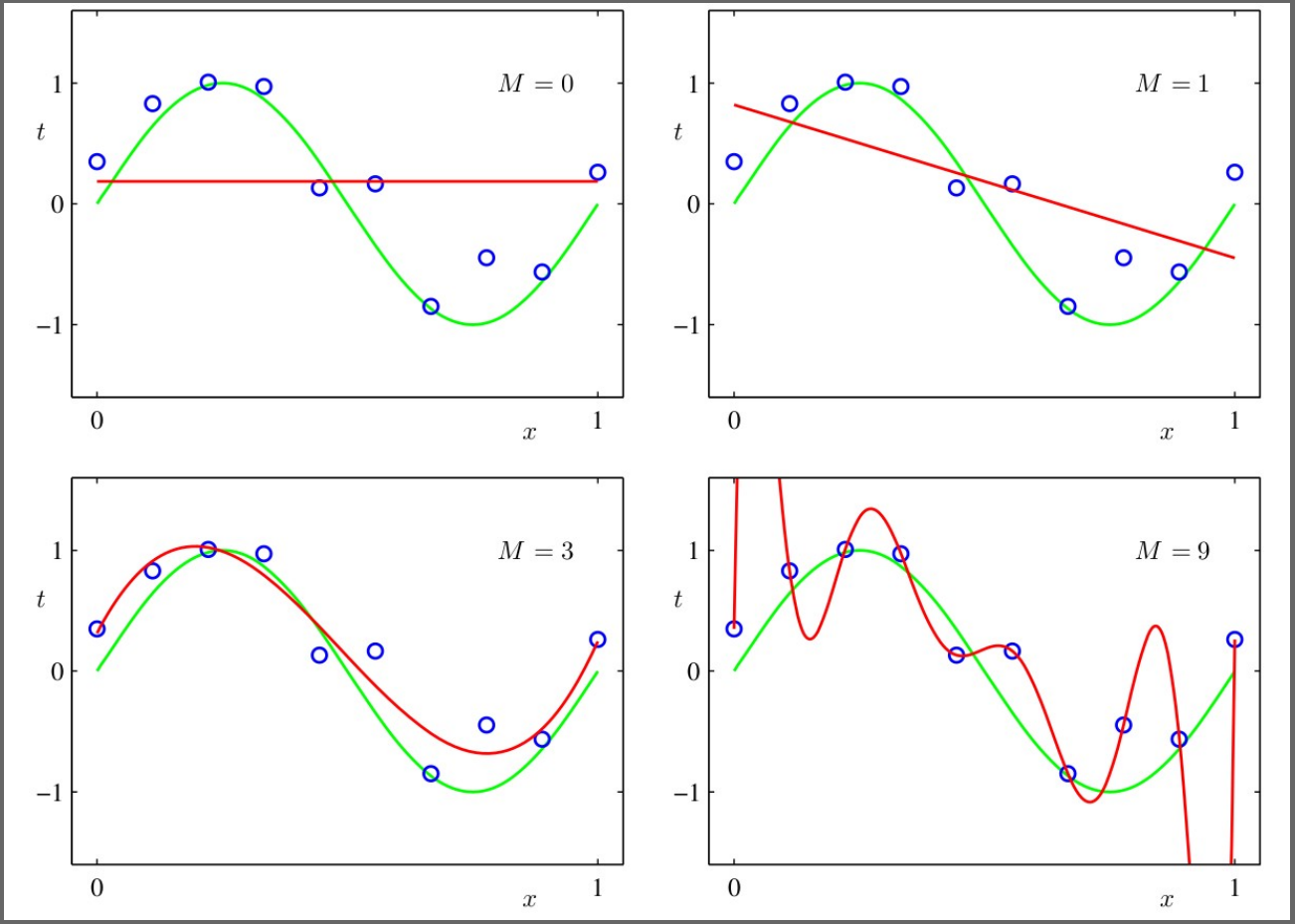
Uygulamalı Örnek

- Parametre tahmin problemimiz artık "w'yi (vektör olarak) en aza indirmek için tahmin et" şeklinde ifade ediliyor.
  - Hata, eğitim setindeki tahmini y_N ve gerçek t_N kullanılarak hesaplanan E(w) olarak tanımlanıyor.



$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 \qquad (1.2)$$

Uygulamalı Örnek



Uygulamalı Örnek

- M=3, M=9'dan daha fazla eğitim verisi kümesinde hataya sahiptir, ancak eğri için daha iyi bir genel uyum sağlayacaktır.
  - Bu nedenle gerçek performans pratik uygulama ile ölçülmelidir.
  - M=9 ile ilgili sorun, aşırı uyum dediğimiz şeydir.

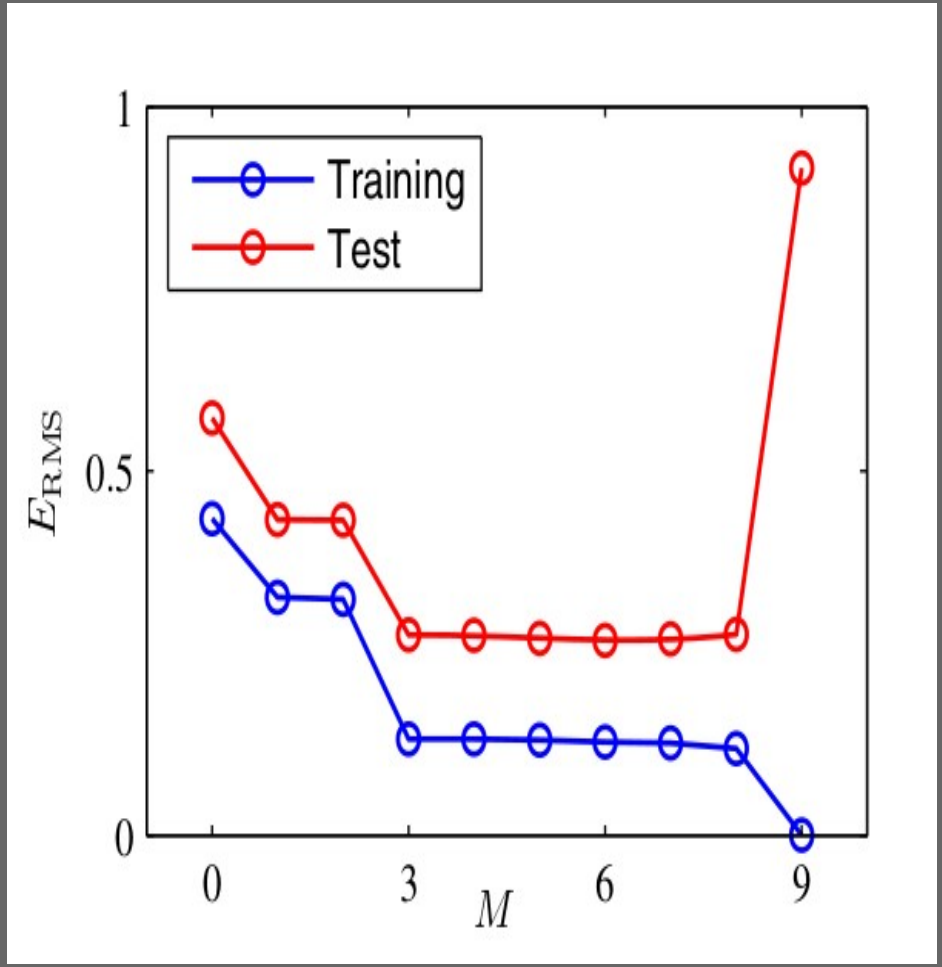
Uygulamalı Örnek

- Performansı ölçmek için, 100 adetlik yeni bir test veri kümesi (yine gürültü ekleyerek) oluşturuyoruz.
  - Ardından hata ölçüsünü ERMS olarak tanımlıyoruz.
  - ERMS'yi hesaplayalım

$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N} \tag{1.3}$$

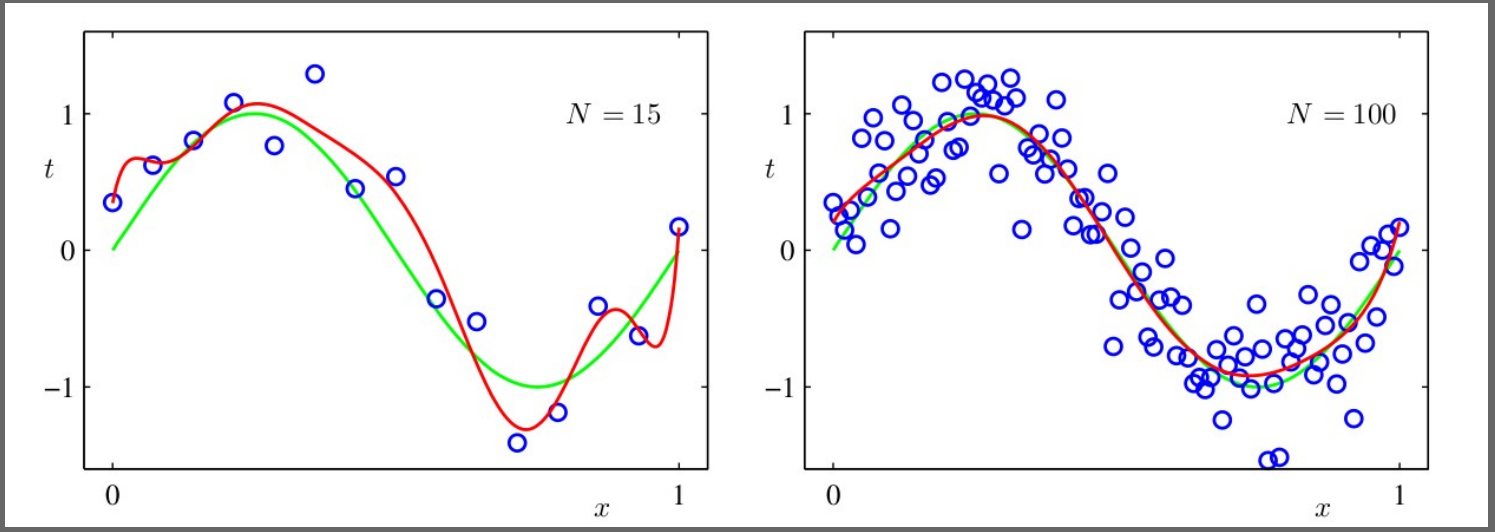
Uygulamalı Örnek

- Gördüğümüz şey
  - Aşırı eğitilmiş modelde, ERMS test veri kümesi ile birlikte çılgınca artar.
  - M=3'ten M=8'e kadar hata minimum görünüyor.
  - Bu yüzden en basit formu, M=3'ü koruyoruz.



Uygulamalı Örnek

- M=9 modelindeki hatayı azaltmanın bir yolu daha büyük bir eğitim veri kümesiyle eğitim yapmaktır.



Uygulamalı Örnek

- Daha büyük bir eğitim verisi kümesine sahip olmanın bazı sorunları çözdüğü anlaşılıyor.
  - Ancak tüm pratik ortamlarda daha büyük bir eğitim verisi kümesine sahip olduğumuzu varsaymak pratik değildir.
  - Bir yaklaşım, veri modellerimizi sık sık yeniden eğitmektir.
  - Bir diğer yaklaşım ise farklı modeller kullanmaktır.
- Model karmaşıklığı gerçek dünyanın karmaşıklığını yansıtmalıdır.

Uygulamalı Örnek

- Örnek durumda, tahmini değerlerdeki büyük dalgalanmanın nedeni, eğitimden sonra elde ettiğimiz modeldeki çok büyük katsayılardır.
  - Daha iyi modeller daha küçük katsayılar içerir.
  - Öyleyse neden modeldeki çok büyük katsayıları cezalandırmıyoruz?

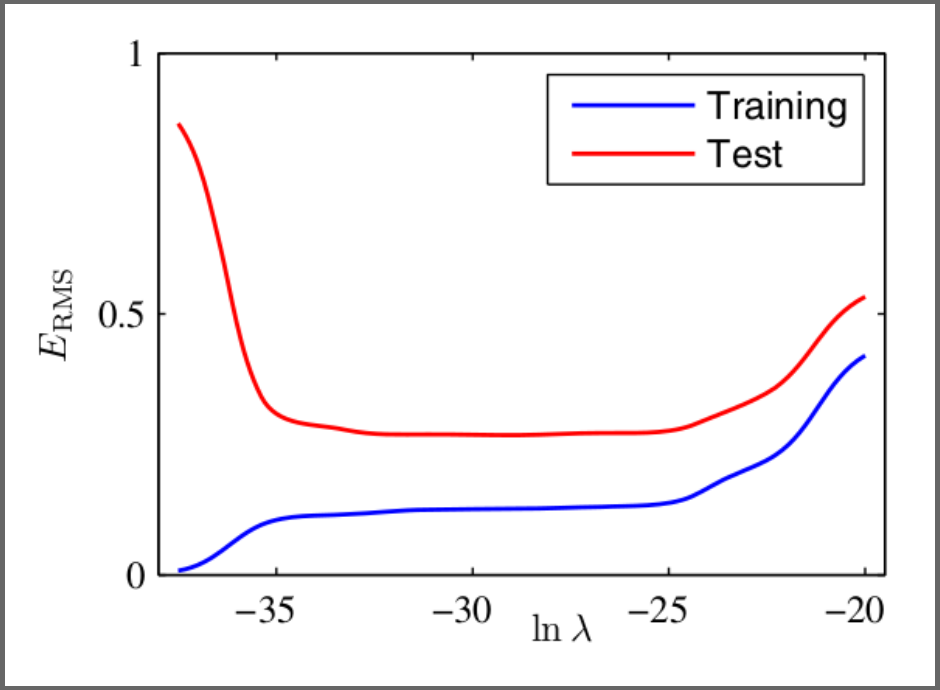
Uygulamalı Örnek

- Çok büyük katsayıları değiştirilmiş bir hata tanımıyla cezalandırıyoruz.
  - Bu değişiklik türüne büzülme (veya ağırlık azalması) denir ve bu özel olana sırt (ridge) regresyonu denir.

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \qquad (1.4)$$

Uygulamalı Örnek

- Model modifikasyonu seçimimiz, küçülmenin ne kadar önemli olduğunu simgeleyen
  - Lambda'yı optimize etmek için yeni bir parametre ekledi.
  - Bu parametreyi de seçmeliyiz.



Uygulamalı Örnek

- Aynı tahmin problemine bir başka yaklaşım Bayesçi (Bayesian) olacaktır.
  - Eğitim verilerimiz D=(t_1, t_2, ..., t_N) bir dizi gözlemi temsil etsin.
  - Parametre kümemizin w artık olasılık dağılımı p(w) olan olasılıksal olduğunu varsayalım.
  - Tüm nicelikleri w'nin bir fonksiyonu olarak tanımlamaya çalışırız.

Uygulamalı Örnek

- p(D|w) olasılık fonksiyonu olarak kullanıldığında:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} \tag{1.43}$$

$$p(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})\,\mathrm{d}\mathbf{w}. \tag{1.45}$$

Uygulamalı Örnek

- Frekansçı düşünce okulunda, olasılık fonksiyonunu p(D|w) maksimize etmek için genellikle maksimum olasılık tahmincisi (max. likelihood estimator) kullanırız.
  - Hata, olasılık fonksiyonunun negatif logaritması olarak tanımlanır. Biri arttıkça diğeri azalmalıdır.
  - Ancak temel fark, w'yi bir sabit olarak görmemiz ve verilen bir w için D'nin değişimini tartışmamızdır.

Uygulamalı Örnek

- Bayes düşünce okulunda, yalnızca D gözlemleri vardır ve D'yi gözlemledikten sonra w'deki belirsizliği değerlendiririz.
  - Bu nedenle D bir sabittir ve w'de tartıştığımız (ve optimize ettiğimiz) bir varyasyon vardır.
- Ayrıca olasılık fonksiyonunun bir olasılık dağılımı olmadığını da unutmayın.
  - Bayes yaklaşımı, frekansçı yaklaşımdan çok daha fazla hesaplama gerektirir ve genellikle elle hesaplanması mümkün değildir.
  - Ancak bilgisayar kaynakları el altında olduğunda daha popüler hale gelmiştir.

Uygulamalı Örnek

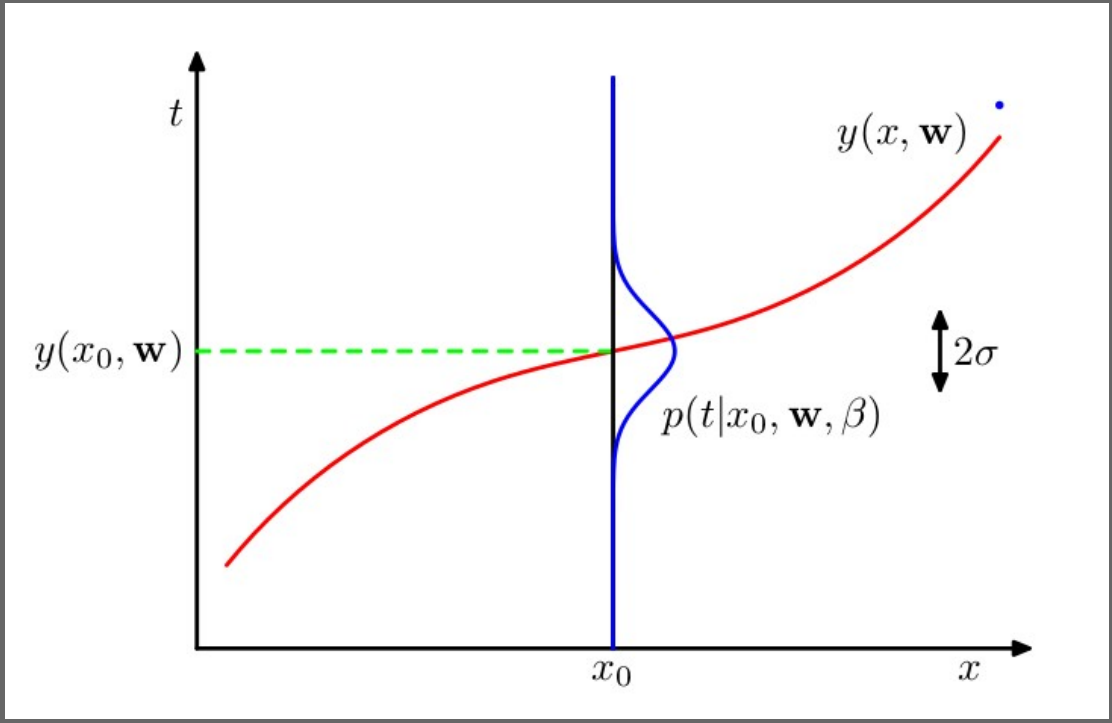
- Peki eğri uydurmaya Bayesçi bir bakış açısıyla nasıl yaklaşılır?
  - $t=(t_1,t_2, .., t_N)^T$ gözlemlerini seçili (bizim seçtiğimiz) bir olasılık yoğunluk fonksiyonu (pdf) ile bir miktar belirsizliğe sahip olarak ele alıyoruz.
    - Model seçimimiz artık PDF seçimimize dayanıyor.
  - Bu çok büyük bir sorun. Frekansçılar sıklıkla Bayesçi yaklaşımı eleştirerek "eğitim verilerinin gözlemlenmesinden sonra PDF seçimi hiledir" diyorlar.
  - Hata dağılımının Gauss olduğunu varsayalım (bu durumda bunun doğru olduğunu biliyoruz).

Uygulamalı Örnek

- Bu model seçimiyle herhangi bir gözlem t (sonuç) artık x'e (girdilere) dayalı, w parametreleri ve (pdf'in tanımlayıcı istatistiği olarak) beta'ya sahip koşullu bir olasılığa sahiptir.
  - Parametre değerlerini bulmak için eğitim veri setini ve maksimum olabilirlik tahmincisini kullanırız.
  - Maksimum olabilirlik fonksiyonunun negatif logaritması olarak tanımlanan hatayı en aza indirmek için, beta açısından maksimum olabilirlik fonksiyonunun logaritmasını en üst düzeye çıkarmaya çalışacağız.
  - Parametrelere sahip olduğumuzda, tahminler yapmaya başlayabiliriz.

Uygulamalı Örnek

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right) \tag{1.60}$$



Uygulamalı Örnek

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n | y(x_n, \mathbf{w}), \beta^{-1}\right). \tag{1.61}$$

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi). \tag{1.62}$$

Uygulamalı Örnek

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\}^2 . \qquad (1.63)$$

- w ve beta'yı belirledikten sonra tahminlerde bulunabiliriz.
  - Elimizde, t üzerinde bir olasılık dağılımı veren, nokta tahmini olmayan bir tahmini dağılım p(t | x, w, beta) var.
  - Ayrıca w'nin olasılıksal olduğunu ve w dağılımında hiperparametrelere sahip olduğumuzu varsayabiliriz. Örneğin p(w|alpha).

Uygulamalı Örnek

- w'nin olasılıksal olduğunu varsaydığımızda, gözlemler D verildiğinde w'nin en olası değerini tahmin ederiz.
  - Bu, w'yi bulmak için hata fonksiyonunu en aza indirmeye eşdeğerdir.
  - Bayes yaklaşımında hesaplama açısından pahalı olan integraller aldığımızı unutmayın.
- Maksimum olabilirlik yaklaşımında aşırı uyumun ve büyük bir eğitim veri kümesine sahip olmanın önemini gösterdik.
- Bayes yaklaşımında bunun ne kadar karmaşık olabileceğini ve görevlerimizin ne kadar hesaplama yoğun hale gelebileceğini gösterdik.

Küçük bir eğitim veri setiyle nasıl çalışılır?

- Bu önemli bir sorudur.
  - Büyük bir eğitim veri kümesi olduğunda, bazılarını eğitim için, bazılarını da doğrulama için seçeriz.
  - Küçük bir eğitim veri kümesi olduğunda, çapraz doğrulama kullanırız.
    - Bu yaklaşımda, her seferinde eğitim kümesinin farklı bir parçası doğrulama için ayrılmış birden fazla çalıştırmamız olur.
    - Bu çalıştırmaların performansından tanımlayıcı istatistikler (yani ortalamalar) geliştiririz ve performansı bu tanımlayıcılara göre değerlendiririz.

Küçük bir eğitim veri setiyle nasıl çalışılır?

- S katlı çapraz doğrulama
  - Eğitim veri kümesinin S (eşit büyüklükte) parçasını oluşturuyoruz.
  - Çoğunluğu (beyaz) kullanarak eğitiyoruz ve saklı (kırmızı) veri kümesiyle doğruluyoruz.
  - Performans değerlerinin ortalamasını alıyoruz.
- Aşırı durum, S=N çalışması yapmaktır, buna bir tane dışarıda bırakma da denir.

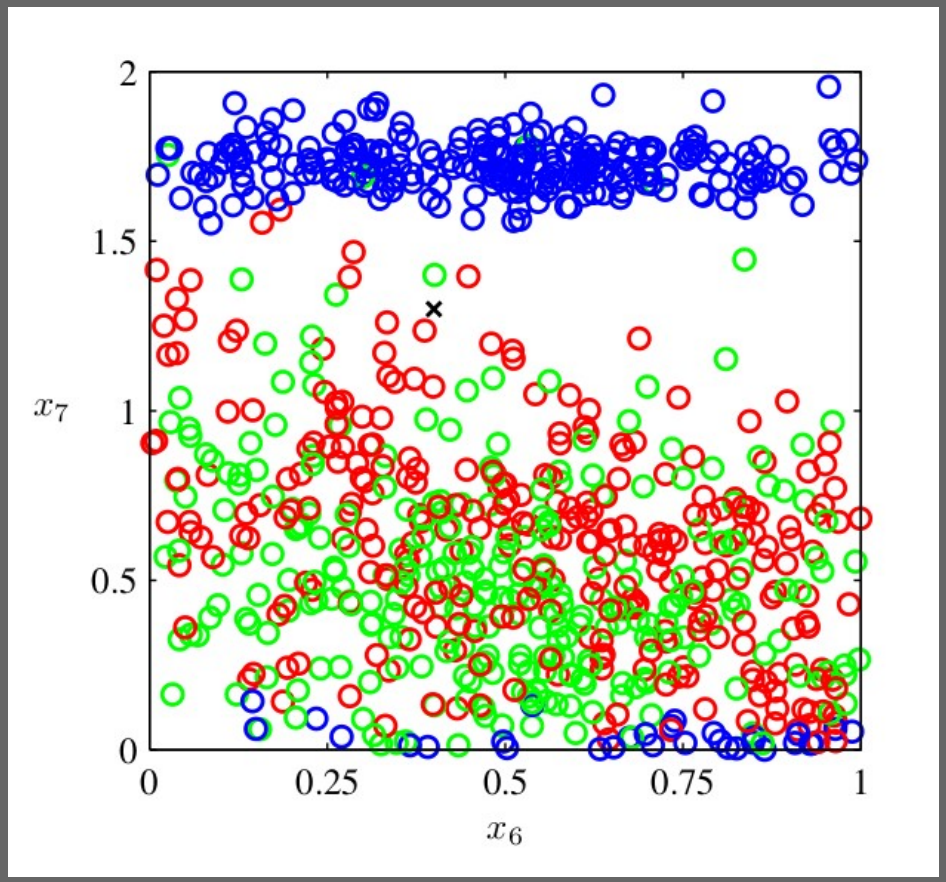
Boyutluluk

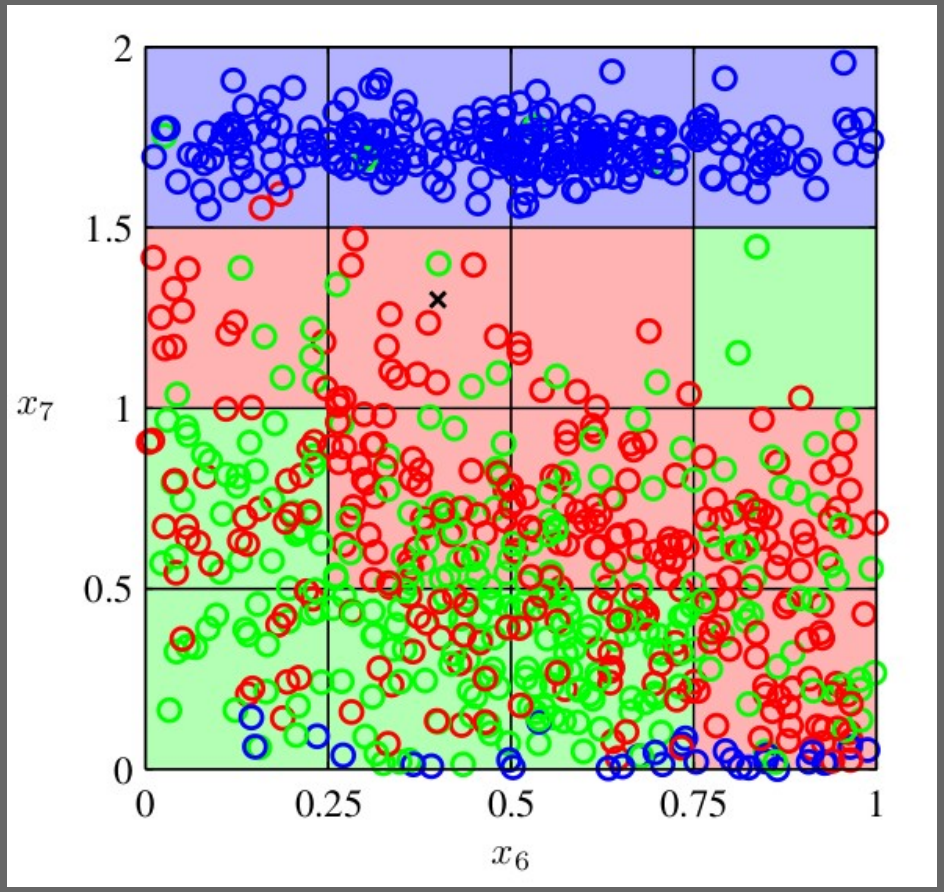
- Basit örneğimizde tek bir değişkenin fonksiyonunu tahmin etmek için bir modelin nasıl geliştirildiğini tartıştık.
  - Birden fazla değişkenimiz olduğunda (daha yüksek boyutluluk) bu çok daha karmaşık hale gelir.

Boyutluluk

- Bishop 2006 örneğini tekrar kullanacağız.
  - Örnek, bir sınıflandırma probleminde (12 tanesi içinden seçilen) iki değişkene odaklanmayı içerir.
  - Önerilen teknik, alanı test veri setindeki çoğunluk sınıfına ait olan "bölgelere" ayırır.
  - Bir nokta bir bölgeye düşerse, o zaman o sınıftan olarak sınıflandırılır.

Boyutluluk





Boyutluluk

- Bu yaklaşım 3 sınıf ve 2 boyutlu bir uzay için “eh işte” düzeyinde işe yarar.
  - Daha fazla sınıf ve tam 12 boyutlu uzay ne dersiniz?
    - Kaç bölgemiz var? 2^{12}? 3^{12}? 4^{12}?
    - Kaç katsayıya ihtiyacımız olacak? 12^2? 12^3? 12^4?
  - Saf yaklaşımlar daha yüksek boyutlarla işe yaramaz.
  - Buna boyutluluğun laneti denir (Bellman, 1961).

Karar Teorisi ve Makine Öğrenmesi

- Karar teorisi, karar almak için birçok yöntem önerdi. Genellikle şu amaçlarla bazı parametreleri optimize etmeye odaklanırlar:
  - Yanlış kararları en aza indirmek
  - Beklenen kayıpları en aza indirmek
  - Verilmesi zor kararları tespit etmek

Karar Teorisi ve Makine Öğrenmesi

- Ayrıca von Neumann'ın minimax teoremini ve Yao'nun rekabetçi bir ortamda çoklu tur karar alma ilkesini de göz önünde bulundurun.
  - Kısmen gözlemlenen ortamda (test veri seti) rakibin kararları hakkında sahip olduğumuz gözlemler D, parametreler (w) hakkında bilgi edinmek (yani bir aralık tahmin etmek) için kullanılabilir, dolayısıyla onun kullandığını varsaydığımız belirli bir karar alma modelinin sağladığı karar aralığı.
  - Model seçimi, olasılık dağılımları olarak parametreler, verilen gerçekler olarak gözlemler...
  - Bunlar Bayesçi yaklaşımın temel taşlarıdır.
  - Bayesçi Karar Teorisi, pratik uygulamalarda makine öğrenimini kullanır.

Model Değerlendirme Kriterleri

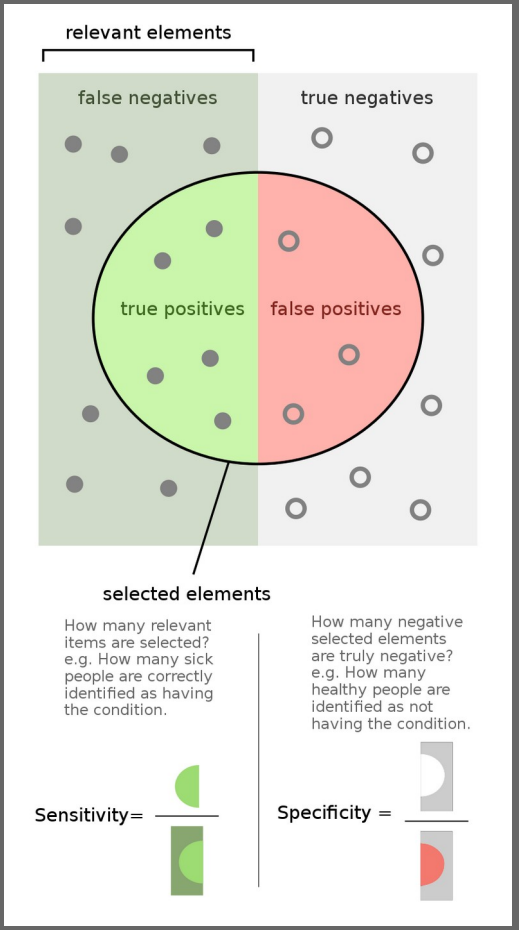
- Gördüğümüz gibi, genellikle modelimizi eğitim veri setiyle eğitmeden önce tasarlarız ve ardından doğrulama veri setiyle doğrularız.
  - Pratikte kullanmadan önce modelle ilgili sorunları gözlemleyebilmeliyiz.
- Bu nedenle bazı model değerlendirme kriterlerine ihtiyacımız var.

Model Değerlendirme Kriterleri

- Sınıflandırma problemlerinde birden fazla tipte problemimiz olacak.
  - Çekirdek performans ölçüsü olan sınıflandırma doğruluğu, doğru tahminlerin (gerçek pozitifler ve gerçek negatifler) yapılan toplam tahminlere oranıdır. Ancak birçok durumda, diğer tüm durumların (yanlış negatifler, yanlış pozitifler) oranı olarak hata oranı da önemlidir.
  - İki sınıf probleminde gerçek pozitif, yanlış pozitif, gerçek negatif, yanlış negatif yaklaşımını kolayca tanımlarız.

Model Değerlendirme Kriterleri

- Duyarlılık (geri çağırma) ve özgüllük kullandığımız iki önemli performans kriteridir. (Şekil Wikipedia'dan alınmıştır).
  - Yüksek duyarlılık her zaman iyidir.
  - Yüksek özgüllük, yüksek duyarlılıktan daha önemli olabilir çünkü birçok uygulamada yanlış pozitifler onarılması zor hasarlara neden olur.

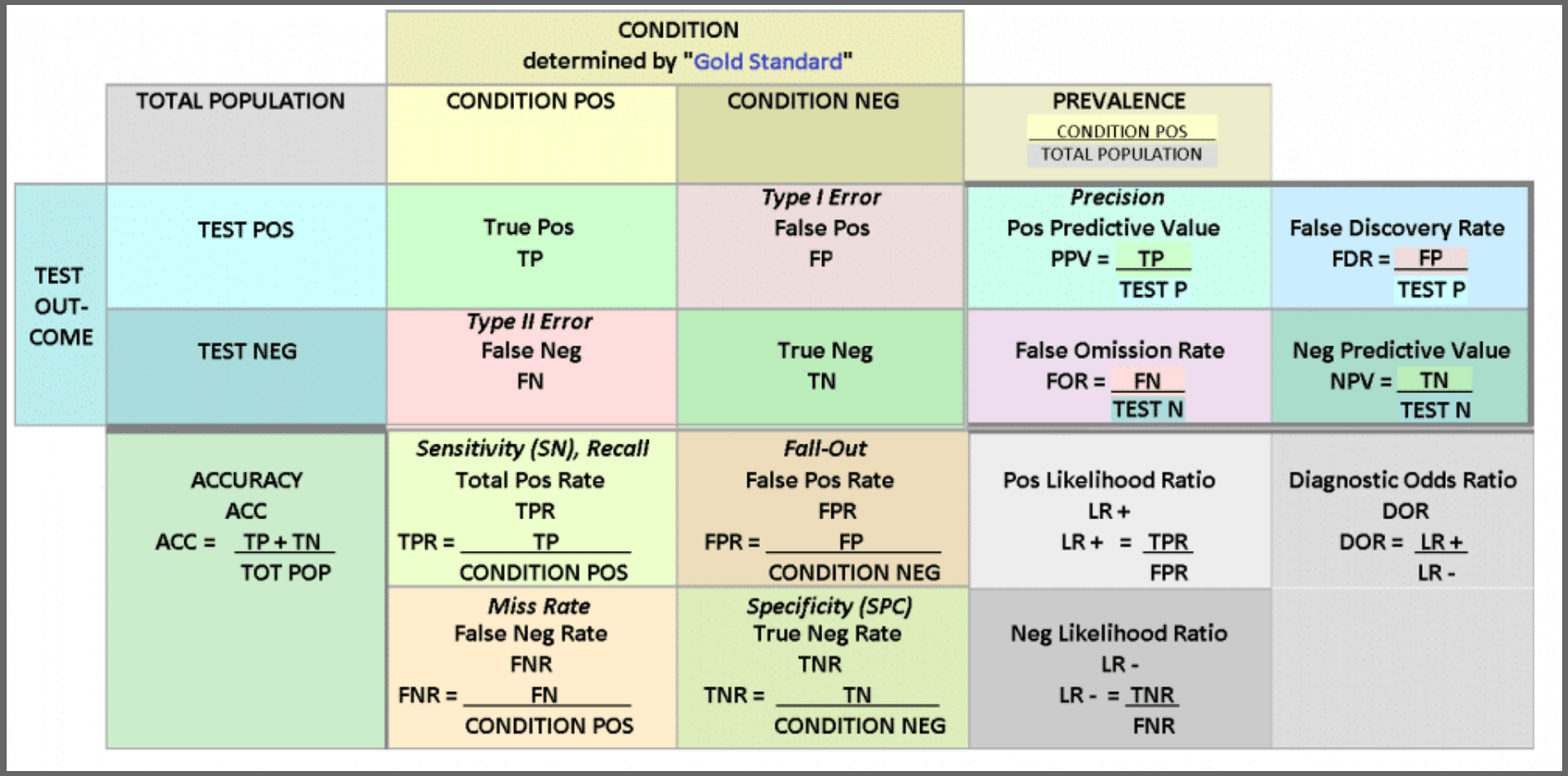


Model Değerlendirme Kriterleri

- Karışıklık matrisi, bir sınıflandırma algoritmasının performansını özetlemek için kullanılan bir tekniktir.
  - Gerçek ve tahmin edilen bir dizi sınıflandırmamız var ve bunların doğru/yanlış pozitif/negatif doğasını görüyoruz.
  - Bir matris kullanarak özetliyoruz.
  - Kesinlik, doğruluk ve geri çağırmayı hesaplıyoruz.

		Predicted	
		Negative (**N**) **-**	Positive (**P**) **+**
Actual	Negative **-**	True Negatives (T**N**)	False Positives (F**P**) **Type I error**
	Positive **+**	False Negatives (F**N**) **Type II error**	True Positives (T**P**)

Model Değerlendirme Kriterleri



Model Değerlendirme Kriterleri

- 2'den fazla sınıfımız olduğunda TP, TN vb. hesaplamak daha karmaşık hale gelir, ancak daha zor olmaz.

Model Değerlendirme Kriterleri

- R2 (belirleme katsayısı), bağımlı değişkendeki varyasyonun bağımsız değişken(ler)den tahmin edilebilir olan oranıdır.
  - Regresyon tabanlı tahmin modellerinin performansını değerlendirmek için kullanılır.
    - 1'lik bir R2, regresyon tahminlerinin verilere mükemmel şekilde uyduğunu gösterir.
  - Ancak, yüksek bir R2, bir ilişkinin işlevsel biçiminin yanlış belirlenmesi veya gerçek ilişkiyi bozan aykırı değerlerin varlığında ortaya çıkabilir.
- Birçok durumda SSE (kare hataların toplamı), R2'ye tercih edilir.

Ödev #2

- Karışıklık Matrisi hazırlama konulu.
- Detaylar Github’da duyurulacak.
- Teslimat 28 Mart

Ödev #3

- Bir sınıflandırıcının tasarımı ile ilgili.
  - İki kısımdan oluşacak. Birinci kısım rapor ve ön çalışma değerlendirmesi.
  - İkinci kısım veri analizi.
- Detaylar Github'da duyurulacak.
- Teslimat
  - Birinci Kısım: 4 Nisan
  - İkinci Kısım: 11 Nisan

Questions?

CONTACT:
bora.gungoren@atilim.edu.tr

License: Creative Commons Attribution Non-Commercial Share Alike 4.0 International (CC BY-NC-SA 4.0)