İKT484 Makine Öğrenmesi

3. Denetimli Öğrenme ve Regresyon

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

Denetimli Öğrenme

- Temel sınıflandırma probleminde 2 “sınıf” (kırmızı/mavi, hasta/sağlıklı, insan/insan değil, satın al/tut vb.) bulunur ve amacımız, erken örneklerin “gözetimi” yoluyla ayırt edici tanı koymayı öğrenmektir.
  - Ayrıca 2^N hakkında kabaca yaptığımız tartışmayı da hatırlayın.
  - Şimdi gözetimli öğrenme hakkında biraz daha teori öğrenme zamanı.

Denetimli Öğrenme

- Diyelim ki belirli bir sınıfın özelliklerini tanımlamak istiyoruz; örneğin, “önceden (2021’de) kâr etmiş ve gelecek yıl (2022’de) da kâr edecek bir şirket.”
  - 2020 ve 2021 yıllarında kâr eden birçok şirketin bilançosuna bakıyoruz. Ayrıca 2020 yılında kâr etmiş ama 2021 yılında kâr etmemiş şirketlerin bilançosuna da bakıyoruz. Yani onları etiketliyoruz.
  - Daha sonra her iki grup için de 2020 yılına ait birçok finansal oran hesaplıyoruz. Bunlar, öğrenmeyi temellendirdiğimiz özelliklerdir.
  - Öğrenme, **etiketler üzerinden** gerçekleşir.

Denetimli Öğrenme

- Öğrenme süreciyle, karar vermede kullanılan özellikleri ve bu kararı vermek için gerekli olan bazı aritmetik ve/veya mantıksal kuralları belirleriz.
  - Bu özellikler ve kurallar bütününe “tanıyıcı” (recognizer) diyebiliriz.
  - Şirketler girdilerdir, ancak özellikler bu girdilerin temsilidir (input representation).
  - Öğrenme sonrasında oluşturduğumuz ve yeni bir girdiyi etiketlemek için kullanacağımız kural kümesine hipotez sınıfı (örneğin, x1 > a) denir.
  - Yeni bir girdiyi belirli bir sınıfa atayan (yani etiketleyen) belirli bir hipotez ise bu sınıfa ait tekil bir hipotezdir (örneğin, x1 > 0.5).

Denetimli Öğrenme

- Denetimli öğrenmede öğrenebileceğimiz sonlu sayıda örneğimiz vardır.
  - Bu sonlu örneklerden yola çıkarak belirli bir hipotez oluştururuz ve bu hipotezi, cevabını (yani etiketini) bildiğimiz örneklerle test ederiz.
  - Ampirik hata, yapılan hata sayısının toplam örnek sayısına oranıdır. Bu hataların hata olduğunu biliriz, çünkü doğru cevapları zaten biliyoruz.
  - Dolayısıyla en iyi uyan hipotezi bulmak, hipotez sınıfındaki tüm olası hipotezleri denemek ve ampirik hatası en düşük olanı bulmak anlamına gelir.

Denetimli Öğrenme

- Denetimli öğrenmede, öğrenebileceğimiz sonlu sayıda örnek vardır.
  - Bu nedenle, belirli bir hipotez sınıfından kaç farklı hipotezin türeyebileceği (sınıfın kapsama alanı) önemli bir konu haline gelir.
  - Bir diğer önemli konu ise her hipotezin genelliğidir. Örneğin x1 > a şeklindeki bir kural için, hem x1 > 0.5 hem de x1 > 0.6 aynı tahminleri yapıyor ve aynı ampirik hatayı veriyorsa (elimizdeki mevcut örneklerle), o zaman daha az genel (daha sıkı) olan hipotez daha mı iyidir, değil midir?
  - Bu soru, genellikle modelin aşırı öğrenme (overfitting) yapıp yapmadığını ya da yeterince genelleyip genelleyemediğini anlamada önemlidir.

Denetimli Öğrenme

- Denetimli öğrenmede, öğrenmek için elimizde sonlu sayıda örnek vardır.
  - Hipotezin ne kadar sıkı (dar tanımlı) olduğunun önemli olmadığını, yalnızca sonuçların önemli olduğunu varsayarsak:
  - 2 sınıf ve N örnekle, 2^n farklı sonuç elde edebiliriz; dolayısıyla hipotez sınıfında 2^n adet hipotez olabilir.
    - Sadece merak açısından bakarsak, 2^{10} yaklaşık 1.000 eder, 2^{30} yaklaşık 1 milyar eder, 2^{350} ise çok büyüktür (bir googol'dan bile büyük).
    - Dolayısıyla, örnek sayısı arttıkça tüm hipotezleri değerlendirmek ve en iyisini bulmak çok daha fazla zaman alır.

Denetimli Öğrenme

- 2^n sonucu, Vapnik-Chervonenkis (VC) boyutu olarak bilinir.
  - VC terminolojisine göre, farklı sonuçlar üreten herhangi bir hipotez, N noktayı (sınıflara ayırarak) parçalamış olur.
  - VC-boyutu bize şunu öğretir: Denetimli öğrenmede tüm hipotezleri tek tek denemek (exhaustive search) pratikte mümkün değildir.
  - Bu nedenle, zaman kısıtlı herhangi bir uygulamanın sonucu yalnızca yaklaşık olarak doğru (Approximately Correct – AC) olacaktır.

Denetimli Öğrenme

- Bilgisayar bilimi açısından bakıldığında, denetimli öğrenmede kullandığımız tüm yöntemler zaman kısıtlı arama algoritmalarıdır; amaçları, şimdiye kadarki en iyi (best-so-far) hipotezi, en düşük ampirik hatayla bulmaktır.
  - Ancak pratikte, test etmeden önce zaten bir hipoteze sahibiz.
  - İstatistiksel açıdan bakıldığında ise, bu hipotezi doğrulamak için gereken örnek sayısını tahmin edebiliriz.
  - Muhtemelen Yaklaşık Doğru Öğrenme (Probably Approximately Correct – PAC) yaklaşımı, bilinmeyen ama sabit bir dağılımdan gelen örnekler üzerine odaklanır ve ampirik hatanın, önceden belirlediğimiz bir sınır içinde kalmasını sağlayacak yeterli sayıda örneği bulmaya çalışır.
-

Denetimli Öğrenme

- PAC bize, belirli bir hata oranı için ne kadar “sıkı” bir hipotezin kabul edilebilir olduğunu söyler. Bu nedenle pratikte büyük öneme sahiptir.
  - PAC ile şu tür ifadeler kurabiliriz:
    - Eğer ..... sayıda örnek alırsak,
    - .... güven olasılığıyla tahminlerde bulunabiliriz ve
    - sınıflandırma hatası olasılığı en fazla ..... olur.
  - Gürültü (noise), işleri zorlaştırır; dolayısıyla veri gürültülü olduğunda, aynı sayıda örnekle elde edilen sınıflandırma hatası olasılığı daha yüksek olur.
  - Peki örnek sayısını ne kadar artırmalıyız?

Denetimli Öğrenme

- İkiden fazla sınıf olduğunda, problemin ne kadar büyüdüğünü anlamamız gerekir.
  - Sistematik bir yaklaşımla, K-sınıflı bir problemi K adet 2-sınıflı probleme dönüştürebiliriz.
  - Bu durumda VC-boyutu, doğrudan K×NK×N değil, K×2NK×2N olur. Bu daha hızlı büyüyen, daha karmaşık bir yapı anlamına gelir.
    - Örneğin, A, B, C sınıflarımız olsun:
      - İlk problem: A mı, yoksa değil mi?
      - İkinci problem: A değilse, B mi yoksa değil mi?
      - Üçüncü problem: A değilse ve B de değilse, C mi yoksa değil mi? (Yani aykırı mı?)

Denetimli Öğrenme

- Eğer paralel çalışabilecek imkânlara sahipsak, arama sürecini de paralelleştirebiliriz.
  - Ancak paralel arama her zaman aynı sonuçları garanti etmez. Özellikle dallar arasında **iletişim yoksa**, ortaya çıkan sonuçlar tutarsız veya çakışmalı olabilir.
  - **İletişim olmadan paralel arama:**
    - **Dal 1**: A mı, değil mi?
    - **Dal 1 (devam)**: A değilse, B mi yoksa değil mi?
    - **Dal 2:** A değilse, C mi yoksa değil mi?
    - Bu iki dal sonunda birleşirken, hem B'leri hem de C'leri sınıflandırmaya çalışırız.
    - Ancak **aykırı değerlerin (outlier)** kim olduğunu belirlemek zorlaşır çünkü dallar arasında **bilgi paylaşımı yoktur.**

Denetimli Öğrenme

- Eğer paralel çalışabilecek imkânlara sahipsak, arama sürecini de paralelleştirebiliriz.
  - Ancak paralel arama her zaman aynı sonuçları garanti etmez. Özellikle dallar arasında **iletişim yoksa**, ortaya çıkan sonuçlar tutarsız veya çakışmalı olabilir.
  - **İletişimli paralel arama (koordinasyonlu):**
    - **Dal 1:** A mı, değil mi?
    - **Dal 1 (devam)**: A değilse ve **Dal 2** tarafından C olarak işaretlenmediyse, B mi, değil mi? **(Finders keepers: önce bulan sınıflandırır)**
    - **Dal 2:** A değilse ve **Dal 1** tarafından B olarak işaretlenmediyse, C mi, değil mi?
- Sonuçta, her iki dal da birbirinin işaretlemelerini göz önüne alarak çalışır. Bu yaklaşım, sınıflandırma çakışmalarını azaltır ve daha sağlam sonuçlar elde edilmesini sağlar.
- Yine de, dallar birleştiğinde **aykırıları** (hiçbir sınıfa net olarak uymayanları) belirleyip doğru şekilde etiketlemek gerekir.
- Bu tür koordinasyon, özellikle büyük veri kümelerinde ve dağıtık sistemlerde sınıflandırma yapılırken önem kazanır.

Regresyon

- Regresyonda, hipotezimiz bir sınıflandırma hipotezi değildir; çünkü burada Boole (doğru/yanlış) ya da kategorik bir sonuç yoktur.
  - Tahmin etmemiz gereken şey **sürekli bir fonksiyondur**.
  - Fonksiyonu tam olarak bilmeyiz, ancak **biçimini (formunu)** varsayarız — bu, hipotez sınıfımızı oluşturur.
  - Bu biçim üzerinden **parametreleri tahmin ederek** belirli bir fonksiyona (yani tekil bir hipoteze) ulaşırız.
  - Bu parametreleri tahmin ederken, artık ampirik hatayı farklı bir şekilde (örneğin, **ortalama kare hata – MSE** gibi) hesaplarız ve bu hatayı **minimize etmeye çalışırız**.

Regresyon

- Ancak gürültü (noise) ve zaman kısıtları nedeniyle ulaştığımız sonuçlar yine **PAC (Probably Approximately Correct)** çerçevesine girer:
  - Yani belirli bir güven aralığıyla, yaklaşık doğru sonuçlar elde ederiz.
- Regresyonda **gürültüyü**, genellikle varsaydığımız fonksiyon biçiminde yer almayan **eksik değişkenler** olarak açıklarız.
  - Bu yüzden modelin açıklayamadığı sapmalar "gürültü" olarak kabul edilir.

Model Seçimi ve Genelleştirme

- Sınıflandırma problemlerinde, her yeni örnek, hipotez sınıfımızdaki olası hipotezlerin sayısını yarıya indirir (veya önemli ölçüde azaltır).
  - Bu nedenle, belirli bir noktadan sonra (yani gereken örnek sayısına ulaşıldığında), sınıflandırma istenildiği şekilde çalışmaya başlar.
  - Ancak bu noktadan sonra daha fazla örnek eklemek, hipotezi gittikçe sıkılaştırır.
  - Bu durum genellikle aşırı öğrenmeye (overfitting) yol açar — yani model, verideki gerçek yapıyı değil, veri üzerindeki rastlantısal desenleri öğrenmeye başlar.

Model Seçimi ve Genelleştirme

- Ancak, çok sayıda örneğe sahip olmak bize şu rahatlığı da sağlar:
  - Gereken güven aralığına sahip birden fazla hipotez adayı oluşturabiliriz.
  - Yani eğer ihtiyacımız olan örnek sayısı N ise, ama elimizde N + K örnek varsa, o zaman elimizde yaklaşık 2^K kadar geçerli alternatif hipotez olabilir.
- Peki bu alternatif hipotezlerden hangisini seçmeliyiz?
  - Yanıt basit: Test ederek.

Model Seçimi ve Genelleştirme

- Test etme süreci, **öğrenmek için yeterli sayının üzerinde örneğe sahip olduğunuzda** en iyi şekilde çalışır.
  - Bu durumda, istenen hata oranlarını sağlayan **birden fazla alternatif model** eğitirsiniz (eğitim verisi üzerinde).
  - Ardından bu modelleri **test veri kümesinde** test edersiniz.
  - Ve en iyisini seçersiniz.
- **S-katlı çapraz doğrulama (S-fold cross validation)**, yeterince büyük bir veri kümesi üzerinde çok daha fazla alternatif model ve çok daha fazla test veri kümesi üretir. Bu yöntem, özellikle az örnekle bile güvenilir seçimler yapmak için güçlüdür. Ancak, hangi model sınıfını (kural biçimini) seçtiğiniz de sonuçları etkiler.
- Hipotez sınıfının getirdiği bu tür bir sapma, yani *kuralların biçiminden kaynaklı hata**, **tümevarımcı önyargı (inductive bias)** olarak adlandırılır. Çünkü bu önyargı, veriden değil, model tasarımcısının seçtiği **model yapısından (formdan)** gelir — yani modelleyenin “indüksiyonu” sonucudur.

Model Seçimi ve Genelleştirme

- Bir hipotez sınıfının daha fazla hipotez (veya fonksiyon) içerebilmesi için, **daha karmaşık bir forma** sahip olması gerekir — yani daha karmaşık kurallar ve/veya daha karmaşık fonksiyonlar.
  - Ancak, mümkün olduğunca **basit bir form** denemek, her zaman bir miktar **tümevarımcı önyargı** (inductive bias) getirecektir.
  - Bu yüzden **model seçimi**, bu önyargıyı tamamen ortadan kaldırma süreci değildir.
  - Aksine, **en az zarar veren önyargıyı seçme** sürecidir.

Model Seçimi ve Genelleştirme

- Amaç, modelin hem öğrenme sürecinde anlamlı sonuçlar üretmesini hem de gerçek hayattaki yeni örneklere iyi genelleme yapmasını sağlamaktır.
- Yani “doğruya en yakın ve mümkün olduğunca sade” bir hipotezi seçmektir.

Model Seçimi ve Genelleştirme

- Karmaşık bir problem için karmaşık bir hipotez gereklidir.
  - Eğer ihtiyacımızdan daha basit bir hipotez formu seçersek, bu durum genellikle underfitting ile sonuçlanır.
  - Çünkü gereğinden basit bir hipotez, problemi tam olarak ifade edemez ve daha fazla yanlış sonuç üretir.
  - Öte yandan, ihtiyacımızdan daha karmaşık bir hipotez formu seçersek, bu da genellikle overfitting ile sonuçlanır — model, verideki gerçek desenlerin yanı sıra rastlantısal gürültüyü de öğrenir.

Model Seçimi ve Genelleştirme

- Karmaşık bir problem için karmaşık bir hipotez gereklidir.
  - Regresyondaki gürültüyü eksik değişkenler olarak açıklamak, aslında şunu varsayar:
    - Modelimiz, olması gerekenden daha basit — çünkü bazı önemli değişkenleri hesaba katmıyoruz.
    - Bu da demektir ki, bu açıklama doğası gereği sürekli olarak underfitting yaptığımızı varsayar.
    - Eksik değişken → Eksik bilgi → Basitleştirilmiş model → Underfitting
    - Bu nedenle, gürültüyü azaltmak ya da daha iyi tahmin yapmak için **ya hipotez formunu zenginleştirmemiz ya da modele katkı sağlayacak eksik değişkenleri bulmamız** gerekir.

Model Seçimi ve Genelleştirme

- Denetimli öğrenmede **üç yönlü bir denge (three-way trade-off)** vardır:
  - **Hipotezin karmaşıklığı** – Veriye uyum sağlayacak kadar güçlü bir model gerekir. Daha karmaşık bir hipotez sınıfı, daha fazla hipotezi kapsayabilir ve karmaşık desenleri öğrenebilir.
  - **Eğitim verisinin miktarı** – Yeterli sayıda ve temsili örnek olmadan, karmaşık modeller aşırı öğrenmeye (overfitting), basit modeller ise yetersiz öğrenmeye (underfitting) meyillidir.
  - **Genelleme hatası** – Yeni örneklerde yapılan hatalardır. Aşırı öğrenme, eğitim verisinde düşük hata ama yeni veride yüksek hata üretir. Yetersiz öğrenme ise hem eğitimde hem testte yüksek hata üretir.

Model Seçimi ve Genelleştirme

- Bu üç unsur arasında şu tür ilişkiler vardır:
  - Karmaşık model + az veri → overfitting
  - Basit model + karmaşık veri → underfitting
  - Yeterli veri + uygun karmaşıklık → düşük genelleme hatası
- Amaç, **doğru miktarda veriyle, uygun karmaşıklıkta bir model** seçerek genelleme hatasını minimize etmektir.
- Bu dengeyi kurmak, denetimli öğrenmenin en temel zorluklarından biridir.

Model Seçimi ve Genelleştirme

- Karmaşıklık arttıkça, **genelleme hatası** (generalization error) önce azalır, sonra artar.
  - Başlangıçta daha karmaşık modeller, veriye daha iyi uyum sağlar ve hatayı azaltır.
  - Ancak belli bir noktadan sonra model, verideki **gürültüyü** de öğrenmeye başlar — yani **aşırı öğrenme (overfitting)** ortaya çıkar ve genelleme hatası yeniden yükselir.

Model Seçimi ve Genelleştirme

- Genelleme hatasını ölçebildiğimiz için, bu yeteneğimizi avantajımıza kullanırız:
  - **Doğrulama kümesi (validation set)** kullanırız (veya ayrı bir test verisi).
  - Her modelin doğrulama kümesindeki hatasını ölçeriz.
  - **Hata azaldıkça, modelin önyargısı (bias)** da daha düşük kabul edilir.
    - Bu hata, aynı zamanda **kayıp fonksiyonu (loss function)** olarak da adlandırılır.
- Amaç, bu fonksiyonu minimize etmektir.

Model Seçimi ve Genelleştirme

- Tam bu noktada önemli bir **varsayımda** bulunuruz:
  - Tüm örneklerin **bağımsız ve özdeş dağılımlı (IID – independent and identically distributed)** olduğunu kabul ederiz.
  - Bu dağılımın tam olarak ne olduğunu bilmesek de, tüm örneklerin aynı bilinmeyen dağılımdan geldiğini varsaymak, hem teorik analiz hem de pratik modelleme için temel bir çerçeve sağlar.

Sorular?

İLETİŞİM:
bora.gungoren@atilim.edu.tr