K-En Yakın Komşu Algoritması Raporu

Hazırlayan: Mehmet Emir ERDEM

Öğrenci No: 20360859091

Bölümü: Bilgisayar Mühendisliği 3. Sınıf

K-En Yakın Komşu Algoritması Tarihçesi

K-en yakın komşu (KNN) sınıflandırması, en temel ve basit sınıflandırma yöntemlerinden biridir ve verilerin dağılımı hakkında çok az veya hiç ön bilgi olmadığında bir sınıflandırma çalışması için ilk seçeneklerden biri olmalıdır. K-en yakın komşu sınıflandırması, olasılık yoğunluklarının güvenilir parametrik tahminlerinin bilinmediği veya belirlenmesinin zor olduğu durumlarda diskriminant analizi gerçekleştirme ihtiyacından geliştirilmiştir. 1951'de yayınlanmamış bir ABD Hava Kuvvetleri Havacılık Tıbbı Okulu raporunda, Fix ve Hodges, o zamandan beri k-en yakın komşu kuralı olarak bilinen model sınıflandırması için parametrik olmayan bir yöntem tanıttı (Fix & Hodges, 1951). Daha sonra 1967'de, k-en yakın komşu kuralının bazı biçimsel özellikleri üzerinde çalışıldı; örneğin k=1 ve $n \rightarrow \infty$ için k-en yakın-komşu sınıflandırma hatasının Bayes hata oranının iki katı ile sınırlandırıldığı gösterilmiştir (Cover & Hart, 1967). K-en yakın-komşu sınıflandırmasının bu tür biçimsel özellikleri belirlendikten sonra, yeni reddetme yaklaşımlarını (Hellman, 1970), Bayes hata oranına ilişkin iyileştirmeleri (Fukunaga & Hostetler, 1975), mesafe ağırlıklı yaklaşımları (Dudani, 1976; Bailey ve Jain, 1978), yumuşak hesaplama (Bermejo ve Cabestany, 2000) yöntemleri ve bulanık yöntemler (Jozwik, 1983; Keller ve diğerleri, 1985) içeren uzun bir araştırma dizisi ortaya çıktı.

K-En Yakın Komşu Algoritması

KNN veya k-NN olarak da bilinen k-en yakın komşular algoritması, bireysel bir veri noktasının gruplandırılması hakkında sınıflandırmalar veya tahminler yapmak için yakınlığı kullanan, parametrik olmayan, denetimli bir öğrenme sınıflandırıcıdır. Regresyon veya sınıflandırma problemleri için kullanılabilirken, tipik olarak benzer noktaların birbirine yakın bulunabileceği varsayımından yola çıkarak bir sınıflandırma algoritması olarak kullanılır.

Sınıflandırma problemleri için, çoğunluk oyu temelinde bir sınıf etiketi atanır. Belirli bir veri noktası çevresinde en sık temsil edilen etiket kullanılır. Bu teknik olarak "çoğunluk oyu" olarak kabul edilirken, "çoğunluk oyu" terimi literatürde daha yaygın olarak kullanılmaktadır. Bu terminolojiler arasındaki fark, "çoğunluk oylamasının" teknik olarak %50'den fazla bir çoğunluk gerektirmesidir, bu da öncelikle yalnızca iki kategori olduğunda işe yarar. Birden fazla sınıf olduğunda, örneğin dört kategori, bir sınıf hakkında bir sonuca varmak için mutlaka %50 oy alınması gerekmez; %25'ten fazla oy alan bir sınıf etiketi atanabilir.

Regresyon problemleri, sınıflandırma problemi ile benzer bir kavram kullanır, ancak bu durumda, bir sınıflandırma hakkında tahminde bulunmak için en yakın k komşunun ortalaması alınır. Buradaki temel ayrım, sınıflandırmanın ayrık değerler için kullanılması, regresyonun ise sürekli olanlarla kullanılmasıdır. Ancak bir sınıflandırma yapılmadan önce mesafenin tanımlanması gerekir.

Uzaklık Hesaplama Yöntemleri

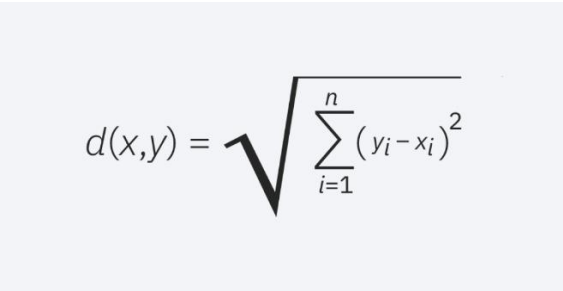
- Euclidean Uzaklığı
- Manhattan Uzaklığı
- Minkowski Uzaklığı
- Hamming Uzaklığı
- Jacckard İndeksi Uzaklığı
- Mahalonobis Uzaklığı
- Haversiene Uzaklığı
- Sorensen-Dice Uzaklığı

Genellikle Euclidean, Manhattan, Minkowski uzaklıkları kullanılır.

1. Euclidean Uzaklığı

Öklid uzaklığı, çok boyutlu kartezyen veri uzayında iki nokta arasındaki uzaklığın doğrusal bağlantı yöntemi ile ölçülmesidir.

KNN algoritmasında en çok kullanılan uzaklık ölçüsüdür. Genel olarak denklemi aşağıdaki gibidir:

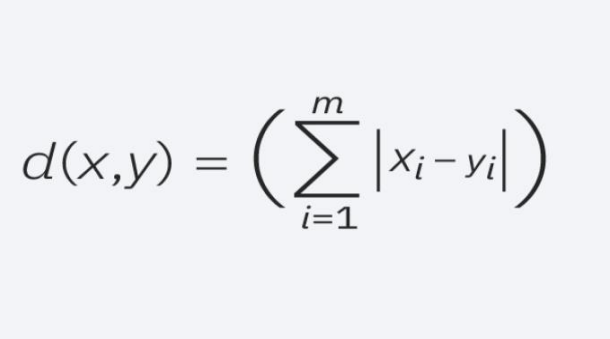


$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

2. Manhattan Uzaklığı

Manhattan uzaklığı, n boyutlu iki nokta arasındaki farkların mutlak değerlerinin toplamıdır.

Genel olarak denklemi aşağıdaki gibidir:

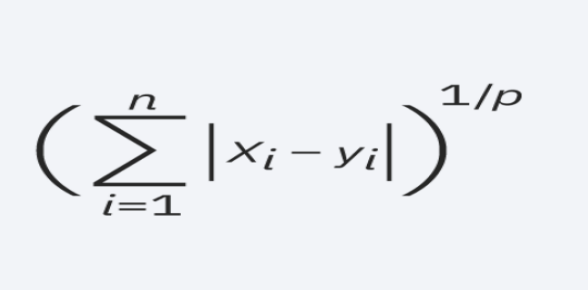


$$d(x,y) = \left(\sum_{i=1}^{m}|x_i - y_i|\right)$$

3. Minkowski Uzaklığı

Q sayıda bir değişkene bağlı bir uzaklık hesaplamak isteniyorsa Minkowski yöntemi kullanılır.

Eğer Q değişkeni iki ise o zaman formül q 2 değişkeni için Öklid uzaklığı ile aynı olur. Genel olarak denklemi aşağıdaki gibidir:



$$\left(\sum_{i=1}^{n}|x_i - y_i|\right)^{1/p}$$

Lazy Learning vs Eaeger Learning

Eager Learning:

Bir model yoluyla öğrenmektir. Model bütün verilerden etkilenir. Bu sebeble yerel yoğunluklara tepkisi çok iyi değildir. Sürekli bir şekilde gelen veri ile değerlerini ufak ufak da olsa değiştirir ve bunu sürekli bir biçimde yapar. Bu tür yaklaşımı kullanan Makine Öğrenmesi Algoritmasına ise Decision Tree (Karar Ağacı yapısıdır). Bu öğrenme Yönteminde modeli değiştirmek oldukça zordur ancak Algoritmanın istenen inputa cevap süresi ise kısadır.

Lazy Learning:

Burada model söz konusu değildir. Verinin hepsi kullanılmayabilir. Yerel yoğunlaşmalara çok daha iyi tepki verebilir. Verinin her zaman hızlı bir şekilde çağırılabileceği bir yerde tutulması gerekmektedir. İstenen değere yakın olanların etkisi uzak olanlara göre daha azdır (Bu Eager Learning'de genellikle eşittir.). Bu tür yaklaşımı kullanan Makine Öğrenmesi Algoritmasını örnek Kth Nearest Neighbor (KNN) algoritmasıdır. Bu öğrenme yönteminde çıktıya sonuç verilmesi Eager Learninge göre yavaştır. Verinin bir kısmından faydalanması belli açılardan çıkarımın kalitesini arttırabilir ancak kötü tercihler ile yanlış sonuçlara ulaşmanıza da yol açabilir.

Parametrik Algoritmalar

Parametrik algoritmalar, girdiler ve çıktılar arasındaki ilişkiyi tanımlayan matematiksel bir modele dayanır. Bu onları parametrik olmayan algoritmalardan daha kısıtlayıcı yapar, ancak aynı zamanda onları daha hızlı ve daha kolay eğitilebilir hale getirir. Parametrik algoritmalar, girdi verilerinin iyi tanımlanmış ve tahmin edilebilir olduğu problemler için en uygundur.

Non-Parametrik Algoritmalar

Parametrik olmayan algoritmalar matematiksel bir modele dayalı değildir; bunun yerine, verilerin kendisinden öğrenirler. Bu onları parametrik algoritmalardan daha esnek yapar, ancak aynı zamanda hesaplama açısından daha pahalıdır. Parametrik olmayan algoritmalar, girdi verilerinin iyi tanımlanmadığı veya parametrik bir algoritma kullanılarak modellenemeyecek kadar karmaşık olduğu problemler için en uygundur.

KNN Avantajları

- Basit algoritma olması sebebiyle yorumlaması, uygulaması kolaydır.
- Mesafe metriği istenildiği gibi seçilebilir.
- Regresyon ve sınıflandırma için kullanışlıdır.
    - Doğrusal olmayan karar sınırlarını öğrenebilir.
- Eğitim (Training) süresi kısadır.
    - Tüm iş tahmin sırasında gerçekleşir.
- Sürekli olarak gelişir.
    - Eğitim adımı olmadığından, veri kümesine her yeni veri eklediğimizde gelişmiş olur.
- Veriler üzerinde varsayımda bulunmaz (non-parametric).
- Yüksek doğruluk (accuracy) ile sonuca varır.

KNN Dezavantajları

- Non-parametric olması sebebiyle her tahminde bütün eğitim verilerinin kullanılması gerekir.
- Tembel (lazy) öğrenme algoritmasıdır.
- Tüm veri seti eğitim seti olarak kullanılır.
- Non-parametrik olması sebebiyle overfitting duruma düşmeye daha yatkın olmasına rağmen, k hiperparametresinin büyük/küçük olmasına göre de overfitting/underfitting durumu gerçekleşebilir.
- Yanlış etiketlenmiş bir veri sınıf sınırlarını değiştirebilir.
- Veri kalitesine göre doğruluk (accuracy) değişebilir.
- Gürültülü veya kullanılması mümkün olmayan veriler.
- Büyük veri ile çalışmalarda tahmin yavaş olabilir. Hesaplama maliyeti yönünden pahalı olabilir.

- Veri sayısı artınca, tahmin uzun sürebilir.
- Tüm eğitim (training) verilerini saklamak için yüksek bellek gerekir.

KNN Kullanım Alanları

- Tavsiye Sistemleri için KNN kullanılabilir. KNN, yüksek boyutlu veriler için uygun değildir, ancak KNN, sistemler için mükemmel bir temel yaklaşımdır. Netflix, Amazon, YouTube ve daha pek çok şirket, tüketicileri için kişiselleştirilmiş önerilerde bulunur.
- KNN, anlamsal olarak benzer belgeleri arayabilir. Her belge bir vektör olarak kabul edilir. Belgelerin birbirine yakın olması, belgelerin aynı konuları içerdiği anlamına gelir.
- KNN, aykırı değerlerin tespitinde etkin bir şekilde kullanılabilir. Örnek olarak, Kredi Kartı dolandırıcılık tespiti verilebilir.

Kaynakça

- https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-23832490e3f4
- https://towardsdatascience.com/parametric-vs-nonparametric-machine-learning-algorithms-5bf31393d944
- https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.
- https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
- https://aanilkayy.medium.com/eager-learning-ve-lazy-learning-nedir-d133f81e1a3d
- https://www.geeksforgeeks.org/difference-between-parametric-and-non-parametric-methods/
- https://www.codecademy.com/learn/introduction-to-supervised-learning-skill-path/modules/k-nearest-neighbors-skill-path/cheatsheet
- https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/