Meta Öğrenme ile KNN Parametre Seçimi
KNN Parameter Selection Via Meta Learning

Zeynep Banu Özger
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
zozger@yildiz.edu.tr

Mehmet Fatih Amasyalı
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
mfatih@ce.yildiz.edu.tr

***Özetçe*—- Bu çalışmada, meta öğrenme teknikleri ile K Nearest Neighbor (KNN) için en iyi hiper parametre değeri tahmin edilmeye çalışılmıştır. Uygulamada bir veri seti çeşitli meta özelliklerine göre değerlendirilerek, k parametresi seçilir. Eğitim kümesini oluşturmak için 200 adet veri seti kullanılarak her birinin meta özellikleri çıkarılmıştır. Her bir veri setine en yaygın kullanılan 6 farklı k değeri için KNN algoritması uygulanmış ve en yüksek başarının elde edildiği k değeri seçilmiştir. Böylece elde edilen yeni eğitim kümesi ile yeni gelen bir veri seti KNN ile sınıflandırılmak istendiğinde hangi k değerinin seçileceği tahmin edilmektedir. Deneysel çalışmalar neticesinde en yoğun kullanılan k değeri 1 çıkmıştır. Geliştirilen modelin doğruluğu 4 farklı yöntem ile denemiş ve hepsinde de en belirleyici özelliğin veri seti PART Rules ile modellendiğinde oluşan kural sayısının örnek sayısına oranı çıkmıştır. Seçilen meta özelliklerin bazılarının hiçbir yöntemde de kullanılmadığı gözlemlenmiştir.**

Anahtar Kelimeler — Meta öğrenme; k-nn; k-nn hiperparametreleri.

***Abstract*—- In this study, the K Nearest Neighbor's (KNN) parameter k is predicted by system. Meta learning method is used for prediction. Getting training set with meta features, 200 data sets were used. For each of them, 16 meta fetures were extracted. The KNN algorithm was applied each of them with most common 6 k values the best one is selected. With this training set it is possible to predict a new data set's best k value. In 200 dat sets the most common k value which has best performance is 1. 4 method is applied on the model. Generally all methods used same features and some meta features are never used.**

Keywords — Meta Learning; k-nn; k-nn hyper parameters.

I. GİRİŞ

Günümüzde sınıflandırma, kümeleme veya regresyon problemlerinde yaygın olarak kullanılmakta olan bir çok Makine Öğrenmesi tekniği bulunmaktadır. Mevcut teknikler her veri kümesi için her zaman başarılı sonuçlar alamamaktadır. Aynı şekilde bir öğrenme algoritmasının başarısı, veri setinin hiper parametre değerlerine göre de değişiklik gösterebilmektedir. Bir veri setinin hangi öğrenme algoritması ile veya bir öğrenme algoritmasının hangi parametresi ile daha başarılı sonuç alacağını öğrenebilmek için çeşitli yöntem veya parametrelerin tek tek denenmesi zaman almaktadır. Bu nedenle, bir veri setinin karakteristiğine bakılarak, kullanılacak öğrenme algoritmasının veya bir algoritmanın hiper parametresinin belirlenmesine dayalı meta öğrenme teknikleri geliştirilmektedir. [1]

Meta öğrenme teknikleri sınıflandırma, regresyon, optimizasyon gibi bir çok alanda kullanılmaktadır. Kate A. Smith-Miles [2] de algoritma veya parametre seçimine yönelik, meta öğrenme yöntemleri kullanılarak yapılan çalışmalardan bahsetmişlerdir.

Tembel öğrenici olarak bilinen K en yakın komşuluk algoritması bir çok sınıflandırma probleminde basit ancak etkili bir çözüm sunmaktadır. Algoritmanın temel mantığı sınıflandırılacak verinin en yakınındaki k adet örneğin sınıf bilgisine bakarak yeni veriyi çoğunluğun ait olduğu sınıfa atamaktır. [3][4]

Algoritmanın sınıflandırma işlemindeki başarısını etkileyen faktörlerden biri k değerinin doğru seçilmiş olmasıdır. Eğer k değeri çok büyük seçilirse farklı sınıflara ait örnekler aynı sınıfa dahil edilebilir. K değerinin çok küçük seçildiği diğer bir durumda ise tam tersi şekilde aynı sınıfta bulunması gereken örnekler farklı sınıflara yerleştirilebilir.[4]

K parametresini seçmek için kullanılan bir çok yöntem geliştirilmiştir. Ancak bunların en yaygını veri setini çeşitli k değerleri ile deneyerek en başarılı olan değerin seçilmesidir. Bu işlem ise zaman kaybına neden olmaktadır.

Uygun k değerini seçmek için geliştirilen bir yöntem uzaklıkların ağırlıklandırılmasına dayanmaktadır. K-Nearest Neighbor with Distance Weighted (KNNDW) yöntemi en başarılı k değerini seçerken test verisinin en yakınındaki k örneği, uzaklıklarına göre ağırlıklandırarak başarıyı hesaplar. Xie et al.[5] k değerini doğru belirleyerek KNN başarısını artırmak için Selective Neighborhood Naive Bayes (SNNB) algoritmasını geliştirmiştir. Çalışmada bir test örneği için faklı k değerleri kullanılarak çoklu Naive Bayes sınıflandırıcısı oluşturulur ve ardından en başarılı test örneği seçilerek en başarılı k değeri bulunur. Locally Weighted Naive Bayes(LWNB)'de [6] en yakındaki her bir komşu test verisine uzaklığına göre ağırlıklandırılarak oluşturulan ağırlıklandırılmış eğitim setine lokal Naive Bayes uygulanır. En iyi k değerini öğrenmek için kullanılan en etkin yöntemlerden biri de weka.classifiers.lazy.IBk.java

978-1-4673-5563-6/13/$31.00 ©2013 IEEE

dosyasında anlatılmaktadır. Algoritma çapraz doğrulamaya dayalı kaba-kuvvet arama yapmaktadır.[7]

Geliştirilen uygulamada, eğitim işleminde kullanılan veri setlerinin bir takım meta özellikleri çıkarılarak yeni bir veri seti elde edilerek k en yakın komşu sınıflandırıcısı için hiper parametre değeri tahmin edilmeye çalışılmıştır. Her bir veri seti için, k hiper parametresinin en yaygın olarak kullanıldığı 6 farklı değeri ile sınıflandırma yapılarak hangi özelliklerdeki veri setinin hangi k değeri ile daha başarılı sonuç üreteceğine dair kurallar tanımlanmaya çalışılmıştır. Çıkarımı yapılan meta özelliklerin k hiper parametresinin tahmin edilmesinde ne kadar ayırt edici olduğu da incelenerek geliştirilen modelin doğruluğu araştırılmıştır.

Makalenin ikinci bölümünde sistemin genel yapısı tanıtılmış, üçüncü bölümde alınan sonuçlar karşılaştırmalı olarak verilmiş, dördüncü bölümde tartışma ve son bölümde de sonuç sunulmuştur.

II. SİSTEMİN GENEL YAPISI

Uygulama 4 adımdan oluşmaktadır; uygulamada kullanılacak veri setlerinin oluşturulması, veri setlerinin meta özelliklerinin çıkarılması, her bir veri setinin en yüksek başarıyı elde ettiği k değerinin belirlenmesi ve yeni gelen bir veri setinin meta özelliklerine bakılarak hangi k değeri ile sınıflandırılması gerektiğinin belirlenmesi.

A. Veri Seti

Uygulamada 200 adet veri seti kullanılmıştır. İlgili veri setlerinin 36 tanesi UCI makine öğrenmesi veri seti havuzundan alınmış olup diğer 164 tanesi bu veri setlerinden sınıf sayısı ikiden fazla olanlarının çeşitli sınıf kombinasyonları ile oluşturulmuştur. Çizelge I'de kullanılan UCI veri setleri ve her birinden kaçar adet üretildiği gösterilmektedir.

ÇİZELGE I. Kullanılan Veri Setleri

UCI Veri Seti - Sayı	**UCI Veri Seti - Sayı**
Abalone-11	Iris -4
Anneal-11	Kr-vs-kp - orijinal 1
Audiology-16	Labor - orijinal 1
Autos-16	Letter -27
Balance-Scale-4	Lymph - orijinal 1
Breast-Cancer- orijinal 1	Mushroom - orijinal 1
Breast-w- orijinal 1	Primary-tumor -11
Col10-13	Ringnorm - orijinal 1
Colic- orijinal 1	Segment-14
Credit-a- orijinal 1	Sick - orijinal 1
Credit-g - orijinal 1	Sonar - orijinal 1
D159 - orijinal 1	Soybean -17
Diabetes - orijinal 1	Splice - orijinal 1
Glass -16	Vehicle -11
Heart-Statlog - orijinal 1	Vote - orijinal 1
Hepatitis - orijinal 1	Vowel - orijinal 1
Hypothyroid -4	Waveform -4
Ionosphere - orijinal 1	Zoo - orijinal 1

B. Meta Özellik Çıkarımı

Her bir veri seti için 16 adet meta özellik çıkarımı yapılarak, 200 örnek ve sınıf bilgisi hariç 16 özellikten oluşan eğitim veri kümesi oluşturulmuştur. Kullanılan meta özellikler aşağıda sıralanmaktadır.

1. Sınıf sayısı
2. Özellik sayısı
3. Örnek sayısı
4. Özellik sayısının örnek sayısına oranı
5. Cfs ile seçilmiş özellik sayısı
6. Cfs ile seçilmiş özellik sayısının toplam özellik sayısına oranı
7. Veri kümesi C45 karar ağacı ile modellendiğinde ağaçtaki yaprak sayısı
8. Veri kümesi C45 karar ağacı ile modellendiğinde ağaçtaki yaprak sayısının örnek sayısına oranı
9. Veri kümesi C45 karar ağacı ile modellendiğinde ağaçtaki yaprak sayısının cfs ile seçilmiş özellik sayısına oranı
10. Veri kümesi PART Rules ile modellendiğinde oluşan kural sayısı
11. Veri kümesi PART Rules ile modellendiğinde oluşan kural sayısının örnek sayısına oranı
12. Veri kümesi PART Rules ile modellendiğinde oluşan kural sayısının özellik sayısına oranı
13. Veri kümesi PART Rules ile modellendiğinde oluşan kural sayısının cfs ile seçilmiş özellik sayısına oranı
14. Veri kümesi REP Tree karar ağacı ile modellendiğinde ağaçtaki düğüm sayısı
15. Veri kümesi REP Tree karar ağacı ile modellendiğinde ağaçtaki düğüm sayısının özellik sayısına oranı
16. Veri kümesi REP Tree ağacı ile modellendiğinde ağaçtaki düğüm sayısının cfs ile seçilmiş özellik sayısına oranı

C. k Değerinin Bulunması

Her bir veri setine, k hiper parametresinin 6 farklı değeri (1,3,5,7,9,11) için, Weka ile k en yakın komşu algoritması uygulanarak her birinin sınıflandırma başarısı hesaplanır. Sınıflandırma başarısının en yüksek olduğu k değeri, sınaması yapılan veri setinin sınıf bilgisi olarak atanır. Çizelge II'de veri setlerinin meta özelliklerine dayalı olarak oluşturulan eğitim veri kümesi bulunmaktadır.

ÇİZELGE II. Meta Özellikler

	Abalone	**Diabetes**	**Letter**
#Sınıf	19	2	26
#Özellik	11	9	17

#Örnek	4153	768	20000
#Öz/Örn	0.00265	0.01172	0.00085
#CfsOz	5	4	11
#CfsOz/Oz	0.45454	0.44444	0.64706
#C45Yaprak	1163	20	1226
#C45Yaprak/Orn	0.28004	0.02604	0.0613
#C45Yaprak/CfsOz	232.6	5	11.4545
#PARTRulesKural	925	13	776
#PARTRulesKural/Ornek	0.22273	0.01693	0.0388
#PARTRulesKural/Oz	84.0909	1.44444	45.6471
#PARTRulesKural/CfsOz	185	3.25	70.5454
#REPTreeDugum	287	49	1247
#REPTreeDugum/Oz	26.0909	5.44444	73.3529
#REPTreeDugum/CfsOz	57.4	12.25	113.364
K degeri	11	7	1

D. En Uygun k Değerinin Belirlenmesi

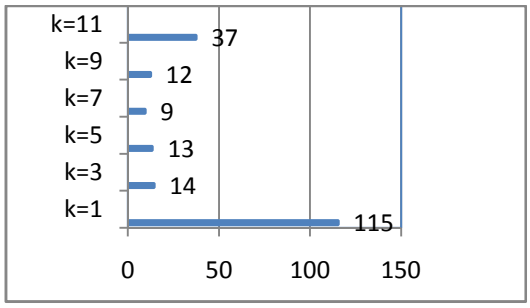
Uygulamanın amacı; k en yakın komşuluk algoritması ile sınıflandırılmak istenilen bir veri setinin meta özelliklerine bakılarak hangi k değeri ile sınıflandırılmasına karar verebilmektir.

Bu amaçla kullanıcının k değerini belirlemek istediği veri setini uygulamaya yüklemesinin ardından eğitim veri setini oluşturmakta kullanılan meta özellik çıkarımı ilgili veri seti için de yapılır. Meta özellikleri içeren eğitim veri kümesi ile öğrenme işlemi yapılarak yeni veri setinin hangi k değeri ile sınıflandırılması gerektiği bilgisi elde edilir.

III. DENEYSEL ÇALIŞMALAR

Meta özellik eğitim veri kümesi 16 meta özellik ve 1 sınıf bilgisi olmak üzere toplam 17 özellikten oluşmaktadır. Sınıf bilgisi her bir veri setinin k en yakın komşuluk sınıflandırıcısı için en başarılı olduğu k değeridir.

200 veri setinden 115 tanesi k'nın 1 değeri ile etiketlenmiştir. Veri setlerinin 6 farklı k değerine göre dağılımı Şekil I 'deki gibidir.



ŞEKİL I. K Değerlerinin Dağılımı

Tüm veri setleri için k değerlerinin bulunması işleminin sonucunda, bir UCI veri seti hangi k değeri ile etiketlendi ise o veri setinden türetilen diğer veri setlerinin de genellikle aynı k değeri ile etiketlendiği gözlemlenmiştir.

Belirlenen 16 adet meta özelliğin hangilerinin belirleyici olduğuna karar verebilmek için meta özellik veri kümesine Weka ile Correlation Feature Selection (Cfs) işlemi uygulanmış ve Cfs işlemine göre belirleyici olan özellikler çıkarılmıştır. İşlem sonucunda, Cfs, tanımlanan 16 adet meta özellikten aşağıdaki 3 tanesini seçmiştir.

1. Örnek sayısı
2. Cfs ile seçilmiş özellik sayısı
3. PART Rules kural sayısının örnek sayısına oranı

Cfs ile seçilmiş yukarıdaki 3 adet özelliğin tek başına yeterli olup olmadığına karar verebilmek için; tüm meta özellikleri içeren veri seti ile sadece Cfs ile seçilmiş meta özellikleri içeren veri seti ayrı ayrı değerlendirilerek her ikisi için de Root Mean Squared Error (RMSE) değerleri hesaplanmıştır. Çizelge III 'de 4 farklı yöntem ile 10-çapraz geçerleme kullanılarak elde edilmiş RMSE değerleri bulunmaktadır

ÇİZELGE III. Meta Özellik Seçimi

	Tüm Özellikler	**Cfs ile Seçilen Özellikler**
REP Tree	2.9946	3.1599
SMO	3.5785	3.6382
M5Rules	2.7767	3.1291
Decision Table	3.4633	3.3133

Çizelge III incelendiğinde Karar Tablosu dışındaki tüm yöntemlerde sadece Cfs ile seçilen özellikler kullanıldığında hata oranının arttığı görülmüştür.

Meta özellik eğitim veri seti için Rep Tree ile oluşturulan ağaç incelendiğinde kökte "PART Rules kural sayısının örnek sayısına oranı" özelliğinin yer aldığı görülmüştür. Rep Tree ağacında kullanılan diğer meta özellikler ise aşağıdaki şekilde sıralanmaktadır

1. Sınıf sayısı
2. PART Rules kural sayısının cfs ile seçilmiş özellik sayısına oranı

SMO algoritması ile elde edilen sonuçlar incelendiğinde ise en yüksek ağırlık değerine sahip meta özellikler aşağıdaki şekilde sıralanmaktadır.

1. PART Rules kural sayısının örnek sayısına oranı
2. Sınıf sayısı
3. Özellik sayısının örnek sayısına oranı
4. Örnek sayısı

M5Rules yöntemi ise elde ettiği kurallarda aşağıdaki meta özellikleri kullanmıştır.

1. PART Rules kural sayısının örnek sayısına oranı
2. Özellik sayısının örnek sayısına oranı
3. Sınıf sayısı
4. C45 yaprak sayısının örnek sayısına oranı
5. C45 yaprak sayısının cfs ile seçilmiş özellik sayısına oranı
6. Özellik sayısı
7. Örnek sayısı

8. Cfs ile seçilmiş özellik sayısının toplam özellik sayısına oranı
9. PART Rules kural sayısı
10. PART Rules kural sayısının özellik sayısına oranı

Decision Table kurallarında ise sınıf sayısı, özellik sayısı ve PART Rules kural sayısının örnek sayısına oranı özellikleri kullanmıştır. Çizelge IV'de decision table kullanılarak elde edilen kurallar bulunmaktadır.

ÇİZELGE IV. Decision Table Kuralları

Kural	K değeri
İf ((14< ss <=18.8) or (4.4< ss <=6.8)) and (61.6<özS<=89.9) and (0.045239<part<=0.06749)	1
İf (4.4< ss <=11.6) and (özS <=33.3) and (part<=0.8974)	
İf (21.2< ss) and (özS <=33.3) and (0.22989<part<=0.45239)	
İf (ss<=4.4) and (61.6< özS <=89.9) and (0.022989<part<=0.08974)	3
İf (ss <=9.2) and (özS <=61.6) and (part<=0.08974)	5
İf (ss <=4.4) and (61.6< özS <=89.9) and (part<=0.022989)	
İf (ss <=4.4) and (33.3< özS <=61.6) and (0.045239<part<=0.06749)	7
İf (ss<=4.4) and (33.3< özS <=61.6) and ((part<=0.022989) or (0.08974<part<=0.11199))	9
İf ((ss<=4.4) or (16.4<ss<=18.8)) and (özS <=61.6) and ((0.06749<part<=0.08974) or (0.200992<part))	10
İf ((18.8<ss<=21.2) or (11.6<ss<=16.4)) and (özS <=33.3) and (0.200992<part)	11
İf (6.8< ss <=9.2) and (özS <=33.3) and (0.08974<part<0.200992)	
İf (9.2< ss <=11.6) and (özS <=33.3) and ((0.178742<part<=0.200992) or (0.11199<part<=0.134241))	
İf (ss<=4.4) and (259.7< özS) and (part<=0.22989)	

ss=Sınıf Sayısı
özS=Özellik Sayısı
part=Part rules kural sayısının örnek sayısına oranı

Veri seti havuzunda en fazla 26 en az 2 olmak üzere çeşitli sınıf sayılarına sahip veri setleri bulunmaktadır. Ancak bunların çoğunluğu 2 veya 3 sınıfa sahiptir. Sınıf sayısının daha az olduğu veri setleri k nın 3, 5, 7 ve 9 değerleri için daha başarılı olurken sınıf sayısının daha fazla olduğu veri setlerinde k nın 1 ve 11 değerlerinin daha başarılı olduğu görülmüştür.

KNN sınıflandırıcısı mesafe ölçümüne göre sınıf belirlediği için özellik sayısı arttıkça algoritmanın karmaşıklığı da artmaktadır. Çizelge IV'deki kurallarda özS zaman zaman farklı değer ve aralıklarda olsa da ss ve part değerlerine göre belirleyiciliği daha az çıkmıştır.

Bir veri seti ile ilgili oluşturulan kural sayısı ne kadar çoksa veri seti o kadar zor modellenebiliyor demektir. Yani part sayısı aslında bir ölçüde verinin zorluğunu göstermektedir. Geliştirilen uygulamada kullanılan meta özelliklerden en belirleyicisi part değeri olmuştur. Part sayısının daha yüksek olduğu durumlarda ilk sırada 1 olmak üzere k'nın 1, 10 ve 11 değerleri seçilmiştir. Yani veri seti zorlaştıkça k'nın en düşük veya en büyük değerleri daha başarılı olmaktadır.

Genel olarak bakıldığında, sayısal değerler bire bir aynı olmasa bile ss, özS ve part değişkenlerinde ki artış ve azalışların benzerliği yönünden değerlendirildiğinde 2 grup oluşmaktadır. k nın 1 ve 11 değerleri bir grubu 3, 5, 7 ve 9 değerleri ikinci grupta bulunur.

Tüm yöntemler göz önüne alındığında meta özelliklerden REP tree ile ilgili olanların hiç kullanılmadığı görülmüştür. Cfs yönteminin seçtiği özelliklerden sınıf sayısı ile PART Rules kural sayısının örnek sayısına oranı özellikleri ise tüm yöntemler tarafından kullanılmıştır.

IV. SONUÇ

Bu çalışma K-en yakın komşuluk sınıflandırıcısı için veri setinin karakteristiğine bakılarak k hiper parametresini tahmin edebilmeyi amaçlamıştır. Eğitim işlemi için 200 adet veri setinin bir takım meta özellikleri çıkarılmış, her bir veri setine 6 farklı k değeri (1, 3, 5, 7, 9, 11) ile KNN uygulanarak en yüksek doğruluğu elde eden k değeri seçilmiş ve bu bilgilerle yeni bir eğitim seti oluşturulmuştur.

200 veri setinden 115 tanesinde KNN için en başarılı k değeri 1 çıkmıştır. Çıkarılan meta özelliklere cfs uygulandığında 16 tane özelliğin 3 tanesi ayırt edici bulunmuştur. Meta özellik veri seti tüm özellikler ile ve sadece cfs ile seçilmiş özellikler ile SMO, REP tree, M5 Rules ve Decision Table yöntemleri kullanılarak denenmiş hepsi için de sadece cfs ile seçilmiş özellikler kullanıldığında hatanın arttığı görülmüştür. Bununla birlikte denenen tüm yöntemlerde cfs ile seçilmiş özelliklerde sınıf sayısı ve veri seti PART Rules ile modellendiğinde oluşan kural sayısının örnek sayısına oranı özelliklerinin belirleyici bulunduğu ve ilk sıralarda kullanıldığı görülmüştür. Decision table ile elde edilen kurallar incelendiğinde ise en çok kuralın k'nın 1 ve 11 değerleri için tanımlandığı, sınıf sayısının daha fazla olduğu veri setlerinde genellikle k'nın 1 ve 11 değerlerinin daha başarılı sonuç verdiği görülmüştür.

KAYNAKÇA

[1] Ricardo Vilalta and Youssef Drissi, A Perspective View and Survey of Meta-Learning, (2002)

[2] Kate A. Smith-Miles, Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection, (2009)

[3] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer.: Principles of Data Mining. The MIT Press. (2001)

[4] Mitchell, T. Machine Learning, McGraw Hill. ISBN (1997)

[5] Xie, Z., Hsu, W., Liu, Z., Lee, M.: SNNB: A Selective Neighborhood Based Naive Bayes for Lazy Learning. Proceedings of the Sixth Pacific-Asia Conference on KDD. Springer (2002) 104-114

[6] Frank, E., Hall, M., Pfahringer, B.: Locally Weighted Naive Bayes. Proceedings of the Conference on Uncertainty in Artificial Intelligence (2003). Morgan Kaufmann(2003)

[7]http://grepcode.com/file/repo1.maven.org/maven2/nz.ac.waikato.cms.weka/wekadev/3.7.5/weka/classifiers/lazy/IBk.java#IBk.crossValidate%28%29