PROTEİN ETKİLEŞİM TAHMİNİ İÇİN POZİTİF ETİKETSİZ ÖĞRENME ALGORİTMALARININ GELİŞTİRİLMESİ

DORUK PANCAROĞLU

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

NİSAN 2014

ANKARA

Fen Bilimleri Enstitü Onayı

_____

Prof. Dr. Necip CAMUŞCU

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

_____

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Doruk PANCAROĞLU tarafından hazırlanan PROTEİN ETKİLEŞİM TAHMİNİ İÇİN POZİTİF ETİKETSİZ ÖĞRENME ALGORİTMALARININ GELİŞTİRİLMESİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

_____

Yrd. Doç. Dr. Mehmet TAN

Tez Danışmanı

Tez Jüri Üyeleri

Başkan: Doç. Dr. Osman ABUL _____

Üye: Yrd. Doç. Dr. Ersin Emre ÖREN _____

Üye: Yrd. Doç. Dr. Mehmet TAN _____

TEZ BİLDİRİMİ

Tez içerisindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazın kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

_____

Doruk PANCAROĞLU

Üniversitesi	**:**	**TOBB Ekonomi ve Teknoloji Üniversitesi**
Enstitüsü	**:**	**Fen Bilimleri**
Anabilim Dalı	**:**	**Bilgisayar Mühendisliği**
Tez Danışmanı	**:**	**Yrd. Doç. Dr. Mehmet TAN**
Tez Türü ve Tarihi	**:**	**Yüksek Lisans – Nisan 2014**

Doruk PANCAROĞLU

PROTEİN ETKİLEŞİM TAHMİNİ İÇİN POZİTİF ETİKETSİZ ÖĞRENME ALGORİTMALARININ GELİŞTİRİLMESİ

ÖZET

Protein etkileşim tahmini için ikili sınıflandırmada, mevcut iki adet proteinin negatif (etkileşime girmeyen) olduğunu tespit edebilmek zor bir işlemdir. Bu zorluğun sebeplerinden biri bu sınıflandırmayı yapmaya yardımcı olacak eğitim kümesi için hiçbir zaman etkileşmeyen örnekleri temin etmenin güç olmasıdır. Ayrıca, bir protein çiftinin etkileşmediği ispatlanmış olsa bile, protein etkileşim veri tabanlarında bu negatif örneklere yer verilmez. Bu durum sebebiyle gerçek negatif örnek kullanmayan öğrenme algoritmalarına bir ihtiyaç doğmuştur. Bu çalışmada, yüksek performansları sebebiyle seçilen iki adet pozitif etiketsiz öğrenme algoritması, AGPS ve Roc-SVM için geliştirmeler yapılması hedeflenmiştir. Bu algoritmalara iki adet geliştirme yapılacaktır: algoritmaların sınıflandırma için kullandığı support vector Machines (SVM) sınıflandırıcısı yerine Random Forest sınıflandırıcısını kullanmak (AGPS-RF ve Roc-RF) ve iki algoritmayı birleştirerek sonuçlarını bir oylama sistemine sokmak (Karma Algoritma). Bu geliştirmeler yapıldıktan sonra algoritmalar önceki halleri ile ve yaygın olarak kullanılan iki adet sınıflandırma algoritması (CLR ve ARACNE) ile karşılaştırılarak performansları incelenmiştir. Yapılan karşılaştırmalarda, AGPS-RF, Roc-RF ve Karma Algoritma, SVM kullanan seleflerine göre daha iyi performans vermiştir. CLR ve ARACNE ile yapılan karşılaştırmalarda ise Roc-RF ve Karma Algoritma'nın daha performanslı olduğu görülmüştür.

Anahtar Kelimeler: Protein Etkileşim Ağları, İkili Sınıflandırma, Pozitif Etiketsiz Öğrenme, Random Forests, Support Vector Machines

University	**:**	**TOBB Economics and Technology University**
Institute	**:**	**Institute of Natural and Applied Sciences**
Science Programme	**:**	**Computer Engineering**
Supervisor	**:**	**Asst. Prof. Dr. Mehmet TAN**
Degree Awarded and Date	**:**	**M.Sc. – April 2014**

Doruk PANCAROĞLU

IMPROVING POSITIVE UNLABELED LEARNING ALGORITHMS FOR PROTEIN INTERACTION PREDICTION

ABSTRACT

In binary classification for protein interaction prediction, labeling two proteins as negative (not interacting) is a hard task. This problem is caused by the difficulty of obtaining two training samples that would never interact. Furthermore, the protein interaction databases do not include negative samples, even if the samples have been shown to be non-interacting. The aforementioned difficulty in obtaining true negative samples created a need for learning algorithms that does not use negative samples. This study aims to improve upon two well-performing positive unlabeled learning algorithms, AGPS and Roc-SVM for protein interaction prediction. Two extensions to these algorithms is proposed; the first one is to use Random Forests as the classifier instead of support vector Machines (AGPS-RF and Roc-RF) and the second is to combine the results of AGPS and Roc-SVM using a voting system (Hybrid Algorithm). After these two approaches are implemented, the results were compared to the original algorithms as well as two well-known learning algorithms, ARACNE and CLR. In the tests and comparisons, both Random Forest algorithms and the Hybrid algorithm performed well against the original SVM-classified ones. The improved Roc-RF and Hybrid Algorithms also performed well against ARACNE and CLR.

Keywords: Protein Interaction Networks, Binary Classification, Positive Unlabeled Learning, Random Forests, Support Vector Machines

TEŞEKKÜR

Tez çalışmalarım süresince beraber çalıştığım, yardımını, desteğini ve tavsiyelerini aldığım değerli hocam Yrd. Doç. Dr. Mehmet TAN'a teşekkür ederim. Yaptığı önceki çalışmalar ve tez çalışmam süresindeki yardımları ile bu çalışmanın gerçekleşmesine katkıda bulunan Cumhur KILIÇ'a teşekkür ederim.

İÇİNDEKİLER

ÇİZELGELERİN LİSTESİ

KISALTMALAR

Kısaltmalar	Açıklama
AGPS	Annotating Genes for Positive Samples
ARACNE	Algorithm for the Reconstruction of Accurate Cellular Networks
Bagging	Bootstrap Aggregation
CART	Classification and Regression Tree
CLR	Context Likelihood of Relatedness
E. Coli	Escherichia Coli
GN	Gerçek Negatif
GP	Gerçek Pozitif
LibSVM	Library for Support Vector Machines
MCC	Matthews Correlation Coefficient
PE	Pozitif Etiketsiz
PPE	Protein-Protein Etkileşimi
PosOnly	Positive Only
$PSoL_m$	Positive Sample Only Learning Modified
$PSoL_o$	Positive Sample Only Learning Original
RF	Random Forest
Roc	Rocchio Metodu
S-EM	Spy Technique and the Expectation-Maximization
SN	Sahte Negatif
SP	Sahte Pozitif
SVM	Support Vector Machines
WEKA	Waikato Environment for Knowledge Analysis

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
C()	Güçlü negatif seçimi için oluşturulan prototip vektörler
f(i)	i. Tekrarlama için Sınıflandırıcı (Random Forest veya SVM)
GN	Güçlü negatifler kümesi (Güçlü, olasılığı yüksek anlamında)
N	Negatif Örnekler Kümesi
NN	Nihai Negatif Örnekler Kümesi (Algoritmada en son kullanılacak negatifler kümesi)
P	Pozitif Örnekler Kümesi
P1	Pozitif Örnekler Alt Kümesi (Çapraz Sağlama İçin)
P2	Pozitif Örnekler Alt Kümesi (Çapraz Sağlama İçin)
U	Etiketsiz Örnekler Kümesi
$\mathbf{U_{evo}}$	Etiketsiz Örnekler ile Oluşturulan Geçici Küme
$\mathbf{U_{yeni}}$	Etiketsiz Örnekler ile Oluşturulan Geçici Küme

1 GİRİŞ

1.1 Giriş ve Çalışmanın Amacı

İkili sınıflandırma (Binary Classification), mevcut bir veri kümesinin elemanlarının belirli bir özelliğe sahip olup olmadığını bulma işlemine verilen isimdir. Bu işlem, mevcut veri kümesindeki örneklerin iki gruba ayrılması ile yapılır: sayıca daha az örnek içeren küçük grup ve veri kümesinin kalan örneklerini içeren büyük grup. Aynı zamanda elimizdeki sayıca az örnek içeren gruptaki elemanların belirli bir özelliğe sahip olduğu ya da olmadığı bilinmektedir. Küçük gruptaki örneklerden belirli bir özelliğe sahip olanlar bu sebeple pozitif sınıf olarak da adlandırılırken bu özelliğe sahip olmayan örnekler negatif sınıf olarak adlandırılır. Büyük gruptaki örneklerin ise pozitif veya negatif olduğu bilinmediği için bu gruptaki örnekler etiketsiz (unlabeled) sınıf olarak adlandırılır.

Gözetimli öğrenme (supervised learning), bir veri kümesini sınıflandırmak için hem pozitif hem negatif örnekleri kullanan bir öğrenme türüdür. Fakat, bu tip sınıflandırmaların yapıldığı başta tıp olmak üzere bir çok mecrada negatif olduğu bilinen örnekleri temin etmek negatif örnekleri sınıflandırmanın doğası gereği oldukça zordur ve çoğu zaman imkansızdır.

Proteinler canlı organizmalardaki faaliyetlerin neredeyse tamamının gerçekleşmesini sağlayan ve aynı zamanda bu faaliyetleri düzenleyen biyolojik moleküllerdir. Yapıtaşı olarak amino asitlerden oluşan proteinler, bu amino asitlerin belirli bir sıra ile dizilmesi ile birbirlerinden farklılaşırlar.

Canlı organizmalardaki faaliyetleri sırasınca proteinler çoğu zaman başka proteinlerle beraber çalışmaktadırlar. Proteinler tek bir faaliyet için birçok farklı protein ile de etkileşime geçebilmektedirler. Proteinler arası etkileşim şekilleri olarak hücre içindeki sinyallerin iletimi, diğer proteinlerin taşınması, enzimler kullanılarak hücrede üretim yapılması örnek gösterilebilir.

Protein-Protein Etkileşimi (Protein-Protein Interaction) adı verilen bu durum, birden fazla bilimsel alanda kullanılmaktadır. PPE, proteinlerin görevlerini öğrenme, insan vücudunun işleyişi konusunda yeni bulgulara yön verme ve önemli hastalıkların araştırılması gibi konularda kilit rol oynamaktadır.

Proteinler arasındaki etkileşimlerin tespit edilmesi için farklı yöntemler mevcuttur. Bu yöntemler Protein-Protein Etkileşimi Tahmini (Protein-Protein Interaction Prediction) şemsiyesi altında toplanmıştır. Bu yöntemlerden Filogenetik Profilleme [1] (Phylogenetic Profiling), karşılaştırılan proteinleri bu proteinlerin bulunduğu canlılar açısından ele alarak bu canlıların birbirine yakın türler olup olmadığı üzerinden tahmin yürütürken Rosetta Taşı (Rosetta Stone) [2] yöntemi ise bileşik proteinleri baz alarak etkileşim tahminlerini bu koldan yapmaktadır.

Protein-Protein Etkileşim Tahmini yöntemlerinden öne çıkan bir diğeri de bu çalışmanın öznesi olan sınıflandırma metotlarını içerir. Sınıflandırma metotları da etkileşime geçtiği ve geçmediği bilinen pozitif ve negatif örnekler üzerinden tahminlerini yapar. Sınıflandırma metotlarının önceki metotlara göre olumlu yönlerinden biri alanlardan (domain) bağımsız olmasıdır.

İki protein örneği arasındaki bir etkileşimin mevcut olması laboratuvar koşullarında kolaylıkla anlaşılabilen bir durumdur. Bu ortamda gözlenen herhangi bir etkileşim bu iki proteinin her zaman bir etkileşim içinde olacağını belirtir. Fakat iki protein arasında laboratuvar ortamında bir etkileşim gözlenmemesi bu proteinlerin hiçbir zaman etkileşimde bulunmayacağını garantilemez. Etkileşim olmadığının gözlemlenmesi çevre koşulları, zaman ve benzeri birçok farklı sebepten dolayı olmuş olabilir. Bu proteinlerin hiçbir zaman etkileşime giremeyeceğini kesin anlamda söyleyebilmek için etkileşim testlerinin olabilecek tüm koşullarda yapılması, zaman ve kaynak açısından uygulanabilir değildir. Bu durum negatif örneklerin temininin zorluğuna sebep olmaktadır.

Negatif örnekleri temin etmenin zorluğundan dolayı ikili sınıflandırma işleminde sadece pozitif ve etiketsiz (unlabeled) örnek kümelerini kullanan algoritmalara ihtiyaç duyulmuştur. Bu türdeki, negatif örnekleri kullanmayan algoritmalara pozitif etiketsiz öğrenme algoritmaları adı verilmiştir.

Pozitif Etiketsiz (PE) öğrenme algoritmaları bir çok alanda kullanılsa da bu çalışmanın hedefinde proteinler arası etkileşim ağları vardır. Proteinler arası etkileşim ağları temelde bir çizgedir (graph). Bu çizgede düğümler (nodes) proteinleri temsil ederken kenarlar (edges) da bu proteinler arasındaki ilişkilerin varlığını temsil eder. Herhangi bir protein kümesinin içinde bulunduğu bu tip bir etkileşim ağında toplam protein sayısına göre oldukça az sayıda kenar, yani ilişkisi bulunan protein bulunması genel bir kanıdır.

Etiketsiz örneklerin fazlalığı ve negatif örneklerin olmaması gibi koşullar pozitif etiketsiz öğrenme algoritmalarını yarı gözetimli öğrenme sınıfına sokmaktadır. Bu sınıf hiyerarşisine rağmen pozitif etiketsiz öğrenme algoritmaları mevcut yarı gözetimli öğrenme algoritmalarından farklıdır.

Pozitif etiketsiz öğrenme algoritmalarıyla ilgili detaylı bir inceleme Mehmet Tan ve Cumhur Kılıç tarafından yapılmıştır [3]. Mevcut pozitif etiketsiz öğrenme algoritmalarından AGPS [4], Roc-SVM [5], PSoL [6] ($PSoL_m$ ve $PSoL_o$ beraber), Carter [7], PosOnly [8], Bagging SVM [9] ve S-EM [10] bu incelemede test edilmiştir. Bu incelemede karşılaştırmalarda baz alınmak üzere sadece SVM [11] (Support Vector Machines) algoritmasına sınıflandırma yaptıran SVM_{only} isimli bir algoritma da teste katılmıştır.

Çalışma boyunca algoritmalar ile yapılan denemelerde alınan sonuçlar, bu denemelerde kullanılan veri kümesindeki elemanların arasındaki ilişkinin varlığına yönelik yapılan sınıflandırmanın sonuçlarının bir derlemesidir. Bu sonuçlar Gerçek Pozitif (GP), Sahte Pozitif (SP), Gerçek Negatif (GN) ve Sahte Negatif (SN) olarak dört farklı şekilde elde edilebilmektedir. Keskinlik, doğruluk, f-ölçüsü[12] ve MCC[13] değerleri bu dört sonucu kullanarak bulunmaktadır.

Keskinlik (Precision), bir sınıflandırma işleminde alınan sonuçların farklı işlemlerde de benzerlik göstermesidir. Doğruluk (Recall), bir sınıflandırma işleminde alınan sonuçların gerçeğe ne kadar yakın olduğudur. F-Ölçüsü (F-Measure) bu iki değerin harmonik ortalaması, MCC (Matthews Correlation Coefficient) değeri ise ikili sınıflandırma işleminin genel kalitesini ölçmeye yarayan bir değerdir. Bu değerlerin nasıl hesaplandığı Ek A bölümünde Çizelge A.1'de görülebilir.

Yapılan bu karşılaştırmada iki adet algoritma öne çıkmıştır: AGPS ve Roc-SVM. İki algoritma da yapılan karşılaştırmalı testlerde keskinlik, doğruluk, f-ölçüsü, MCC gibi alanlarda birbirlerine yakın sonuçlar alarak üst sıraya yerleşmişlerdir.

Yapılan çalışmanın detaylı sonuçları çizelge 1.1'de görülebilir.

Çizelge 1.1 Pozitif Etiketsiz Öğrenme Algoritmalarının F-Ölçüsü Değeri Üzerinden Karşılaştırmalı Sonuçları

Algoritma Adı	Ortalama F-Ölçüsü
AGPS	0,235
Roc-SVM	0,230
S-EM	0,218
$PSoL_m$	0,210
PosOnly	0,200
CART	0,183
Bagging	0,183
$PSoL_o$	0,171
SVM_{only}	0,114

Bu çalışmadaki amaç pozitif etiketsiz öğrenme algoritmaları arasında verdiği sonuçların doğruluğu açısından öne çıkan AGPS ve Roc-SVM algoritmalarını farklı yollar ile daha da iyi hale getirmektir. Bu iyileştirme çalışması iki koldan yürütülecektir. İlk kolda algoritmaların kullandığı SVM modeli Random Forest (RF) [14] modeli ile değiştirilmiş ve algoritmalar aynı veri grubu ile test edilmiştir. Random Forest modeli kullanan bu iki algoritma bundan sonra AGPS-RF ve Roc-RF olarak adlandırılacaktır. İkinci kolda ise karma bir yol uygulanmıştır. Bu karma yolda algoritmaların kullandığı SVM modeli korunmuş ve algoritmaların oluşturduğu sonuçlar bir oylama sistemi geliştirilerek birleştirilmiştir. Bu şekilde iki algoritmanın da vereceği sonuçların en doğru olanlarının edinilmesi hedeflenmiştir. Bu karma yolu kullanarak oluşturulan algoritma ise Karma Algoritma olarak adlandırılacaktır.

İki farklı şekilde yürütülen algoritma iyileştirme aşaması tamamlandığında oluşturulan sonuçlar algoritmaların iyileştirme yapılmamış halleri ile karşılaştırılmıştır. Bu karşılaştırmalarda AGPS-RF ve Roc-RF algoritmaları SVM kullanan orijinal algoritmalara göre doğruluk oranı daha yüksek olan sonuçlar vermişlerdir. Karma Algoritma da orijinal iki algoritmaya göre doğruluk oranı daha yüksek olan sonuçlar vermiştir. İyileştirilen algoritmaları iyileştirme yapılmamış halleriyle karşılaştırdıktan sonraki adımda benzer veri grupları ile iyi sonuç verdiği bilinen diğer algoritmaları benzer bir karşılaştırmaya sokmak düşünülmüştür. Karşılaştırılacak bilinen algoritmalar olarak CLR [15] ve ARACNE [16] seçilmiştir. Yapılan karşılaştırmalarda Roc-RF ve Karma Algoritma'nın CLR ve ARACNE'ye göre doğruluk oranı daha yüksek olan sonuçlar verdiği görülmüştür.

2 KONUYLA İLGİLİ YAPILMIŞ ÖNCEKİ İŞLER

Bu bölümde geliştirme yapılacak pozitif etiketsiz öğrenme algoritmaları AGPS ve Roc-SVM ile ilgili bilgi verilecektir. Önceki bölümde bahsedildiği gibi Mehmet Tan ve Cumhur Kılıç'ın yaptığı çalışmada [3] öne çıkan AGPS ve Roc-SVM algoritmaları, yüksek performansları ile yapılacak geliştirmelerde kullanılacak iki algoritma olarak belirlenmiştir ve daha detaylı olarak sonraki kısımlarda incelenecektir.

2.1 SVM

SVM (Support Vector Machines), gözetimli öğrenme metotlarını kullanan bir sınıflandırma modelidir. Eğitim ve test kümelerini kullanan SVM, elindeki örnekleri bir düzleme yerleştirerek yerleştirdiği örneklerin düzlemin hangi tarafında olduğunu tespit ederek sınıflandırma yapar. SVM, Vladimir Vapnik ve Corinna Cortes tarafından 1995 [17] yılında bulunmuş bir yöntemdir. SVM hem doğrusal (linear) hem de doğrusal olmayan (non-linear) sınıflandırma yapabilmektedir. Aradaki fark sınıflandırmanın ayracının doğrusal bir fonksiyon mu yoksa polinom ya da hiperbol cinsi bir fonksiyon mu olduğudur. SVM metin sınıflandırma, görsel sınıflandırma ve protein sınıflandırma gibi farklı uygulama alanlarında iyi performans gösteren bir model olarak öne çıkmaktadır.

2.2 AGPS

AGPS (Annotating Genes with Positive Samples), X-M. Zhao, Yong Wang, Luonan Chen ve Kazuyuki Aihara tarafından geliştirilmiş bir pozitif etiketsiz öğrenme algoritmasıdır. İki adımlı bir algoritma olan AGPS ön hazırlık, başlangıç negatif

kümesini oluşturma, negatif küme genişletme ve sınıflandırma adında dört bölümden oluşur.

Ön hazırlık bölümü kısmen tüm algoritmayı kapsamaktadır. AGPS'nin sınıflandırma yapacağı örnekler U (unlabeled, etiketsiz) kümesi ve P kümesi (pozitif) olarak tanımlanır. AGPS örneklerini tanımlarken 10 katlı çapraz sağlama metodunu kullanmaktadır. Çapraz sağlama [18] metodunun kullanılmasındaki amacın yapılacak sınıflandırmada kullanılacak örneklerin dağıtımının homojen bir şekilde yapılması ve bu sayede sınıflandırmalardaki isabet oranının artırılmasıdır. Çapraz sağlama metodunun her tekrarlamasında (10 katlı olduğu için algoritma 10 kez tamamen tekrarlanacaktır) P kümesi P1 ve P2 olarak ikiye bölünecektir. Bu kümelerden P1 kümesi eğitim kümesi (training set), P2 kümesi ise sağlama kümesi (validation set) olarak adlandırılacaktır. Örneğin, P kümesinin toplam 1000 elemanı olduğunu varsayarsak çapraz sağlamanın ilk tekrarlamasında P1 kümesi P kümesindeki ilk 100 elemandan oluşacaktır. P2 kümesi ise 101'den 1000'e kadar olan elemanlardan oluşacaktır. İkinci tekrarlamada P1 101-200 arasındaki elemanlardan oluşurken P2 kümesi 1-100 ve 201-1000 arasındaki elemanlardan oluşacaktır. Bu şekilde P kümesinin tüm elemanları hem eğitim kümesi, hem de sağlama kümesinde kullanılmış olacaktır.

Algoritmanın başında ikiye bölünen P kümesinin P2 alt kümesi, U kümesine eklenecektir. Bu şekilde etiketsiz elemanlardan oluşan kümeye pozitif olduğunu bildiğimiz elemanlar eklenecektir. U ve P2'den oluşan bu yeni kümeye U_{yeni} adı verilecektir.

Bu ön hazırlık aşaması bittikten sonraki ilk adımda P1 isimli eğitim kümesi ve U_{yeni} kümeleri kullanılarak f(1) ismi verilecek sınıflandırıcı üretilecektir. Bu eğitim adımında SVM kullanılacaktır. Oluşan f(1) sınıflandırıcısı, U_{yeni} kümesini SVM ile etiketlemede kullanılacaktır. Bu etiketleme sonucunda U_{yeni} içindeki örneklerden negatif olarak etiketlenecek olanlar N(1) isimli (1 çapraz sağlama sayısı olacak şekilde) negatifler kümesinde toplanacaktır. U_{yeni} kümesinden N(1) kümesi çıkartılarak etiketsiz kümenin boyutu küçültülecektir. Bu şekilde N(1) isimli başlangıç negatif kümesi oluşturularak ilk adım tamamlanacaktır.

İkinci adım kendi içinde tekrarlanacak bir adımdır. Bu adımda ilk adımdaki gibi P1 ve N(1) kümeleri SVM kullanılarak f(i) sınıflandırıcısını üretecektir (i değeri tekrarlama sayısı olacak şekilde). Üretilen sınıflandırıcı FC isimli sınıflandırıcılar dizisine eklenirken N(i) kümesi ise NS isimli negatifler dizisine eklenecektir. U_{yeni} kümesi f(i) sınıflandırıcısını kullanarak etiketlenecektir. Bu etiketlemede negatif olarak etiketlenen U_{yeni} elemanları N(2) kümesine eklenecektir. N(2) kümesi (ilk tekrarlamada 2 (1+1), sonraki tekrarlamalarda i+1 şeklinde numaralandırılacaktır. Kümenin elemanları önceki adımda olduğu gibi U_{yeni} kümesinden çıkarılacaktır. Bu adımın sonraki tekrarlamalarında f(i + 1) ve N (i + 1) kümeleri yaratılmaya devam edecektir. Tekrarlamalar ve bu adım U_{yeni} kümesinin kendisinden çıkarılan elemanlar ile boyutunun P kümesinin boyutuna kadar indiği zaman bitecektir.

Üçüncü adımda FC dizisinin içindeki tüm f() sınıflandırıcılar U_{yeni} kümesini etiketleyecek ve bu etiketlemeler içinde en isabetli tahminleri yapan sınıflandırıcının negatif küme karşılığı NN (Nihai negatif küme) olarak belirlenecektir. Örneğin en isabetli tahminler f(4) sınıflandırıcısından gelmiş ise NN olarak N(4) kümesi seçilecektir. Bu adımın devamında f(son) isimli bir sınıflandırıcı P ve NN kümelerini kullanarak SVM ile oluşturulacaktır. Başlangıçtaki U kümesi f(son) sınıflandırıcısını kullanarak etiketlenecek ve sonuçlar elde edilecektir.

AGPS algoritmasının kısa kodu çizelge 2.1'de görülebilir.

Çizelge 2.1 AGPS Algoritması Kısa Kodu

Kısaltmalar: U: etiketsiz örnekler kümesi; P: pozitif örnekler kümesi; P1,2: P kümesinin alt kümeleri; N: negatif örnekler kümesi; f: sınıflandırıcı
0. Çapraz Sağlama Adımı
- U_{yeni} = U + P2 (Çapraz doğrulama işlemi için P kümesinin bir kısmı alınmaktadır) olarak belirlenir. - Bu adım algoritmanın kalanını kapsayarak çapraz sağlama sayısı kadar tekrarlayacaktır. Amacı U kümesini sınıflandırırken kullanılacak örnekler arasında homojen bir dağılım yapmaktır. Detaylı bilgi için 2.1 bölümüne bakılabilir.

1. İlk Negatif Küme Oluşturma
- Sınıflandırıcı f(1), P1 ve U_{yeni} kullanılacak oluşturulur. - Bu adımdaki sınıflandırıcı SVM sınıflandırıcısıdır. - U_{yeni}, f(1) kullanılarak sınıflandırılır. Bu sınıflandırmadan N(1) negatif kümesi elde edilir. Bu küme 2. adımda kullanılacaktır. - U_{yeni} kümesinden N(1) kümesindeki elemanlar çıkarılır.
2. Negatif Küme Genişletme Adımı
- Bu adım tekrarlamalı bir adımdır. Adım sayısı i ile gösterilir. - Bu adımdaki sınıflandırıcı SVM sınıflandırıcısıdır. - Sınıflandırıcı f(i), P1 ve N(1) kullanılarak eğitilir. - U_{yeni}, f(i) kullanılarak sınıflandırılır. - Bu sınıflandırmadan bir sonraki adımın negatif kümesi N(i + 1) oluşur. - U_{yeni} kümesinden N(i + 1) kümesi çıkarılır. - U_{yeni} kümesinin boyutu P1 kümesinden büyükse tekrarlama devam eder.
3. Nihai Sınıflandırıcı ve Nihai Negatif Küme Seçimi
- 2. adımdaki tekrarlarda oluşturulan sınıflandırıcılar arasından tahmin isabet oranı en yüksek olanı en iyi sınıflandırıcı olarak seçilir. - En iyi sınıflandırıcının oluşturduğu adımda oluşturulan negatif küme nihai negatif küme (NN) olarak seçilir.
4. Sınıflandırma Adımı
- U kümesi, P ve NN kullanılarak sınıflandırılarak sonuçlar alınır. - Sınıflandırma işlemi adım 2'deki gibidir. (SVM sınıflandırıcısı)

2.3 Roc-SVM

Roc-SVM (Rocchio Technique and SVM), Li X. ve Liu B. tarafından geliştirilmiş bir sınıflandırma algoritmasıdır. Algoritma Rocchio Method [19] adı verilen bir metodu kullanmaktadır. Rocchio metodu temelde yapılan sınıflandırma işlemindeki doğruluk ve keskinlik değerlerini arttırmayı hedefler. Bu artırımın mümkün olması için sınıflandırmayı yapacak olan algoritmanın ya da kişinin elindeki örnekler konusunda belirli bir bilgisi olduğunu varsayarak örnek kümesine bu doğrultuda örnekler ekler.

Roc-SVM'de de AGPS'deki gibi pozitif örnekler P kümesinde, negatif örnekler de U kümesinde bulunmaktadır. Çapraz sağlama safhası AGPS'de olduğu gibi 10 katlı olarak yapılmaktadır. AGPS'den farklı olarak ön hazırlık aşamasında P kümesi değil U kümesi çapraz sağlamaya alınmaktadır. U kümesi her çapraz sağlama tekrarında

10 parçaya bölünecek ve bu parçalar tekrarlarda sırayla kullanılacaktır. Bölünmüş U kümesi parçalarına U_{evo} denecektir. U_{evo} kümesi algoritmanın son kısmında kullanılacaktır.

Roc-SVM algoritmasının ön hazırlık aşaması bittikten sonraki ilk adımdaki amaç güçlü negatifleri tespit etmektir. AGPS'nin son adımında yapılan bu işlem Roc-SVM'de başta yapılmaktadır. Güçlü negatifleri tespit aşamasında U kümesinin tamamen negatif örneklerden oluştuğu farz edilecektir. P ve U kümeleri için C(P) ve C(U) şeklinde iki adet prototip vektör oluşturulduktan sonra bu vektörlerin U kümesinde bulunan örneklere olan benzerliği hesaplanacaktır. C(U) vektörüne daha yakın olan örnekler GN (güçlü negatifler) kümesine aktarılacaktır ve bu örnekler U kümesinden çıkarılacaktır. U kümesinin yeni hali U_{yeni} olarak adlandırılacaktır.

Güçlü negatiflerin tespit edildiği ilk adımdan sonra tekrarlamalı adım başlayacaktır. Her tekrarlamada (tekrarlamalar AGPS'deki gibi i olarak adlandırılacaktır) f(i) ismi verilecek sınıflandırıcı P kümesi ve GN kümesi kullanılarak SVM tarafından oluşturulacaktır. Bu işlemden sonra U_{yeni} kümesi f(i) tarafından sınıflandırılacak ve negatif ve pozitif sonuçlardan oluşan bir çıktı elde edilecektir. Bu çıktıdaki negatif değerler N(i) ismi verilen bir kümede toplanacaktır. N(i) kümesine eklenen değerler U_{yeni} kümesinden çıkarılacaktır ve aynı değerler GN kümesine de eklenecektir. Tekrarlamalı adım, k adet tekrarlama sonucunda üretilen N(k) kümesi boş küme olana kadar, yani sınıflandırma aşamasında negatif bir sonuç üretilemeyene kadar devam edecektir. Tekrarlamalı adımın son tekrarlamasında üretilmiş olan sınıflandırıcı f(k), f(son) olarak adlandırılacaktır.

Bu adımdan sonra, nihai bir sınıflandırıcının seçimi yapılacaktır. Bunun için f(son) sınıflandırıcısı P kümesini sınıflandırmak için kullanılacaktır. Eğer yapılan bu sınıflandırmada, yüzde beş veya daha fazla oranda negatif sonuç elde edilmişse (yani öncesinde pozitif olduğu farz edilen örnek kümesinden belli bir oranın üstünde negatif örnek çıktığı görülürse) f(son) sınıflandırıcısı başarısız olarak görülecek ve f(son) yerine tekrarlamalı adımda üretilen ilk sınıflandırıcı olan f(1) nihai sınıflandırıcı olarak seçilecektir. Nihai sınıflandırıcı seçimine göre yapılacak en son etiketleme de farklı olacaktır. Eğer seçilen nihai sınıflandırıcı f(1) ise U_{yeni} kümesi

sınıflandırılacaktır. Nihai sınıflandırıcının f(son) olması durumunda ise P kümesi sınıflandırılacaktır. Sınıflandırma işleminden sonra etiketleme yapılacak ve algoritma sona erecektir.

Roc-SVM algoritmasının kısa kodu çizelge 2.2'de görülebilir.

Çizelge 2.2 Roc-SVM Algoritması Kısa Kodu

Kısaltmalar: U: etiketsiz örnekler kümesi; P: pozitif örnekler kümesi; P1,2: P kümesinin alt kümeleri; N: negatif örnekler kümesi; GN: güçlü negatifler kümesi f: sınıflandırıcı
0. Hazırlık Adımı
- Rocchio metodu kullanılarak güçlü negatifler kümesi (GN) oluşturulur. - U_{yeni} = U - GN olarak belirlenir. - U kümesi çapraz sağlama sebebiyle çapraz sağlama sayısına bölünür. Bu bölüm U_{evo} olarak adlandırılır. - Algoritma temelde çapraz sağlama sayısı kadar tekrarlayacaktır.
1. Sınıflandırıcı eğitme adımı
- Bu adım tekrarlamalı bir adımdır. - Sınıflandırıcı f(i), P ve GN kullanılacak eğitilir. - Bu adımdaki sınıflandırıcı SVM sınıflandırıcısıdır. - Bu adımın ilk çalıştığı zaman üretilen sınıflandırıcı f(1) olarak adlandırılır ve ileride kullanılmak üzere saklanır.
2. Sınıflandırma adımı
- Bu adım 1. adımın devamıdır. Aynı tekrarlamalı döngü içinde yer alır. - U_{yeni} kümesi, f(i) kullanılarak sınıflandırılır. - Bu adımdaki sınıflandırıcı SVM sınıflandırıcısıdır. - Sınıflandırmada ortaya çıkan negatif sonuçlar N(i) olarak adlandırılır.
3. Negatif Küme Kontrol Adımı
- Bu adım 2. Adımın devamıdır. Aynı tekrarlamalı döngü içinde yer alır. - Önceki adımda üretilen N(i) kümesi boş ise tekrarlama durdurulur ve 4. adıma geçilir. - N(i) kümesi boş değilse U_{yeni} kümesinden N(i) kümesi çıkarılır. - 1. adıma geri dönülerek tekrarlama işlemine devam edilir.
4. Son Sınıflandırıcı Üretimi Adımı
- Tekrarlamalı adımda en son üretilen sınıflandırıcı f(son) olarak adlandırılır. - P kümesi f(son) kullanılarak sınıflandırılır.
5. Sınıflandırıcı Seçme Adımı

- 4. adımda yapılan sınıflandırma sonuçlarındaki negatif sonuç sayısı tüm örnek sayısının yüzde 5'inden fazla ise sınıflandırıcı başarısız bulunur ve nihai sınıflandırıcı olarak 1. adımda üretilen f(1) seçilir. - Eğer 4. Adımda yapılan sınıflandırmanın sonuçlarındaki negatif sonuç sayısı tüm örnek sayısının yüzde 5'inden az ise sınıflandırıcı başarılı bulunur ve nihai sınıflandırıcı olarak 4. adımdaki f(son) seçilir.
6. Nihai Sınıflandırma Adımı
- 0. adımda üretilen U_{evo} kümesi, 5. adımda seçilen nihai sınıflandırıcı ile sınıflandırılır. - Bu adımdaki sınıflandırıcı SVM sınıflandırıcısıdır.

3 ÖNERİLEN ALGORİTMALAR

Bu bölüme, bölüm 2.1 ve 2.2'de anlatılan AGPS ve Roc-SVM algoritmalarına yapılacak iyileştirme çalışmalarından detaylı olarak bahsedilecektir. Öncelikle Random Forest metodunun SVM yerine kullanılması detaylandırılacak, devamında da Karma algoritma anlatılacaktır.

3.1 Random Forest

Random Forest iyileştirmesi algoritmaların sınıflandırıcı üretimi ve sınıflandırma aşamalarında kullanılan SVM metodunu Random Forest metodu ile değiştirmeyi amaçlamaktadır. Bu yaklaşım doğrultusunda değiştirilen AGPS ve Roc-SVM algoritmaları AGPS-RF ve Roc-RF olarak adlandırılacaktır.

Random Forest metodu 2001 yılında Leo Breiman [14] tarafından geliştirilmiştir. Random Forest metoduna toplama (ensemble) bir öğrenme metodu da denebilir. Toplama öğrenme, tek seferde birçok sınıflandırıcı üreten ve bu sınıflandırıcılar tarafından oluşturulan sonuçları oylama, kümeleme gibi farklı şekillerde birleştiren bir öğrenme yöntemidir. Random Forest metodu da bu amaç doğrultusunda Bagging (Bootstrap Aggregation) [20] adı verilen bir sistemi kullanmaktadır. Bagging, isabetlilik oranını arttırmak için yapılan iyileştirme çalışmaları anlamına gelen Bootstrap kelimesi ile, elde edilen sonuçları belirli bir şekilde birleştirme anlamına gelen Aggregation kelimesinin birleşimidir.

Leo Breiman tarafından 1994 yılında geliştirilen [20] Bagging sisteminde D adı verilen n boyutunda bir eğitim kümesi, D_i adında, m adet n' boyutunda alt sınıflandırıcılara ayrılır. Oluşturulan m adet eğitim kümesinin oluşturduğu sonuçlar oylama veya sonuçları ortalama gibi yöntemlerle birleştirilir. Bagging sistemi temelde algoritmanın isabetlilik oranını arttırmak ve sapmayı düşürmek için kullanılmaktadır.

Random Forest algoritması eldeki N adı verilen bir veri kümesine önceki paragrafta bahsedilen Bagging işlemini uygular. Oluşturulan her eğitim kümesi için budanmamış (unpruned) bir karar ağacı (decision tree) oluşturulur. Oluşturulacak karar ağaçları Leo Breiman tarafından 1984 yılında geliştirilen CART (Classification and Regression Tree) [21] modelindedir. Karar ağacının budanması kavramı, ağaçtaki fazla genellenmiş sayılabilecek bölümlerin ağaçtan atılarak isabet oranını arttırma ve hata payını azaltma işlemidir.

Budanmamış karar ağacı oluşumu aşamasından sonra her karar ağacı için rastgele bir şekilde m_{try} adı verilen tahmin ediciler (predictor) seçilir. Tahmin ediciler, bir örneğin başka bir örnekle olan ilişkisini kurmaya yarayan fonksiyonlardır. Tahmin edicilerin rastgele bir şekilde seçilmesi Bagging işlemi ile beraber algoritmaya fazladan rastlantısallık ekleyen bir özelliktir. Seçilen tahmin ediciler karar ağacında hangi yöne kırılım yapılacağını belirleyecektir. Nihai olarak N kümesinin elemanlarından oluşturulan alt kümeler tarafından yaratılan tüm ağaçlarda bu işlem tekrarlanacak ve her ağaçtan gelecek sonuç oylama usulüyle seçilecektir.

3.1.1 Random Forest Kullanımının Sebepleri ve Amaçları

AGPS ve Roc-SVM algoritmalarını geliştirme ve iyileştirme işlemlerinin ilki olan sınıflandırmada SVM yerine Random Forest metodunu kullanmanın en önemli sebeplerinden biri Random Forest metodunda kullanılan toplama sınıflandırıcı (Ensemble Classifier) [22] yapısıdır.

AGPS ve Roc-SVM algoritmalarında oluşturulan sınıflandırıcılar tekrarlamalı aşamalarla sınırlıdır ve Random Forest metodunda var olan şekilde bu sınıflandırıcıların sonuçlarını oylama veya birleştirme ile asıl sonuca ulaşmak gibi bir işlem yapılmamaktadır. AGPS ve Roc-SVM algoritmalarında Random Forest metodunun tam tersine tek bir nihai sınıflandırıcı vardır ve üretilen diğer sınıflandırıcılar nihai karar aşamasında kullanılmamaktadır.

Random Forest metodunu kullanmadan AGPS ve Roc-SVM algoritmalarında toplama sınıflandırıcı kullanma fikri, tekrarlamalı aşamalarda üretilen sınıflandırıcıların boyutlarının çok büyük olması ve bu sebeple sağlıklı bir sınıflandırma yapılamaması ve sonuç alınamamasından ötürü uygulanabilir değildir.

Algoritmalarda SVM yerine Random Forest metodunun kullanılmasının bir diğer sebebi de Random Forest metodunun mikrodizi (microarray) tipindeki verilerde isabetlilik, performans, hata oranı gibi kalemlerde SVM ile başa baş ve kimi zaman daha iyi performans verdiğinin görülmesidir. Bu konuda J. Nappi, D. Regge ve H. Yoshida'nın yaptığı araştırmanın [23] yanı sıra Y. Tang, S. Krasser, Y. He, W. Yang ve D. Alperovitch [24] ve G. Rios ve H. Zha'nın yaptıkları [25] araştırmalar yol gösterici olmuştur.

3.1.2 Random Forest metodunun Algoritmalara Uygulanması

SVM kullanan AGPS ve Roc-SVM algoritmalarının sınıflandırıcı olarak Random Forest kullanmaya uygun hale getirilmesi temelde sınıflandırıcının değiştirilmesi ve kullanılan verilerin Random Forest sınıflandırıcısına uygun hale getirilmesi ile başarılmıştır.

Random Forest sınıflandırıcısı kullanan AGPS-RF ve Roc-RF algoritmalarının kısa kodları Çizelge 3.1 ve Çizelge 3.2'de görülebilir.

Çizelge 3.1 AGPS-RF Algoritması Kısa Kodu

Kısaltmalar: U: etiketsiz örnekler kümesi; P: pozitif örnekler kümesi; P1,2: P kümesinin alt kümeleri; N: negatif örnekler kümesi; f: sınıflandırıcı
0. Çapraz Sağlama Adımı

- U_{yeni} = U + P2 (Çapraz doğrulama işlemi için P kümesinin bir kısmı alınmaktadır) olarak belirlenir. - Bu adım algoritmanın kalanını kapsayarak çapraz sağlama sayısı kadar tekrarlayacaktır. Amacı U kümesini sınıflandırırken kullanılacak örnekler arasında homojen bir dağılım yapmaktır. Detaylı bilgi için 2.1 bölümüne bakılabilir.
1. İlk Negatif Küme Oluşturma
- Sınıflandırıcı f(1), P1 ve U_{yeni} kullanılacak oluşturulur. - Bu adımdaki sınıflandırıcı SVM sınıflandırıcısıdır. - U_{yeni}, f(1) kullanılarak sınıflandırılır. Bu sınıflandırmadan N(1) negatif kümesi elde edilir. Bu küme 2. adımda kullanılacaktır. - U_{yeni} kümesinden N(1) kümesindeki elemanlar çıkarılır.
2. Negatif Küme Genişletme Adımı
- Bu adım tekrarlamalı bir adımdır. Adım sayısı i ile gösterilir. - Bu adımdaki sınıflandırıcı Random Forest sınıflandırıcısıdır. - Random Forest örneği I(i), P1 ve N(1) kullanılarak eğitilir. - Sınıflandırıcı ağaç T(i), I(i) kullanılarak oluşturulur. - U_{yeni}, T(i) kullanılarak sınıflandırılır. - Bu sınıflandırmadan bir sonraki adımın negatif kümesi N(i + 1) oluşur. - U_{yeni} kümesinden N(i + 1) kümesi çıkarılır. - U_{yeni} kümesinin boyutu P1 kümesinden büyükse tekrarlama devam eder.
3. Nihai Sınıflandırıcı ve Nihai Negatif Küme Seçimi
- 2. adımdaki tekrarlarda oluşturulan sınıflandırıcılar arasından tahmin isabet oranı en yüksek olanı en iyi sınıflandırıcı olarak seçilir. - En iyi sınıflandırıcının oluşturduğu adımda oluşturulan negatif küme nihai negatif küme (NN) olarak seçilir.
4. Sınıflandırma Adımı
- U kümesi, P ve NN kullanılarak sınıflandırılarak sonuçlar alınır. - Sınıflandırma işlemi adım 2'deki gibidir. (Random Forest sınıflandırıcısı)

Çizelge 3.2 Roc-RF Algoritması Kısa Kodu

Kısaltmalar: U: etiketsiz örnekler kümesi; P: pozitif örnekler kümesi; P1,2: P kümesinin alt kümeleri; N: negatif örnekler kümesi; GN: güçlü negatifler kümesi f: sınıflandırıcı
0. Hazırlık Adımı

- Rocchio metodu kullanılarak güçlü negatifler kümesi (GN) oluşturulur. - U_{yeni} = U - GN olarak belirlenir. - U kümesi çapraz sağlama sebebiyle çapraz sağlama sayısına bölünür. Bu bölüm U_{evo} olarak adlandırılır. - Algoritma temelde çapraz sağlama sayısı kadar tekrarlayacaktır.
1. Sınıflandırıcı eğitme adımı
- Bu adım tekrarlamalı bir adımdır. - Sınıflandırıcı f(i), P ve GN kullanılacak eğitilir. - Bu adımdaki sınıflandırıcı Random Forest sınıflandırıcısıdır. - Bu adımın ilk çalıştığı zaman üretilen sınıflandırıcı f(1) olarak adlandırılır ve ileride kullanılmak üzere saklanır.
2. Sınıflandırma adımı
- Bu adım 1. adımın devamıdır. Aynı tekrarlamalı döngü içinde yer alır. - U_{yeni} kümesi, f(i) kullanılarak sınıflandırılır. - Sınıflandırmada ortaya çıkan negatif sonuçlar N(i) olarak adlandırılır.
3. Negatif Küme Kontrol Adımı
- Bu adım 2. Adımın devamıdır. Aynı tekrarlamalı döngü içinde yer alır. - Önceki adımda üretilen N(i) kümesi boş ise tekrarlama durdurulur ve 4. adıma geçilir. - N(i) kümesi boş değilse U_{yeni} kümesinden N(i) kümesi çıkarılır. - 1. adıma geri dönülerek tekrarlama işlemine devam edilir.
4. Son Sınıflandırıcı Üretimi Adımı
- Tekrarlamalı adımda en son üretilen sınıflandırıcı f(son) olarak adlandırılır. - P kümesi f(son) kullanılarak sınıflandırılır.
5. Sınıflandırıcı Seçme Adımı
- 4. adımda yapılan sınıflandırma sonuçlarındaki negatif sonuç sayısı tüm örnek sayısının yüzde 5'inden fazla ise sınıflandırıcı başarısız bulunur ve nihai sınıflandırıcı olarak 1. adımda üretilen f(1) seçilir. - Eğer 4. Adımda yapılan sınıflandırmanın sonuçlarındaki negatif sonuç sayısı tüm örnek sayısının yüzde 5'inden az ise sınıflandırıcı başarılı bulunur ve nihai sınıflandırıcı olarak 4. adımdaki f(son) seçilir.
6. Nihai Sınıflandırma Adımı
- 0. adımda üretilen U_{evo} kümesi, 5. adımda seçilen nihai sınıflandırıcı ile sınıflandırılır.

3.2 Karma Algoritma

Karma Algoritma AGPS ve Roc-SVM algoritmalarının birleşimi olarak düşünülmüştür. İki algoritmanın da aynı veri kümesinde oluşturduğu sonuçlar

geliştirilecek bir oylama sistemi ile karşılaştırılacak ve yapılacak oylama sonucunda nihai sonuçlar elde edilecektir.

3.2.1 Karma Algoritma Kullanımının Sebepleri ve Amaçları

Karma Algoritmanın kullanımının amaçları Random Forest metodunun algoritmalara uygulanma sebebi ile benzerlik göstermektedir. Karma Algoritma metodunda da Random Forest metoduna benzer şekilde nihai sonuçların temini için bir oylama yapılması söz konusudur. Bölüm 3.1.1'de bahsedildiği gibi AGPS ve Roc-SVM mevcut halleri ile toplama sınıflandırıcı kullanmak için uygun algoritmalar değildir. Bu sebeple sınıflandırıcı sayısını arttırmadan oylama sistemini geliştirilmiştir.

3.2.2 Karma Algoritmanın Uygulanması

Karma Algoritma temelde algoritmaların yapısını korurken iki algoritmayı da aynı anda çalıştırmaktadır. İki algoritma da aynı veri grupları ve aynı çapraz sağlama kümeleri ile çalışacaktır. Sınıflandırıcı üretimi ve sınıflandırma işlemleri AGPS ve Roc-SVM algoritmalarında olduğu gibi SVM tarafından yapılacaktır. İki algoritmanın da tekrarlamalı ve nihai adımları bitip iki algoritma da veri kümesini tamamen etiketledikten sonra sonuçları oylama aşamasına geçilecektir.

Karma Algoritmada sınıflandırıcı olarak AGPS-RF ve Roc-RF algoritmalarında kullanılan Random Forest yerine SVM kullanılmasının sebebi Random Forest sınıflandırıcısının yapısı gereği toplama bir sınıflandırıcı olması ve oluşturduğu alt sınıflandırma ağaçlarını bir oylamaya tabi tutmasıydı. Önceki cümlede bahsedilen sebepten dolayı Karma Algoritmada SVM sınıflandırıcısı kullanımına devam edilerek bu oylamayı sınıflandırıcının kendi iç döngüsünden bağımsız bir şekilde yapması amaçlanmıştır.

Sonuçları oylama aşamasında etiketsiz veriler kümesindeki her sonuç sırayla karşılaştırılacaktır. Eğer bir veri için iki algoritma da aynı sonucu vermişse (örneğin iki algoritmada da sonuç Gerçek Pozitif olarak etiketlenmişse) sonuç oylamayı geçerek nihai sonuçlar kümesine eklenecektir. Bir veri için iki algoritmanın da farklı sonuçlar verdiği durumlarda o veri için algoritmaların sınıflandırma sırasında verdiği çıktıya bakılacaktır. SVM tarafından üretilen çıktıda sınıflandırılan bir verinin etiketlenen sonuca ne kadar yakın olduğu olasılık değeri üzerinden gösterilmektedir. Oylama aşamasında bir veride sonuç farkı olması durumunda iki algoritmanın da o veri için hangi olasılık değerine göre bu sınıflandırmayı yaptığı karşılaştırılacaktır. Yapılan karşılaştırmada sonuçların olasılık ölçeğinden ne kadar uzakta olduğuna bakılacaktır. Örneğin, bir sonuç AGPS algoritması tarafından Gerçek Negatif, Roc-SVM algoritması tarafından ise Yanlış Pozitif olarak etiketlenmiş durumda ve AGPS algoritmasındaki sonuç çıktısında olasılık değeri 0.76 (değer 0.5'den yüksek olduğundan dolayı gerçek (true) bir sonuç olarak görülmektedir), Roc-SVM algoritmasındaki sonuç çıktısında ise olasılık değeri 0.44 (değer 0.5'den düşük olduğundan dolayı yanlış (false) bir sonuç olarak görülmektedir) ise iki olasılık değerinin orta nokta olan 0.5'e olan uzaklığına bakılacaktır. Bu durumda AGPS algoritmasının verdiği 0.76 değeri 0.5 değerine 0.26 uzaklıktayken Roc-SVM algoritmasının verdiği 0.44 değeri 0.5 değerine 0.6 uzaktır. Bu durumda AGPS algoritmasının verdiği sonuç daha güçlü bir sonuç olarak alınır ve yapılan oylamada AGPS algoritmasının verdiği sonuç nihai sonuçlar listesine eklenir.

Oylama etiketlenen tüm verilen için yapılıp nihai sonuçlar listesi doldurulduğunda Karma Algoritma da bitmiş olacaktır.

4 DENEY SONUÇLARI

Bu bölümde geliştirilen algoritmalar ile yapılan denemeler ile ilgili detaylar ve yapılan denemelerin karşılaştırmalı sonuçları ile ilgili bilgi verilecektir.

4.1 Deney Ortamı

Deneyde kullanılan veri kümesi J. J. Faith tarafından derlenen [26] E. Coli (Escherichia Coli) gen çıkarımı (Gene Expression, Protein ve RNA gibi yapıtaşlarını üretmeye yarayan gen dizilimidir) veri kümesidir. Veri kümesinde 4345 adet gen ve her gen için 445 adet mikrodizi örneklemesi bulunmaktadır. Mikrodizi veri yapısı, söz konusu proteinlere tek tek değil bir bütün olarak bakmaya olanak sağlayan, etiketsiz örnekleri sınıflandırmak için uygun bir yapı olarak öne çıkmaktadır. Mikrodiziler algoritmalar kodlanırken vektör veri yapıları olarak aktarılmıştır.

E. Coli veri kümesine ek olarak bu veriler arasında etkileşimde olan proteinleri tespit etmek için (Pozitif kümeyi oluşturmak için) IntAct [27] adındaki proteinler arası etkileşim veri tabanı kullanılmıştır. Bu veri kümesi M. Tan ve C. Kılıç tarafından yapılan [1] pozitif etiketsiz öğrenme algoritmalarını karşılaştırma çalışmasında da kullanıldığı için bu çalışmada yapılan iyileştirmeler ve algoritmaların eski halleri sağlıklı bir şekilde karşılaştırılabilecektir.

Kullanılan E. Coli veri kümesinden 27 adet alt küme çıkarılmıştır. Buradaki alt küme çıkarımı algoritmalardaki çapraz sağlama aşamasına benzetilebilir. Farklı P ve Q değerleri ile yapılan testler ile isabet oranı artırılırken sapmanın düşürülmesi amaçlanmaktadır. Aynı zamanda veri alt kümelerinde P ve Q kümelerinin eleman sayıları toplamı 250'yi geçmemektedir. 27 adet veri kümesinin yaratımında r ismi verilen oran kullanılmaktadır. r oranı $r = P / (P + Q)$ formülü ile bulunmaktadır. P ve Q kümelerinin boyutları r değeri onarlı katlar haline %10 ile %90 arasında dokuz farklı sonuçta çıkacak şekilde ayarlanır. Dokuz adet r değeri için her biri rastgele

seçilmiş 3 alt küme kullanılmaktadır. Bu şekilde toplam alt küme sayısı 27 olur. Bu dağılım ile pozitif ve etiketsiz örnek kümelerinin farklı oranlarda karışması sağlanmış ve rastgele bir dağılım da eklenerek örneklerin karışımı zenginleştirilmiştir.

AGPS ve Roc-SVM algoritmaları, M. Tan ve C. Kılıç'ın yaptığı pozitif etiketsiz öğrenme algoritmaları testinde kullanılmak üzere teste katılan diğer algoritmalar ile beraber Java dilinde kodlanmıştır. Her iki algoritmada da sınıflandırıcı üretimi ve sınıflandırma işlemlerinde kullanılan SVM metodu, C-C. Chang ve C-J. Lin Tarafından yaratılan LibSVM kütüphanesi [28] yardımı ile kullanılmıştır.

Algoritmaları geliştirme aşamasında kullanılacak Random Forest metodunun kullanımı için Waikato Üniversitesinin WEKA [29] (Waikato Environment for Knowledge Analysis) adı verilen makine öğrenme kütüphanesi kullanılmıştır. Kütüphanede makine öğrenme alanında kullanılan birçok algoritma mevcuttur. Random Forest metodunu kullanarak algoritmaları iyileştirme işleminde de WEKA kütüphanesinin Random Forest bölümü kullanılmıştır. Kullanılan Random Forest sınıflandırıcısında üretilen ağaç sayısı ve seçilen özellik sayısı gibi parametrelerde varsayılan değerler kullanılmıştır. Yapılan denemelerde varsayılan değerlerin haricinde iki farklı parametre grubu ile (ağaç sayısı ve derinlik olarak sırasıyla 100;50 ve 200;100) yapılan denemelerde alınan sonuçlar varsayılan değerler ile dikkate değer bir farka yol açmamıştır.

Random Forest metodunun uygulanması sırasında karşılaşılan sorunlardan biri AGPR ve Roc-SVM algoritmalarında sınıflandırıcı, eğitim kümesi, etiketsiz küme ve benzeri kümelerin metin dosyalarına aktarılmış hallerinin LibSVM kütüphanesinin anlayacağı bir yapıda olması, fakat WEKA kütüphanesinin Random Forest kolu ile uyumsuz olmasıydı. Bu sebeple algoritmada sıkça kullanılan bu metin dosyalarını algoritmanın çalışma süresi içinde WEKA Random Forest kütüphanesi ile uyumlu bir hale getirildi ve algoritmanın kalan bölümlerinde de algoritmanın temel yapısını bozmamak amacıyla eski hallerine döndürüldü.

AGPS, Roc-SVM, AGPS-RF, Roc-RF, Karma Algoritma, CLR ve ARACNE algoritmaları Windows 7 (64 bit) işletim sistemine sahip, Intel Core i7 2.2 GHz

işlemcili, 6 GB RAM'e sahip olan bir bilgisayarda test edildi. Yapılan testlerde AGPS-RF ve Roc-RF algoritmalarının ortalama çalışma sürelerinin 7 ile 8 saat arasında, Karma Algoritmanın ortalama çalışma süresinin ise 11 ile 12 saat arasında olduğu gözlemlendi. Algoritmalar Eclipse entegre geliştirme ortamının Juno (4.2) [30] sürümünde çalıştırıldı.

4.2 AGPS-RF ve Roc-RF Sonuçları

Eski ve yeni algoritmalarla aynı veri kümesi kullanılarak yapılan testlerde hem AGPS-RF hem de Roc-RF algoritmalarının SVM kullanan AGPS ve Roc-SVM algoritmalarına göre isabetlilik ve hata oranının düşüklüğü açısından daha iyi sonuçlar verdiği görüldü. Eski ve yeni algoritmaların oluşturduğu sonuçlar arasındaki fark, AGPS ve AGPS-RF arasında çok yüksek olarak gözlenmezken Roc-SVM ve Roc-RF algoritmaları arasında oldukça fazla olarak gözlendi. Bu durumun sebebi olarak Roc-RF algoritmasının tekrarlamalı yapısının Random Forest metodundaki tekrarlamalı ve eğitim kümesini bölen Bagging işlemine daha yakın olması gösterilebilir. Roc-SVM ve Roc-RF algoritmalarında sınıflandırma ve etiketleme işlemleri tekrarlamalı bir biçimde yapılırken AGPS ve AGPS-RF algoritmalarında sınıflandırma ve etiketleme işlemlerinin tüm tekrarlamalar bitince yapılıyor olması da eski ve yeni algoritmalar arasında büyük bir fark olmamasına neden olmuş olarak gösterilebilmektedir.

SVM kullanan AGPS algoritmasında 27 alt kümeye ayrılmış veri kümesinin ortalama f-ölçüsü değeri 0,212 çıkarken, Random Forest kullanan AGPS-RF algoritmasının 27 alt kümede verdiği sonuçların ortalaması 0,214 çıkmıştır. Bu sonuçlara göre AGPS-RF algoritması f-ölçüsü bakımından AGPS algoritmasına göre binde sekiz gibi bir iyileşme göstermiştir. AGPS ve AGPS-RF algoritmalarının MCC değerleri sırasıyla 0,133 ve 0,130 çıkmış, AGPS-RF algoritmasının AGPS algoritmasına göre MCC değeri açısından yüzde 2,2 oranında daha az performanslı olduğu görülmüştür.

Roc-RF algoritmasında ise bir önceki paragrafta bahsedildiği gibi oluşan fark ve iyileştirme çok daha fazla olarak gözlenmiştir. 27 alt kümeye ayrılmış veri kümesinin ortalama f-ölçüsü değeri SVM kullanan Roc-SVM algoritmasında 0,226 çıkarken Roc-RF algoritmasında 27 alt kümenin ortalama f-ölçüsü değeri 0,593 çıkmıştır. Bu iki sonuca göre Roc-RF algoritması f-ölçüsü açısından Roc-SVM algoritmasına göre yüzde 159'luk bir iyileşme göstermiştir. Roc-SVM ve Roc-RF algoritmalarının MCC değerleri sırasıyla 0,261 ve 0,567 çıkmış, Roc-RF algoritmasının Roc-SVM algoritmasına göre MCC değeri açısından yüzde 53,9 oranında daha performanslı olduğu görülmüştür.

Çizelge 4.1'de AGPS-RF ve Roc-RF algoritmalarının f-ölçüsü ve MCC değerleri AGPS ve Roc-SVM ile karşılaştırmalı olarak görülebilir. Daha detaylı bir sonuç tablosu Ek B bölümünde Çizelge B.1 ve Çizelge B.2'de bulunabilir.

Çizelge 4.1 AGPS-RF ve Roc-RF Algoritmalarının F-Ölçüsü ve MCC Değerleri Üzerinden Karşılaştırmalı Sonuçları

Ölçüm / Algoritma	AGPS	AGPS-RF	Roc-SVM	Roc-RF
F-Ölçüsü	0,212	0,214	0,226	0,593
MCC	0,133	0,130	0,261	0,567

4.3 Karma Algoritma Sonuçları

Karma algoritma, AGPS-RF ve Roc-RF algoritmalarında olduğu gibi 27 alt kümeye ayrılmış olan veri kümesinde denenmiştir. Bu denemelerde Karma Algoritma, karşılaştırılacağı AGPS ve Roc-SVM algoritmalarına göre f-ölçüsü değerleri açısından daha iyi sonuç vermiştir.

AGPS ve Roc-SVM algoritmalarının sırasıyla 0,212 ve 0,226 olan f-ölçüsü değerlerine karşılık olarak 0,515 f-ölçüsü değerine sahip olan karma algoritma, bu iki algoritmaya nazaran sırasıyla yüzde 140 ve yüzde 125 oranlarında daha isabetli sonuçlar vermiştir. Denemelerdeki f-ölçüsü değeri önceki algoritmalarda olduğu gibi 27 alt kümeden elde edilen f-ölçüsü değerlerinin ortalamasıdır. Yapılan denemelerde MCC değeri açısından Karma Algoritmanın 0,294 sonucunu vererek AGPS ve Roc-SVM algoritmalarına göre sırasıyla yüzde 12,6 daha yüksek ve yüzde 48,1 daha az performans verdiği görülmüştür. Roc-RF algoritmasında benzer sonuçlar veren f-ölçüsü ve MCC değerlerinin Karma Algoritmada daha farklı olmasının sebebi sonuçlardaki gerçek negatif değerler arasındaki fark ile açıklanabilir, çünkü f-ölçüsü değeri gerçek negatif sonuçları hesaba katmamaktadır.

Karma algoritmanın AGPS, Roc-SVM, AGPS-RF ve Roc-RF algoritmaları ile f-ölçüsü ve MCC üzerinden karşılaştırması Çizelge 4.2'de görülebilir. Daha detaylı bir sonuç tablosu Ek B bölümünde Çizelge B.3 ve Çizelge B.4'de bulunabilir.

Çizelge 4.2 Karma Algoritmanın F-Ölçüsü ve MCC Değerleri Üzerinden Karşılaştırmalı Sonuçları

Ölçüm / Algoritma	AGPS	AGPS-RF	Roc-SVM	Roc-RF	Karma Algoritma
F-Ölçüsü	0,212	0,214	0,226	0,593	0,515
MCC	0,133	0,130	0,261	0,567	0,294

4.4 Geliştirilen Algoritmaların diğer Bilinen Algoritmalar İle Karşılaştırılması

Önceki bölümlerde AGPS-RF, Roc-RF ve Karma algoritmaların temelini aldıkları AGPS ve Roc-SVM algoritmaları ile yapılan karşılaştırmalara ve alınan sonuçlara yer verildi. Öne sürülen bu üç geliştirilmiş algoritma, özünde bir biyolojik ağ

türettiğinden bu algoritmaların iyileşme seviyelerini denetlemek amacıyla bilinen başka ağ çıkarımı (network inference) algoritmaları [31] ile de karşılaştırma yapılması zorunluluğu doğmuştur.

Bu bölümde, CLR ve ARACNE olarak seçilen iki adet ağ çıkarımı algoritması ile bu çalışmada geliştirilen AGPS-RF, Roc-RF ve Karma algoritmalar karşılaştırılacaktır. Her ne kadar CLR ve ARACNE gözetimli öğrenme algoritmaları olmasalar da bu karşılaştırma ile algoritmaların göreceli performansları konusunda bir fikir sahibi olunabilecektir.

4.4.1 CLR

CLR (Context Likelihood of Relatedness), J. J. Faith v.d. tarafından geliştirilen bir algoritmadır [15]. CLR, ilgi ağı algoritmaları (Relevance Network Algorithms) [32] için yeni bir eklenti olarak tanımlanmaktadır. Temel ilgi ağı algoritma anlayışının üzerine arka planda çalışan bir yanlış doğrulama adımı ile hatalı bir şekilde tanımlanan ilişkileri düzeltecek bir sistem oturtulmuştur. CLR, tahminlerini oluşturduğu ağın içindeki her düğümün istatistiki olarak ilişkili olabilirliğini göz önüne alarak yapan bir algoritmadır. İki düğüm (protein çifti) arasındaki ilişkinin tanımlanması aşamasında bu olabilirlik oranı en yüksek olan çift, ilişkili olarak nitelendirilir.

CLR algoritması, çalışmadaki tüm deneylerde kullanılan ve detayları 4.1 bölümünde bulunabilen E. Coli veri kümesini ve bu veri kümesinden türetilen 27 alt kümeyi kullanarak denenmiştir.

4.4.2 ARACNE

ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks), M. Margolin v.d. tarafından geliştirilmiş bir algoritmadır [16]. Kullandığı yöntem önceki algoritmalarda olduğu gibi bir ilgi ağı kurarak protein çiftleri arasında ilişki olup olmadığını bu ağ üzerinden belirlemeye dayalıdır.

ARACNE de AGPS ve Roc-SVM algoritmaları gibi mikrodizi örneklemesi kullanmaktadır. ARACNE elindeki örnekleri sınıflandırmak için örnek çiftlerini tek tek ele alarak CLR algoritmasına benzer bir şekilde örnek çiftleri arasındaki istatistiki ilişki olabilirliğinin belirli bir seviyenin üzerinde olup olmadığına bakmaktadır. Belirli bir seviyenin üstündeki olasılıklar ilişkili olarak nitelendirilir.

ARACNE özellikle düşük hata oranı ile öne çıkan bir algoritmadır. ARACNE de çalışmadaki diğer tüm deneylerde olduğu gibi E. Coli veri kümesini ve bu kümeden türetilen 27 alt kümeyi kullanarak denenmiştir.

4.4.3 CLR ve ARACNE ile Geliştirilen Algoritmaların Karşılaştırma Sonuçları

Yapılan deneylerde E. Coli veri kümesinin örneklerinden oluşan 27 alt küme CLR algoritmasında denenmiştir. 27 alt kümenin ortalama f-ölçüsü değeri 0,401 olarak bulunmuştur. ARACNE ile yapılan deneyde aynı veri kümelerinin ortalama f-ölçüsü değeri ise 0,315 olarak bulunmuştur.

AGPS-RF algoritması ile 4.2 bölümünde yapılan denemelerde ortalama f-ölçüsü olarak 0,214 bulunmuştu. Bu sonuca göre CLR, AGPS-RF algoritmasına göre yüzde 46 oranında daha iyi bir performans sergilemiştir. ARACNE ise AGPS-RF algoritmasına göre yüzden 32 oranında daha iyi bir performans sergilemiştir. MCC değeri açısından ise AGPS-RF algoritması, CLR ve ARACNE algoritmalarına göre sırasıyla yüzde 189 ve yüzde 101,5 oranlarında daha az performans vermiştir.

Roc-RF algoritması ile 4.2 bölümünde yapılan denemelerde ortalama f-ölçüsü olarak 0,593 bulunmuştu. Bu sonuca göre Roc-RF algoritması CLR algoritmasına göre yüzde 33 oranında daha performanslı olurken ARACNE'ye göre ise yüzde 47 oranında daha performanslı olmuştur. MCC değeri açısından ise Roc-RF algoritması, CLR ve ARACNE algoritmalarına göre sırasıyla yüzde 33,6 ve yüzde 53,7 oranlarında daha yüksek performans vermiştir.

Karma algoritma ile 4.3 bölümünde yapılan denemelerde ortalama f-ölçüsü olarak 0,515 bulunmuştu. Bu değere göre Karma algoritma, CLR algoritmasına göre yüzde 22 oranında daha performanslı olmuştur. Karma algoritma ARACNE karşısında ise yüzde 39 oranında daha performanslı olmuştur. MCC değeri açısından ise Karma Algoritma, CLR ve ARACNE algoritmalarına göre sırasıyla yüzde 27,8 oranında daha az performans ve yüzde 10,8 oranında daha yüksek performans vermiştir.

AGPS-RF, Roc-RF ve Karma algoritmalarının CLR ve ARACNE ile yapılan karşılaştırmalarında AGPS-RF algoritmasının ortalama f-ölçüsü ve MCC değerleri açısından daha geride kaldığı gözlenmiştir. Yine aynı karşılaştırmalarda Roc-RF ve Karma algoritmanın ise ortalama f-ölçüsü değeri açısından CLR ve ARACNE'ye göre daha iyi olduğu gözlenmiştir. MCC değeri açısından Roc-RF her iki algoritmadan da daha iyi sonuçlar vermişken Karma Algoritma sadece ARACNE algoritmasından daha iyi sonuç vermiştir. AGPS-RF ve Roc-RF algoritmalarının bu iki bilinen algoritmaya karşı gösterdiği performans farkı aynı algoritmaların SVM kullanan eski hallerine (AGPS ve Roc-SVM) göre gösterdiği performans ile benzeşen bir görüntü çizmektedir. Roc-RF ve Karma Algoritmanın CLR ve ARACNE algoritmalarına karşı iyi sonuçlar vermesi, Roc-RF ve Karma Algoritmanın pozitif etiketsiz öğrenme konusunda özelleşmiş olmaları ve CLR ve ARACNE algoritmalarının daha genel ağ çıkarımı algoritmaları olmaları ile açıklanabilir.

AGPS-RF, Roc-RF ve Karma Algoritmanın CLR ve ARACNE ile f-ölçüsü ve MCC değeri üzerinden karşılaştırmalı sonuçlar Çizelge 4.3'de görülebilir. Daha detaylı sonuçlar Ek B bölümünde Çizelge B.5 ve Çizelge B.6'da bulunabilir.

Çizelge 4.3 CLR ve ARACNE Algoritmalarının F-Ölçüsü ve MCC Değerleri Üstünden Karşılaştırmalı Sonuçları

Ölçüm / Algoritma	AGPS-RF	Roc-RF	Karma Algoritma	CLR	ARACNE
F-Ölçüsü	0,214	0,593	0,515	0,401	0,315
MCC	0,130	0,567	0,294	0,376	0,262

5 SONUÇ

Yapılan çalışmada Random Forest metodunu kullanan AGPS-RF ve Roc-RF algoritmaları geliştirilmiştir. Bu iki algoritma, SVM kullanan eski halleri AGPS ve Roc-SVM algoritmalarına karşı farklı sonuçlar elde etmiştir. Yapılan denemelerde, AGPS-RF algoritması AGPS algoritmasına göre f-ölçüsü değeri açısından daha iyi sonuçlar elde etse de bu iyileştirme oldukça ufak bir oranda kalmıştır ve MCC değeri açısından da AGPS, AGPS-RF algoritmasına göre geride kalmıştır.

Öte yandan Roc-RF ile Roc-SVM algoritmaları arasında yapılan karşılaştırmalarda f-ölçüsü ve MCC değeri farkı Roc-RF lehine oldukça yüksek olarak gözlemlenmiştir.

AGPS ve Roc-SVM algoritmalarının birleştirilmesi ile oluşturulan Karma algoritma ise temel aldığı AGPS ve Roc-SVM algoritmalarına göre daha iyi bir sonuç vermiştir. MCC değeri sonucunda göre AGPS-RF algoritmasına göre daha iyi olan Karma Algoritma aynı değerde Roc-RF algoritmasının gerisinde kalmıştır. MCC değerinde oluşan bu farkın sebebi MCC ölçümünün hesaba gerçek negatif değerleri de katması ile açıklanabilir. AGPS-RF algoritmasının Karma ve Roc-RF algoritmaları seviyesinde bir performans verememesi ile ilgili sebepler 4.2 bölümünde detaylı olarak açıklanmıştır.

Geliştirilen üç algoritmanın CLR ve ARACNE olarak seçilen iki ağ çıkarımı algoritması ile karşılaştırılması kısmında da önceki paragrafa benzer sonuçlar alınmıştır. Yapılan deneylerde AGPS-RF algoritması CLR ve ARACNE algoritmalarına karşı f-ölçüsü ve MCC değerlerinde geri kalırken Roc-RF ve Karma algoritma bu iki algoritmaya karşı f-ölçüsü değerinde daha iyi bir sonuç vermiştir.

Deneylerden alınan sonuçlar incelendiğinde Roc-RF algoritmasının AGPS-RF algoritmasına nazaran daha iyi performans verdiği görülmektedir. Bu bağlamda Random Forest metodunun Roc-SVM algoritmasında daha performanslı çalıştığı da söylenebilir. Karma algoritma da yapılan deneylerde karşılaştırıldığı algoritmalara göre daha performanslı olsa da Roc-RF algoritmasının ortalama f-ölçüsü değeri daha yüksek çıkmıştır.

KAYNAKLAR

[1] Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D, Yeates TO. (1999) "Assigning protein functions by comparative genome analysis: protein phylogenetic profiles." Proc Natl Acad Sci U S A., 96, 4285-8

[2] Enright A.J.,Iliopoulos I.,Kyripides N.C. and Ouzounis C.A. (1999) "Protein interaction maps for complete genomes based on gene fusion events." Nature (402), 86-90

[3] Kılıç C, Mehmet Tan (2012) Positive unlabelled learning for deriving protein interaction networks. Netw Modeling Anal in Health Inform and Bioinform 1(3): 87–10

[4] Zhao X-M, Wang Y, Chen L, Aihara K (2008) Gene function prediction using labeled and unlabeled data. BMC Bioinformatics. 9:57

[5] Li X, Liu B (2003) Learning to classify texts using positive and unlabeled data. In: IJCAI'03: Proceedings of the 18th international joint conference on artificial intelligence (2003), pp. 587–592

[6] Wang C, Ding C, Meraz RF, Holbrook SR (2006) PsoL: a positive sample only learning algorithm for finding non-coding RNA genes. Bioinformatics 22(21): 2590–2596

[7] Carter RJ, Dubchak I, Holbrook SR (2001) A computational approach to identify genes for functional RNAs in genomic sequences, Oxford Univ Press. Nucleic Acids Res 29(19): 3928–3938

[8] Elkan C, Noto K (2008) Learning classifiers from only positive and unlabeled data. In: KDD '08: Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, New York: ACM 2008:213–220

[9] Mordelet F, Vert J-P (2010) A bagging SVM to learn from positive and unlabeled examples.

[10] Liu B, Lee WS, Yu PS, Li X (2002) Partially supervised classification of text documents. In: Proceedings of the nineteenth international conference on machine learning (ICML).

[11] Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning 20 (3): 273

[12] Powers, David M W (2007/2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies 2 (1): 37–63.

[13] Matthews, B. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta-Protein Structure, 405, 442–451.

[14] Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1): 5–32.

[15] Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, et al. (2007) Large-Scale Map-ping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. PloS Biol 5(1): e8.

[16] Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A: ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC bioinformatics 2006, 7(Suppl 1): S7

[17] Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning 20 (3): 273.
[18] Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 2 (12): 1137–1143.
[19] J. Rocchio. Relevant feedback in information retrieval. In G. Salton (ed.). The smart retrieval sys- tem: experiments in automatic document processing, Englewood Cliffs, NJ, 1971.
[20] Breiman, Leo (1996). "Bagging predictors". Machine Learning 24 (2): 123–140.
[21] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. *CRC press*.
[22] Dietterich, Thomas G. "Ensemble methods in machine learning." Multiple classifier systems. Springer Berlin Heidelberg, 2000. 1-15.
[23] Näppi, J.J., Regge, D., Yoshida, H.: Comparative Performance of Random Forest and Support Vector Machine Classifiers for Detection of Colorectal Lesions in CT Colonography. In: Yoshida, H., Sakas, G., Linguraru, M.G. (eds.) Abdominal Imaging 2011. LNCS, vol. 7029, pp. 27–34. Springer, Heidelberg (2012)
[24] Y. Tang, S. Krasser, Y. He, W. Yang, and D. Alperovitch, "Support Vector Machines and Random Forests Modeling for Spam Senders Behavior Analysis," in Proceedings of IEEE Global Communications Conference (IEEE GLOBECOM 2008), Computer and Communications Network Security Symposium, New Orleans, LA, 2008.
[25] Rios, G. and Zha, H. 2004. Exploring support vector machines and random forests for spam detection. In Proceedings of the First Conference on Email and Anti-Spam. Mountain View, CA, USA.
[26] Faith et al (2008) Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. Nucleic Acids Res 36 (Database is-sue): D866D870, Ocak 2008.
[27] Kerrien S, Aranda B, Breuza L, Bridge A, Broackes-Carter F, Chen C, Duesbury M, Dumousseau M, Feuermann M, Hinz U, Jandrasits C, Jimenez RC, Khadake J, Mahadevan U, Masson P, Pedruzzi I, Pfeiffenberger E, Porras P, Raghunath A, Roechert B, Orchard1 S, Hermjakob H (2011) The IntAct molecular interaction database in 2012. Nucleic Acids Res 40(1): D841–D846.
[28] Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol, 2:27:1–27:27
[29] "H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations" erişim adresi: http://www.cs.waikato.ac.nz/ml/weka/, erişim tarihi: 25 Mart 2014.
[30] "Eclipse.org – Juno Simultaneous Release" erişim adresi: http://www.eclipse.org/juno/, erişim tarihi: 27 Mart 2014.
[31] Butte, A. J., & Kohane, I. S. (2000, Ocak). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In Pac Symp Biocomput (Vol. 5, pp. 418-429).
[32] Altay, Gökmen, and Frank Emmert-Streib. "Revealing differences in gene network inference algorithms on the network level by ensemble methods."

Bioinformatics 26.14 (2010): 1738-1744.

EKLER

Ek A: Ölçüm Formülleri

Çizelge A.1 Keskinlik, Doğruluk, F-Ölçüsü ve MCC Formülleri

Değerin Adı	Formül
Keskinlik	$\frac{Gerçek\ Pozitif}{Gerçek\ Pozitif\ +\ Sahte\ Pozitif}$
Doğruluk	$\frac{Gerçek\ Pozitif}{Gerçek\ Pozitif\ +\ Sahte\ Negatif}$
F-Ölçüsü	$2 \times \frac{Keskinlik \times Doğruluk}{Keskinlik\ +\ Doğruluk}$
MCC	$\frac{(GP \times GN) - (SP \times SN)}{\sqrt{(GP+SP)\times(GP+SN)\times(GN+SP)\times(GN+SN)}}$

Ek B: Detaylı Deney Sonuçları

Çizelge B.1 AGPS ve AGPS-RF Algoritmalarının F-Ölçüsü ve MCC Değerleri Üzerinden Karşılaştırmalı Sonuçları

Veri Kümesi	P Yüzdesi	AGPS f-ölçüsü	AGPS-RF f-ölçüsü	AGPS MCC	AGPS-RF MCC
1	% 0,15	0,289	0,355	0,189	0,221
2	% 0,16	0,376	0,35	0,256	0,215
3	% 0,16	0,324	0,333	0,174	0,190
4	% 0,34	0,315	0,362	0,185	0,250
5	% 0,33	0,348	0,33	0,238	0,211
6	% 0,34	0,332	0,294	0,199	0,150
7	% 0,52	0,275	0,291	0,154	0,165
8	% 0,52	0,217	0,215	0,058	0,047
9	% 0,53	0,218	0,228	0,043	0,057
10	%0,71	0,294	0,276	0,198	0,169
11	%0,71	0,257	0,247	0,164	0,130
12	%0,71	0,222	0,221	0,089	0,090
13	% 0,92	0,214	0,224	0,112	0,128
14	% 0,91	0,228	0,259	0,169	0,195
15	% 0,92	0,216	0,22	0,116	0,130
16	% 1,13	0,210	0,201	0,154	0,131
17	% 1,14	0,202	0,194	0,142	0,132
18	% 1,13	0,204	0,222	0,136	0,170
19	% 1,33	0,148	0,149	0,097	0,112
20	% 1,33	0,172	0,16	0,139	0,121
21	% 1,34	0,150	0,153	0,101	0,061
22	% 1,56	0,105	0,104	0,077	0,077
23	% 1,57	0,118	0,098	0,109	0,074
24	% 1,57	0,110	0,106	0,084	0,081
25	% 1,77	0,071	0,061	0,089	0,065
26	% 1,76	0,063	0,053	0,072	0,045
27	% 1,77	0,058	0,078	0,052	0,100
Ortalama	% 0,94	0,212	0,214	0,133	0,130

Çizelge B.2 Roc-SVM ve Roc-RF Algoritmalarının F-Ölçüsü ve MCC Değerleri Üzerinden Karşılaştırmalı Sonuçları

Veri Kümesi	P Yüzdesi	Roc-SVM f-ölçüsü	Roc-RF f-ölçüsü	Roc-SVM MCC	Roc-RF MCC
1	% 0,16	0,009	0,664	0,061	0,604
2	% 0,16	0,223	0,651	0,195	0,588
3	% 0,17	0,124	0,705	0,229	0,656
4	% 0,34	0,222	0,513	0,331	0,434
5	% 0,32	0,290	0,656	0,328	0,604
6	% 0,34	0,094	0,609	0,184	0,550
7	% 0,52	0,023	0,592	0,100	0,542
8	% 0,51	0,076	0,676	0,148	0,636
9	% 0,52	0,117	0,563	0,211	0,505
10	%0,71	0,256	0,634	0,316	0,597
11	%0,70	0,210	0,701	0,246	0,673
12	%0,70	0,175	0,502	0,242	0,449
13	% 0,92	0,259	0,559	0,343	0,527
14	% 0,92	0,288	0,636	0,348	0,613
15	% 0,91	0,301	0,732	0,316	0,714
16	% 1,13	0,304	0,524	0,311	0,505
17	% 1,12	0,376	0,514	0,413	0,491
18	% 1,13	0,326	0,630	0,345	0,617
19	% 1,33	0,255	0,541	0,265	0,538
20	% 1,33	0,277	0,623	0,295	0,625
21	% 1,34	0,387	0,626	0,374	0,630
22	% 1,57	0,208	0,494	0,178	0,505
23	% 1,58	0,308	0,632	0,279	0,639
24	% 1,57	0,256	0,500	0,234	0,514
25	% 1,76	0,282	0,595	0,282	0,619
26	% 1,75	0,226	0,348	0,210	0,378
27	% 1,76	0,295	0,571	0,283	0,592
Ortalama	% 0,94	0,226	0,593	0,261	0,567

Çizelge B.3 Karma Algoritmanın F-Ölçüsü Değeri Üzerinden Karşılaştırmalı Sonuçları

Veri Kümesi	P Yüzdesi	Karma f-ölçüsü	AGPS f-ölçüsü	AGPS-RF f-ölçüsü	Roc-SVM f-ölçüsü	Roc-RF f-ölçüsü
1	% 0,16	0,608	0,289	0,355	0,009	0,664
2	% 0,16	0,380	0,376	0,35	0,223	0,651
3	% 0,17	0,413	0,324	0,333	0,124	0,705
4	% 0,34	0,427	0,315	0,362	0,222	0,513
5	% 0,34	0,550	0,348	0,33	0,290	0,656
6	% 0,33	0,538	0,332	0,294	0,094	0,609
7	% 0,52	0,466	0,275	0,291	0,023	0,592
8	% 0,53	0,489	0,217	0,215	0,076	0,676
9	% 0,52	0,558	0,218	0,228	0,117	0,563
10	%0,70	0,453	0,294	0,276	0,256	0,634
11	%0,70	0,445	0,257	0,247	0,210	0,701
12	%0,70	0,542	0,222	0,221	0,175	0,502
13	% 0,92	0,517	0,214	0,224	0,259	0,559
14	% 0,92	0,471	0,228	0,259	0,288	0,636
15	% 0,93	0,459	0,216	0,22	0,301	0,732
16	% 1,13	0,571	0,210	0,201	0,304	0,524
17	% 1,12	0,553	0,202	0,194	0,376	0,514
18	% 1,14	0,541	0,204	0,222	0,326	0,630
19	% 1,32	0,521	0,148	0,149	0,255	0,541
20	% 1,33	0,513	0,172	0,16	0,277	0,623
21	% 1,34	0,523	0,150	0,153	0,387	0,626
22	% 1,56	0,551	0,105	0,104	0,208	0,494
23	% 1,55	0,542	0,118	0,098	0,308	0,632
24	% 1,57	0,548	0,110	0,106	0,256	0,500
25	% 1,77	0,583	0,071	0,061	0,282	0,595
26	% 1,77	0,551	0,063	0,053	0,226	0,348
27	% 1,77	0,595	0,058	0,078	0,295	0,571
Ortalama	% 0,94	0,515	0,212	0,214	0,226	0,593

Çizelge B.4 Karma Algoritmanın MCC Değeri Üzerinden Karşılaştırmalı Sonuçları

Veri Kümesi	P Yüzdesi	Karma MCC	AGPS MCC	AGPS-RF MCC	Roc-SVM MCC	Roc-RF MCC
1	% 0,16	0,534	0,189	0,221	0,061	0,604
2	% 0,16	0,111	0,256	0,215	0,195	0,588
3	% 0,16	0,345	0,174	0,190	0,229	0,656
4	% 0,34	0,405	0,185	0,250	0,331	0,434
5	% 0,33	0,500	0,238	0,211	0,328	0,604
6	% 0,34	0,349	0,199	0,150	0,184	0,550
7	% 0,52	0,200	0,154	0,165	0,100	0,542
8	% 0,52	0,341	0,058	0,047	0,148	0,636
9	% 0,54	0,257	0,043	0,057	0,211	0,505
10	%0,71	0,196	0,198	0,169	0,316	0,597
11	%0,70	0,045	0,164	0,130	0,246	0,673
12	%0,71	0,364	0,089	0,090	0,242	0,449
13	% 0,91	0,234	0,112	0,128	0,343	0,527
14	% 0,93	0,209	0,169	0,195	0,348	0,613
15	% 0,93	0,110	0,116	0,130	0,316	0,714
16	% 1,13	0,340	0,154	0,131	0,311	0,505
17	% 1,12	0,261	0,142	0,132	0,413	0,491
18	% 1,12	0,348	0,136	0,170	0,345	0,617
19	% 1,34	0,282	0,097	0,112	0,265	0,538
20	% 1,35	0,306	0,139	0,121	0,295	0,625
21	% 1,34	0,271	0,101	0,061	0,374	0,630
22	% 1,57	0,279	0,077	0,077	0,178	0,505
23	% 1,56	0,282	0,109	0,074	0,279	0,639
24	% 1,57	0,294	0,084	0,081	0,234	0,514
25	% 1,76	0,391	0,089	0,065	0,282	0,619
26	% 1,76	0,291	0,072	0,045	0,210	0,378
27	% 1,76	0,398	0,052	0,100	0,283	0,592
Ortalama	% 0,94	0,294	0,133	0,130	0,261	0,567

Çizelge B.5 CLR ve ARACNE Algoritmalarının F-Ölçüsü Değeri Üzerinden Karşılaştırmalı Sonuçları

Veri Kümesi	P Yüzdesi	CLR f-ölçüsü	ARACNE f-ölçüsü	AGPS-RF f-ölçüsü	Roc-RF f-ölçüsü	Karma f-ölçüsü
1	% 0,15	0,561	0,550	0,355	0,664	0,608
2	% 0,15	0,600	0,531	0,35	0,651	0,380
3	% 0,16	0,617	0,501	0,333	0,705	0,413
4	% 0,35	0,583	0,512	0,362	0,513	0,427
5	% 0,33	0,595	0,498	0,33	0,656	0,550
6	% 0,33	0,614	0,494	0,294	0,609	0,538
7	% 0,51	0,612	0,478	0,291	0,592	0,466
8	% 0,51	0,580	0,462	0,215	0,676	0,489
9	% 0,52	0,572	0,432	0,228	0,563	0,558
10	%0,71	0,593	0,412	0,276	0,634	0,453
11	%0,71	0,573	0,381	0,247	0,701	0,445
12	%0,71	0,518	0,362	0,221	0,502	0,542
13	% 0,92	0,472	0,330	0,224	0,559	0,517
14	% 0,92	0,437	0,317	0,259	0,636	0,471
15	% 0,94	0,385	0,281	0,22	0,732	0,459
16	% 1,13	0,330	0,254	0,201	0,524	0,571
17	% 1,13	0,285	0,200	0,194	0,514	0,553
18	% 1,13	0,272	0,187	0,222	0,630	0,541
19	% 1,34	0,238	0,165	0,149	0,541	0,521
20	% 1,34	0,228	0,157	0,16	0,623	0,513
21	% 1,34	0,200	0,145	0,153	0,626	0,523
22	% 1,57	0,180	0,149	0,104	0,494	0,551
23	% 1,57	0,164	0,139	0,098	0,632	0,542
24	% 1,57	0,155	0,147	0,106	0,500	0,548
25	% 1,76	0,152	0,142	0,061	0,595	0,583
26	% 1,75	0,150	0,145	0,053	0,348	0,551
27	% 1,75	0,151	0,138	0,078	0,571	0,595
Ortalama	% 0,94	0,401	0,315	0,214	0,593	0,515

Çizelge B.6 CLR ve ARACNE Algoritmalarının MCC Değeri Üzerinden Karşılaştırmalı Sonuçları

Veri Kümesi	P Yüzdesi	CLR MCC	ARACNE MCC	AGPS-RF MCC	Roc-RF MCC	Karma MCC
1	% 0,16	0,616	0,397	0,221	0,604	0,534
2	% 0,16	0,582	0,462	0,215	0,588	0,111
3	% 0,16	0,497	0,300	0,190	0,656	0,345
4	% 0,34	0,565	0,287	0,250	0,434	0,405
5	% 0,33	0,55	0,319	0,211	0,604	0,500
6	% 0,35	0,547	0,342	0,150	0,550	0,349
7	% 0,51	0,499	0,265	0,165	0,542	0,200
8	% 0,51	0,516	0,231	0,047	0,636	0,341
9	% 0,52	0,477	0,220	0,057	0,505	0,257
10	%0,71	0,498	0,286	0,169	0,597	0,196
11	%0,70	0,44	0,331	0,130	0,673	0,045
12	%0,71	0,463	0,257	0,090	0,449	0,364
13	% 0,92	0,381	0,249	0,128	0,527	0,234
14	% 0,92	0,364	0,310	0,195	0,613	0,209
15	% 0,92	0,329	0,401	0,130	0,714	0,110
16	% 1,13	0,317	0,330	0,131	0,505	0,340
17	% 1,13	0,252	0,285	0,132	0,491	0,261
18	% 1,14	0,258	0,243	0,170	0,617	0,348
19	% 1,34	0,233	0,230	0,112	0,538	0,282
20	% 1,34	0,23	0,191	0,121	0,625	0,306
21	% 1,34	0,218	0,202	0,061	0,630	0,271
22	% 1,57	0,298	0,277	0,077	0,505	0,279
23	% 1,58	0,209	0,134	0,074	0,639	0,282
24	% 1,59	0,217	0,120	0,081	0,514	0,294
25	% 1,77	0,184	0,194	0,065	0,619	0,391
26	% 1,76	0,223	0,142	0,045	0,378	0,291
27	% 1,75	0,201	0,098	0,100	0,592	0,398
Ortalama	% 0,94	0,376	0,262	0,130	0,567	0,294

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı	: PANCAROĞLU, Doruk
Uyruğu	: T.C.
Doğum Tarihi ve Yeri	: 18.11.1989 Ankara
Medeni Hali	: Bekar
Telefon	: 0 (533) 242 84 62
Faks	:
e-mail	: dpancaroglu@etu.edu.tr

Eğitim

Derece	**Eğitim Birimi**	**Mezuniyet Tarihi**
Lisans	Sabancı Üniversitesi / Bilgisayar Bilimleri ve Mühendisliği	2010
Yüksek Lisans	TOBB Ekonomi ve Teknoloji Üniversitesi / Bilgisayar Mühendisliği	2014

İş Deneyimi

Yıl	**Yer**	**Görev**
2010-2012	TOBB Ekonomi ve Teknoloji Üniversitesi	Burslu Yüksek Lisans Öğrencisi
2012-	Savunma Teknolojileri Mühendislik ve Ticaret A.Ş.	Uzman Yazılım Mühendisi

Yabancı Dil

İngilizce

Yayınlar