KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

BİYOİNFORMATİK ALGORİTMALARIN KONUM TABANLI ÖNERİ SİSTEMLERİNDE UYGULANMASI

ABDURRAHMAN GÜN

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

BİYOİNFORMATİK ALGORİTMALARIN KONUM TABANLI ÖNERİ SİSTEMLERİNDE UYGULANMASI

ABDURRAHMAN GÜN

Prof.Dr. Nevcihan DURU		
Danışman,	**Kocaeli Üniv.**	**.....**
Doç.Dr. Nilüfer YURTAY		
Jüri Üyesi,	**Sakarya Üniv.**	**.....**
Dr.Öğr. Üyesi Alev MUTLU		
Jüri Üyesi,	**Kocaeli Üniv.**	**.....**

Tezin Savunulduğu Tarih: 24.06.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması kapsamında konum tabanlı öneri sistemleri için kullanıcıların bir sonraki davranışlarını rezervasyon kategori verileri üzerinden tahmin etmek için biyoinformatik algoritmaları kullanarak bir yöntem sunulmuş ve farklı parametre değerleri altında yöntem başarısı test edilmiştir.

Yüksek lisans öğrenimim boyunca değerli birikimlerini benimle paylaşan ve yoğun akademik yaşamında kıymetli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygıdeğer hocam Prof. Dr. Nevcihan DURU'ya teşekkürlerimi sunarım.

Ayrıca tez çalışmama bilgi, tavsiye ve yönlendirmeleri ile önemli katkılarda bulunan Dr. Öğr. Üyesi Alev MUTLU'ya,

Tanıştığımız andan itibaren desteğini ve yardımlarını esirgemeyen ve aynı odayı kendisiyle paylaştığım çalışma arkadaşım Arş. Gör. Furkan GÖZ'e ve yine bu süreçte beni yalnız bırakmayan Süleyman EKEN ve Ekin EKİNCİ hocalarıma,

Akademik çalışmalarım sırasında, birçok noktada bana destek olan Bilgisayar Mühendisliği Bölümü hocalarıma ve araştırma görevlisi arkadaşlarıma,

Hayatımın her aşamasında bana güç vererek en büyük dayanağım olan, sıkıntılarımı göğüsleyip mutluluklarımı paylaşan muhterem babacığıma; maddi ve manevi desteklerini tüm yaşamım boyunca bir an bile esirgemeyen fedakâr anneciğime; pek sevgili kardeşlerime; çalışmalarım sırasında bana kuvvet veren manevi büyüklerime şükranlarımı; ve özellikle bu çalışmayı tamamlamayı bana nasip eden Yüce Allah'a hamdlerimi bir borç bilirim.

Haziran–2021 Abdurrahman GÜN

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

A : Geçiş olasılıkları kümesi
$A_{ij,}$: i durumundan j durumuna yapılan geçiş sayısını
a_i : A dizisinin i. elemanı
a_{ij} : i durumundan j durumuna geçiş olasılık değeri
B : Durumlarda salınan gözlem olasılıkları kümesi
b_j : B dizisinin j. Elemanı
b_{ij} : i gözlem elemanının j durumunda gözlenme olasılık değeri
$e_i(x)$: x gözlem değerinin i durumunda gözlenme olasılık fonksiyonu
ey : Test dizisi eğitim yüzde oranı
G : Gözlem elemanları kümesi
k : Rezervasyonun ait olduğu kategori
ks : Modelleme için seçilen maksimum kullanıcı sayısı parametresi
M : İkili hizalama puan matrisi
O : Gözlem elemanları dizisi
o_t : Bir dizideki t anındaki gözlem elemanı
$O(n)$: Big-O karmaşıklık ifadesi
P : Olasılık fonksiyon değeri
Q : Bir süreç boyunca gerçekleşen ardışık durumlar dizisi
q_t : t anında Markov modelindeki bulunulan durum
R : Kullanıcıların rezervasyon veri kümesi
r_i : i. kullanıcının sıralı rezervasyon dizisi
S : Markov modellerinde durumlar kümesi
S_1 : Eşleşme durumu puan parametresi
S_2 : Yanlış eşleşme durumu puan parametresi
S_3 : Boşluk eşleşme durumu puan parametresi
t : Rezervasyonun yapıldığı zaman
td : Test kullanıcısı tercih dizisi parametresi
ts : Test kullanıcısı son tercih sayısı parametresi
U : Sistemdeki kullanıcılar kümesi
u_i : i. kullanıcı
V : Farklı gözlem değişkenlerinin kümesi
x : M matrisindeki yatay eksendeki pozisyon sırası
y : M matrisindeki dikey eksendeki pozisyon sırası
Π : Başlangıç durum olasılıkları kümesi
π_i : i durumunun başlangıç olasılık değeri
λ : SMM fonksiyon ifadesi
$\delta_t(i)$: t anındaki i yolunun kısmi olasılık değeri

Kısaltmalar

API : Application Programming Interface (Uygulama Programlama Arayüzü)
BLAST : Basic Local Alignment Search Tool (Temel Bölgesel Hizalama Arama Aracı)

DNA : Deoksiribo Nükleik Asit
KTÖS : Konum Tabanlı Öneri Sistemi
NP : Non-Polynomial (Polinom Olmayan)
NWA : Needleman-Wunsch Algoritması
PHMM : Profile Hidden Markov Model
RNA : Ribonükleik Asid
SMM : Saklı Markov Model
SP : Sum of Pairs (İkililer Toplamı)

BİYOİNFORMATİK ALGORİTMALARIN KONUM TABANLI ÖNERİ SİSTEMLERİNDE UYGULANMASI

ÖZET

Sonraki rezervasyon tahmin problemi bir sistemdeki kullanıcıların sonraki davranışlarını rezervasyon verileri üzerinden tahmin etmeyi amaçlamaktadır. Tezde konum tabanlı öneri sistemlerindeki kullanıcı tercihlerini tahmin etmek amacıyla Saklı Markov Modelerinin'nin özel bir uzantısı olan PHMM (Profile Hidden Markov Model) ve biyoinformatik algoritmaların birlikte kullanıldığı bir yöntem sunulmaktadır.

Bu çalışmada işbirlikçi filtreleme yaklaşımı temelinde bir sonraki davranışı tahmin edilmesi istenen kullanıcıya benzer profildeki diğer kullanıcıların seçilmesi için biyoinformatik hizalama algoritmalarından yararlanılmıştır. Seçilen kullanıcılar PHMM üzerinde modellenerek yöntemin test edilmesi aşamalarında Weeplaces veri kümesi kullanılmıştır.

Yapılan deneysel testler sırasında modellenecek kullanıcı sayısı ve kullanıcının tercih tahmininde dikkate alınan önceki tercih sayısı parametre değişimlerinin yöntemin başarısına etkisi incelenmiştir. Elde edilen sonuçlar kullanıcı sayısı parametresinin belli bir optimal değerde en yüksek başarıyı verdiği tercih sayısı parametre değişiminin ise etkisinin oldukça zayıf olduğu görülmüştür.

Anahtar Kelimeler: Biyoinformatik Algoritmalar, Konum Tabanlı Öneri Sistemleri, Saklı Markov Modeli, Rezervasyon Tahmini.

IMPLEMENTATION OF BIOINFORMATICS ALGORITHMS IN LOCATION-BASED RECOMMENDATION SYSTEMS

ABSTRACT

The next check-in prediction problem aims to estimate the next behavior of users in a system based on reservation data. In the thesis, a method in which the Profile Hidden Markov Model, which is a special extension of the Hidden Markov Models, and bioinformatic algorithms are used together, is presented in order to predict user preferences in location-based recommendation systems.

In this study, bioinformatics alignment algorithms were used to select other users with a profile similar to the user whose next behavior is desired to be predicted on the basis of a collaborative filtering approach. Selected users were modeled on PHMM and Weeplaces dataset was used in testing the method.

During the experimental tests, the effect of the changes in the parameter of the user number to be modeled and the parameter of previous preferences the number taken into account in the estimation of the user on the success of the method was examined. The results show that the user number parameter gives the highest success at a certain optimal value, and the effect of the preference parameter change is quite weak.

Keywords: Bioinformatics Algorithms, Location Based Recommendation Systems, Hidden Markov Models, Prediction of Check-in.

GİRİŞ

Konuma dayalı öneri, güncel ya da tarihi yerlerle ilgili konumsal bilgileri ve kişisel tercihleri de dikkate alarak kullanıcıya mekânlar, seyahat rotaları, arkadaşlar veya sosyal medya gibi içeriklerden oluşan öğeleri seçenekli bir şekilde sunan bir bilgi filtreleme hizmetidir [1]. Tavsiye sistemlerinin özelleşmiş bir çeşidi olan konum tabanlı öneri sistemlerinde, kullanıcıların geçmişte yaptıkları etkinliklerden ve sahip oldukları profille ilgili özelliklerden yararlanarak onların gelecekte yapması muhtemel sonraki eylemlerini tahmin etmeyi ve bu tahmine uygun olarak konumsal önerilerin üretilmesi amaçlanır.

Günümüzde mobil bilgi araçlarının gelişmesiyle birlikte çeşitli uygulamalar üzerinden konum tabanlı öneri sistemlerinin kullanımı giderek yaygınlık kazanmıştır. Bu sistemler sayesinde kullanıcılar özellikle sosyal ağ servisleri aracılığıyla gitmek istedikleri bir mekân için anında rezervasyon yapabilir, beğendikleri mekânları birbirlerine tavsiye edebilir veya bu yerler hakkında yorumlarını paylaşabilirler. Böylece sistem kullanıcıları ilgi duydukları benzer ya da farklı profildeki diğer kullanıcıları takip edebilme, başka kişilerle tanışarak kendi sosyal ağlarını genişletme ve merak ettikleri yerler hakkında kolayca fikir edinebilme gibi imkânlara sahip olurlar. Bununla birlikte insanlar yaşamları boyunca pek çok kararlar alırlar. Bu kararların önemli bir bölümünü yemek yeme, film izleme, kitap okuma veya bir yeri ziyaret etme gibi belli aralıklarla fakat sıkça aldıkları rutin kararlar oluşturmaktadır. Özellikle sürekli tekrar eden bu rutin içerisinde yapılan eylemler çeşitlendirilmediğinde bir süre sonra insanlar için bu durum can sıkıcı bir hale gelebilmektedir. Yine her ne kadar basit gibi gözükse de bu kararlar için genellikle çok fazla zaman harcanmakta ve uzun vadede yeni alternatiflerin bulunması güçleşebilmektedir. Öbür taraftan insanların ilk defa deneyimleyecekleri şeylerde kendilerini çoğu zaman güvensiz hissetmeleri onların yeni kararlarda daha temkinli davranmalarına yol açmaktadır. Ayrıca tecrübeli kişilerin tavsiyelerine olan ihtiyacın gerçek dünyada karşılamadaki zorluklar bu durumu daha da pekiştirmekte hatta onları yeni kararlar almaktan vazgeçirerek eski tercihlerine yönelmelerine sebep olabilmektedir. İşte bu noktada bu sistemler çeşitli online platformlar aracılığıyla

aynı anda onlarca kişinin tecrübe ve görüşlerine çok hızlı ve kolay bir biçimde erişim fırsatı sunarak karar alma süreçlerindeki zaman, güven ve imkan problemlerini de önemli ölçüde çözüme kavuşturmaktadır. Sağladığı tüm bu kolaylıklar ve avantajlar sayesinde konum tabanlı öneri sistemleri günümüzde giderek daha ilgi çeken araştırma konuları arasına girmiş durumdadır.

Kullanıcıya özgü tavsiyeler sunmayı amaçlayan tüm öneri sistemleri üzerinde işlem yapabilmek ve onu yorumlayabilmek için öncelikle bir veriye ihtiyaç duyar. Çoğu öneri sistemlerinde bu veri genellikle kullanıcıların daha önceki tercihleri, aktiviteleri ve değerlendirmelerinden oluşmaktadır. Bu sistemler için verinin kullanılma şekline göre farklı birçok yaklaşım bulunmakla birlikte bu yaklaşımları iki ana grupta sınıflandırmak mümkündür. Bunlar içerik tabanlı filtreleme ve işbirlikçi filtreleme yöntemleridir. İşbirlikçi filtreleme benzer profillere sahip kullanıcıların davranışlarını esas alırken içerik tabanlı filtreleme kullanıcı ile önerilen öğelerin özellikleri arasındaki ve aynı zamanda bu öğelerin birbirleriyle arasındaki ilişkileri kullanarak bir çıkarım yapmaya çalışır. Örneğin kullanıcılara mekân tavsiyesinde bulunan bir öneri sistemini düşünelim. Burada A ve B isimli iki mekana kullanıcılar tarafından birbirine yakın zamanlarda rezervasyon yapıldığını farzedelim. Mevcut durumda içerik tabanlı filtreleme bu mekanlar arasında bir ilişki olduğunu varsayacak ve A mekanını tercih eden herhangi bir kullanıcıya B mekanını da önerecektir. İşbirlikçi filtreleme ise A ve B mekanlarına bir kullanıcı profilinde daha önce rezervasyon yapılmışsa geçmişte A ve C mekanlarına rezervasyon yapan benzer profile sahip başka bir kullanıcıya B mekanını da önerecektir. Dolayısıyla işbirlikçi filtrelemede kullanıcı profillerine özgü tercihler ön plana çıkarken içerik tabanlı filtrelemede öğeler arasında daha genel bir ilişki söz konusudur. Bu durum içerik tabanlı yaklaşımda birbirinden çok farklı tercihlere sahip kullanıcılara aynı önerilerin yapılması sorununu da beraberinde getirmektedir. Önerilecek nesnelerin içeriğine ihtiyaç duymadan işlem yapabilmesi bu sayede en karmaşık öğelerde dahi etkin sonuçlar üretmesi ve ayrıca kullanıcı eğilimlerini profil özelinde yansıtan öneriler sunabilmesi işbirlikçi filtrelemenin güçlü taraflarını ortaya koymaktadır. Tüm bu avantajları sebebiyle İşbirlikçi filtreleme çoğu öneri sistemlerinde olduğu gibi konum tabanlı öneri sistemlerinde de farklı tekniklerle yaygın bir şekilde uygulanmaktadır. Doğrusal Regresyon Modelleri, Yapay Sinir Ağları Modelleri, Örüntü Eşleme Algoritması ve Saklı Markov Modelleri bu tekniklerden sadece bazılarıdır.

Saklı Markov Modelleri (SMM) özellikle trend tahminleme için birçok öneri sisteminde çokça başvurulan stokastik bir modeldir. SMM, modelleme yaparken kronolojik olarak sıralı gözlem değerlerinden oluşan bir dizideki elemanların ardışıklık ilişkisinden faydalanır. Özellikle zamana bağlı olarak belli bir süreç boyunca değişen verilerin modellenmesinde etkin olduğu bilinmektedir. Kullanıcı rezervasyon bilgileri hem kullanıcı alışkanlıklarıyla ilişkili olarak zamansal bir süreçte oluşmaları hem de KTÖS (Konum Tabanlı Öneri Sistemi) deki kişilerin kişisel tercihlerini yansıtan en karakteristik verilerden olması bakımından SMM ile birlikte kullanılması için oldukça elverişlidir. Kullanıcı rezervasyon tercihlerini KTÖS kapasitesine bağlı olarak etkileyen birçok faktör bulunmaktadır. Bunlardan başlıcaları; kullanıcının rezervasyon yaptığı zamanın periyodu, kullanıcının yapmayı arzuladığı aktivitenin türü ve rezervasyon yerinin konumudur. SMM; rezervasyonların periyot ilişkisini kendi model yapısında bulunan durumlar arası geçiş olasılıklarını kullanarak, rezervasyon türüyle olan ilişkisini ise yine modelin durum yapılarında tutulan gözlem değerlerinin salınma olasılıkları üzerinden modeller.

SMM'nin daha iyi sonuç verebilmesi için verinin modellemeye uygunluğu önem taşımaktadır. Kullanılacak veri yapısal olarak SMM'ye uygun olsa bile veride Markov özelliklerinin daha fazla öne çıkmasını sağlayacak bazı ön işlemlerin uygulanması verimi artırabilmektedir. Bu doğrultuda modelleme öncesinde k-NN, doğrusal regresyon veya Naive Bayes gibi bazı sınıflandırma ve kümeleme algoritmalarından farklı veriler için yararlanılmaktadır [2,3]. Ayrıca biyoinformatikte SMM'nin hizalama algoritmalarıyla birlikte kullanımı da aynı amacı taşımaktadır [3].

DNA, RNA ve protein dizileri biçiminde depolanan verilerin analiz edilmesi, modellenmesi, iki organizma arasındaki ortak genlerin tespiti, ata dizisinin tahmini gibi konular biyoinformatiğin başlıca ilgi alanları arasında yer almaktadır. Giderek daha da artan bu büyük hacimli ve çok sayıda olası kombinasyon içeren karmaşık verilerin makul bir süre içerisinde incelenebilmesinin klasik yöntemlerle gerçekleştirmek neredeyse imkansızdır. Bu durum model tabanlı yaklaşımların kullanımını yaygın hale getirdiği gibi SMM'nin biyoinformatikte protein dizilerinin modellenmesinde çokça kullanıldığı da bilinmektedir. Bununla birlikte hizalama

algoritmaları protein dizilerindeki benzerliklerin bulunması, ikincil ve üçüncül yapılarının tahmin edilmesinde ve modelleme aşamalarında kritik öneme sahiptir. Hizalama algoritmaları sayesinde bu dizilerin mümkün olan çok sayıda aynı elemanları birbiriyle eşleşecek şekilde hizalanır ve böylece ortak özellikler öne çıkarılmış olur. Bu da hem dizi özelliklerinin yorumlanmasında hem de oluşturulan modelin daha ilişkisel sonuçlar elde etmesinde büyük önem taşımaktadır.

Tez çalışması kapsamında biyoinformatik algoritmaların biyolojik verilerden başka kullanıcı rezervasyon bilgilerini içeren konum bazlı sosyal medya veri kümelerinde de uygulanabileceği düşünülerek bu algoritmaların kullanıcı tercihlerini tahmin etmede ve buna uygun önerilerin üretilmesinde etkili olması beklenmiştir. Ayrıca kullanıcı tercihlerindeki benzerliklerin yorumlanmasında ve birbirine yakın profile sahip kullanıcıların tespit edilmesinde kullanılabileceği öngörülmüştür. Bunun yanı sıra biyoinformatik algoritmaların farklı parametre değerleri için sonuçların ortaya konularak böylece en etkin parametre değerlerinin bulunması hedeflenmiştir.

Tezde KTÖS'deki kullanıcı tercihlerini kategorik olarak tahmin etmek amacıyla SMM'nin bir alt çeşidi olan Profile Hidden Markov Model ve biyoinformatik algoritmaların birlikte kullanıldığı bir yöntem sunulmuştur. Yöntemin gerçeklenmesi ve test edilmesi aşamalarında Weeplaces veri kümesi kullanılmıştır. Weeplaces; Facebook Places, Foursquare ve Gowalla gibi diğer konum bazlı sosyal ağ servislerinin API (Application Programming Interface) lerine entegre edilerek kullanıcı aktivitelerinin görselleştirilmesini amaçlayan çevrimiçi bir platformdur. Yöntem, kullanıcıların bir sonraki rezervasyon tercihinin hangi kategoriye ait olacağını tahmin etmeye çalışır. Bu amaçla veri kümesindeki kayıtlar günün belli saat aralıklarına ayrılarak 6 farklı veri kümesi oluşturuldu. Bunun nedeni kullanıcıların rutin ve alışkanlıklarının gün içerisinde değişiklik göstermesidir. Çalışan bir kişi için çalışma saatlerinde genellikle herhangi bir sosyal aktivite yapma durumu çok kısıtlıyken iş çıkış saatinden sonraki zaman dilimlerinin sosyal aktivitelerin gerçekleştirilmesi için daha elverişli olması buna örnek olarak verilebilir. Daha sonra bu veri kümelerinde ayrı ayrı her kullanıcı için tüm kayıtlar kronolojik olarak sıralandı ve bu kayıtların ait olduğu kategori bilgisi alınarak kullanıcı dizileri oluşturuldu. İkinci aşamada ise birbirine en yakın kullanıcılar hizalama algoritmaları yardımıyla tespit edildi ve bunlar birbiriyle çoklu hizalanarak

modellemeye hazır hale getirildi. Üçüncü aşamada bu hizalanmış kullanıcı dizileri Profile Hidden Markov Modeli ile modellendi ve Viterbi algoritmasıyla kullanıcının bir sonraki rezervasyon kategorisi tahmin edilmeye çalışıldı. Son olarak elde edilen bu sonuçlar veri kümesindeki gerçek verilerle karşılaştırılmıştır.

Tez çalışmasının ilk bölümünde öneri sistemleri ve özellikle de konum tabanlı öneri sistemleri hakkında bilgi verilerek problem tanımı yapılmıştır. İkinci bölümde biyoinformatik algoritmalar incelenerek ikili ve çoklu hizalama algoritmaları detaylıca ele alınmıştır. Üçüncü bölümde ise Markov Modellerinin yapısı, çeşitleri tanıtılmış olup Saklı Markov Modelleri'nin matematiksel gösterimlerine yer verilmiştir. Ayrıca modellenen veriden en son tahmin çıkarma işlemi için yararlanılan Viterbi algoritması da üçüncü bölümde anlatılmaktadır. Dördüncü bölümde önerilen yöntem sunularak kullanılan veri kümesi hakkında bilgi verilmiştir. Beşinci bölümde yapılan test sonuçları değerlendirilmiştir. Tezin son bölümü olan Sonuçlar ve Öneriler bölümünde, yapılan çalışmadan elde edilen sonuçlar özetlenerek çalışmanın katkısı vurgulanmıştır.

1. GENEL BİLGİLER

Günlük yaşamımızı sürdürürken onlarca kez karar almak durumunda kalırız. Bu kararlardan bazıları ne yemek yiyeceğimiz, hangi elbiseyi giyeceğimiz, hangi kitabı okuyacağımız veya sosyal bir aktivite için nereye gideceğimiz gibi genelde hızlıca aldığımız ve sürekli tekrarlanan kararlardan oluşurken diğer bir kısmı ise nerede çalışacağımız, kiminle evleneceğimiz gibi etraflıca üzerinde düşünmeyi gerektiren daha karmaşık kararlardır. Farklı zorluk derecelerine sahip farklı kararlar olmasına rağmen çoğu kez çabucak alınması gereken bu basit kararlar için bile oldukça fazla zaman harcayabilmekte ve böylece bunlar yaşamımızda ekstra bir yük haline gelebilmektedirler.

Daha önce insanlar kendi çevrelerindeki diğer insanlardan ve kendi arkadaşlarından aldıkları tavsiyelere ya da reklam, gazete, dergi, televizyon gibi yazılı ve sözlü medya araçları vasıtasıyla topladığı bilgilere göre kararlar alıyorlardı. Ancak bu şekilde toplanan bilgilerin oldukça sınırlı olmasının ötesinde elde edilmesi için ilave bir zaman ve çaba gerektirmekteydi. Ayrıca elde edilen bu önerilerin taraflı veya önyargılı olma durumu da söz konusuydu. Tüm bunların aksine bilgisayar destekli teknoloji ürünü olan öneri sistemlerinde ise sadece tanınan belli kişilerden değil aynı zamanda tanıdık olmayan çok sayıdaki insanlardan da hızlı ve kolay bir şekilde öneriler elde etmek mümkündür.

Öneri sistemleri en basit ifadeyle yığın veri içerisinden kullanıcının tercihlerine en uygun olanları bulmak için bir filtreleme hizmeti sunar. Bunu gerçekleştirebilmesi için üzerinde işlem yapabileceği bir veriye ihtiyaç vardır. Burada veri doğrudan önerilecek öğeleri içerdiği gibi filtrelemede ölçüt ve yardımcı unsur olarak kullanılan kullanıcı bilgileri, etkinlikleri ve diğer harici bilgileri de kapsar. Tüm bu bilgiler kullanıcı derecelendirmelerinin toplanması gibi tipik bir şekilde doğrudan olabilir ya da dinlenen şarkılar, indirilen uygulamalar, ziyaret edilen web siteleri ve okunan kitap verileri gibi kullanıcı davranışlarını izleyerek dolaylı olarak da elde edilebilir [4].

Öneri sistemlerinden film [5-7] müzik [8-10], haber [11,12], televizyon [13-14], kitap [15-16], e-öğrenme [17-18], ürün[19-20] ve konum[21-23] gibi alanlar başta olmak üzere pek çok farklı alanlarda yararlanılmaktadır. Bu sistemlerin gelişerek yaygınlaşmasıyla birlikte araştırmacılara ek olarak endüstri ve işletmeler tarafından da büyük ilgi görmektedir. Buna en iyi örnek müşterilerine yirmi yılı aşkın bir süredir ürün önerisinde bulunan Amazon ve Netflix verilebilir [24].

Öneri sistemleri iki farklı hedef için kullanılabilir: Bunlardan birincisi bir kullanıcının henüz gerçekleştirmediği bir sonraki eylemini tahmin etmek, ikincisi ise kullanıcı için en uygun öğeleri bulup bunları listelemektir [25]. Her iki durum da öz itibariyle kullanıcıya yönelik en uygun önerileri sunması bakımından aynı amaca hizmet ederler.

Öneri sistemleri tavsiye oluştururken benimsediği yaklaşıma göre genelde üç ayrı sınıfta incelenirler: Bunlar içerik filtreleme tabanlı öneri sistemleri, işbirlikçi filtreleme tabanlı öneri sistemleri ve hibrit öneri sistemleridir [26,27].

İçerik tabanlı filtreleme, benzer özelliklere sahip öğelerin kullanıcılar tarafından yine benzer şekilde derecelendirileceği varsayımına dayanır [28]. Dolayısıyla bir kullanıcının seçtiği içeriklerde birbirine benzer karakteristik özelliklerin olduğunu kabul ederek bu özellikler arasındaki ilişkiyi bulmaya çalışır. Bunun için öncelikle kullanıcının daha önce değerlendirdiği öğeler arasından yüksek dereceye sahip olanların ortak özellikleri tespit edilir. Böylece bir nevi kullanıcının tercih profili çıkarılmış olur. Sonrasında ise sistemde bulunan tüm öğeler içerisinden bu profile en uygun olanlar kullanıcıya önerilir [4,27]. İçerik tabanlı filtreleme öneri üretebilmek için hedef kullanıcı dışında başka kullanıcıların değerlendirmelerine gerek duymaz. Bir öğenin analiz edilebilecek kadar özellik bilgisine sahip olması ve hedef kullanıcının daha önceki tercihlerinin bilinmesi bu yöntem için yeterlidir. Bu sayede sisteme yeni eklenen öğelerde dahi kullanıcıya öneride bulunabilir. Kullanıcılardan bağımsız olması ve homojen bir yapıya sahip oluşu içerik tabanlı filtrelemenin avantajlı taraflarıdır. Öbür taraftan aşırı uzmanlaşma, içerik analizi ve yeni kullanıcı problemleri bu yöntemin dezavantajlarını oluşturur. [27].

İşbirlikçi filtreleme benzer özelliklere sahip kullanıcıların yapacakları tercihlerin de birbirine benzer olacağı varsayımına dayanmaktadır [29]. Bu yaklaşım insanların

karar almadaki davranış biçimini model alır. Öteden beri insanların kararlarının oluşmasında kendi düşüncelerinin yanı sıra tanıdığı gruplardan kendilerine ulaşan fikir ve tecrübeler büyük bir etki etmiştir [4]. Bu yöntem başkalarının daha önce tecrübe ettiği şeyler için tekrar detaylı araştırma ve analiz yapmak yerine var olan tecrübelerden yararlanmayı benimser. Bu sayede bir kararın oluşmasında tekraren çaba gerektiren bu aşamalardan tasarruf edilmiş olur. Sağladığı kolaylığın yanında insanların kendisini bir topluluğa kabul ettirmek ya da başkalarından takdir görmek gibi doğrudan sosyal amaçlara sahip kararlar aldığı da göz önünde bulundurulduğunda işbirlikçi filtrelemenin cazibesi daha çok artmaktadır [30].

İşbirlikçi Filtreleme iki alt sınıfa ayrılmaktadır. Bunlar Bellek Tabanlı İşbirlikçi Filtreleme ve Model Tabanlı İşbirlikçi Filtreleme Yöntemleridir.

Bellek tabanlı işbirlikçi filtreleme, tüm kullanıcıların ürünler hakkındaki derecelendirmelerinin tutulduğu bir veri kümesi üzerinde benzer kullanıcıları kümelemek için istatistiksel yöntemlerden yararlanır. Oluşturulan kullanıcı gruplarının tercihleri daha sonra farklı algoritmalarla birleştirilerek kullanıcı profilleri çıkarılır. Son olarak en uygun profiller üzerinden hedef kullanıcıya öneri sunulur. İşbirlikçi filtrelemenin ilk uygulamaları bellek tabanlı filtrelemeye örnektir. Seyreklik ve ölçeklenebilirlik bellek tabanlı filtrelemenin iki ana problemini oluşturur. Seyreklik, tüm kullanıcıların sistemde var olan az sayıdaki öğe için değerlendirme yapması dolayısıyla her öğe için yeterli değerlendirme verisinin olmamasını ifade eder. Ölçeklenebilirlik ise çok sayıdaki kullanıcı ve öğe içeren veri tabanlarında etkinliğinin zayıf olması anlamına gelir [31].

Model tabanlı işbirlikçi filtrelemede kullanıcı tercihleri belirlenen bir model üzerinden tanımlanmaya çalışılır. Bunun için bir eğitim veri kümesi kullanılarak verinin modele başta öğretilmesi gerekmektedir. Model eğitildikten sonra hedef kullanıcının tercihleri tahmin edilir. Özellikle büyük verilerde işbirlikçi filtrelemenin model yaklaşımı daha çok kullanılmakta ve Bellek Tabanlı filtrelemeye göre çok daha etkin sonuçlar vermektedir. Model tabanlı filtreleme yöntemlerinin oluşturulmasında makine öğrenmesi teknikleri ilham vermiştir [31]. Ayrıca bu yaklaşımın uygulanmasında Bayesian Modelleme [32], Kural Tabanlı Yaklaşımlar [33], Matrix Faktörizasyonu [34] ve Markov Modelleri gibi olasılık hesaplamasına dayanan yaklaşımlar da [35] kullanılmaktadır.

Öneri sistemlerinin bir alt çeşidi olan konum tabanlı öneri sistemlerinde İçerik tabanlı filtreleme ve işbirlikçi filtreleme yöntemlerinin her ikisinden de yararlanılmaktadır. Bu sistemlerde kullanıcıya öneride bulunurken konumsal bilgi öne çıkmaktadır. Konum öneri sistemleri genel konum önerisi ve kişisel konum önerisi şeklinde iki kategoriye ayrılabilir [36]. Genel konum önerisinde kullanıcılara genellikle en popüler olan mekânlar sunulmaktadır. İçerik tabanlı filtreleme yöntemi bu tür konum önerilerinde ağırlıklı olarak benimsenmektedir. Ancak bireysel tercihleri karşılamadaki zayıflığından dolayı tüm kullanıcılar bu sistemlerden aynı mekan önerilerini almaktadırlar. Kişisel konum önerisi ise kullanıcıların kendi tercihlerine göre en uygun mekanları önermeyi amaçlar. İşbirlikçi filtreleme yaklaşımının kullanıldığı konum tabanlı sistemler büyük ölçüde bu kategoriye girmektedir [37].

Konum tabanlı öneri sistemlerinin erken örneklerinden Park ve ark. yaptıkları çalışmada işbirlikçi filtreleme sayesinde ilgili anahtar kelimeleri mekan ve kullanıcı profilleri arasında eşleştirmeyi amaçlamışlardır. Bu çalışmada Bayes ağ modeli kullanılarak bir restoranın fiyat ve kategori bilgileri ile kullanıcıların yaş, cinsiyet, gelir ve mutfak tercihleri gibi profil bilgileri ilişkilendirilmiştir [38]. Bir başka çalışmada Horozof ve ark. yine restoran tavsiyesi için iyileştirilmiş bir işbirlikçi filtreleme yöntemi önermişlerdir [39]. Bunun yanında daha kişisel öneriler sunabilmek için Ye ve ark. rezervasyon verilerini kullanarak coğrafi uzaklığın ve sosyal yapıların konum önerilerine etkilerini incelemişlerdir [40]. Benzer şekilde Takeuchi ve ark. bir mağaza öneri sistemi için kullanıcıların geçmiş konum bilgilerini analiz ederek onların kişisel tercihlerini tahmin etmeye çalışmışlardır [41]. Zheng ve ark. nın yaptığı çalışmada ise çok sayıda kullanıcının konumsal hareket modellerini araştırarak tavsiyenin kalitesini geliştirmek için kalabalık bilgeliğini kullanmışlardır [42]. Konum tabanlı öneri, konum önerisinin yanı sıra arkadaş önerisini de içermektedir. Buna örnek olarak Yu ve ark. yaptığı çalışma verilebilir. Bu çalışmada denetimsiz bir bağlantı analizi modeliyle (link analysis model) kullanıcıların coğrafi geçmişleri ve sosyal ilişkileri modellenmiş ve rastgele yürüyüş (random walk) algoritması kullanarak bağlantı olasılığı ve arkadaş önerisi skorları hesaplanmıştır [43]. Konum tabanlı öneri sistemlerinde kullanıcıların rezervasyon (check-in) verisinden tavsiye ve profil çıkarımında etkin olarak yararlanılmaktadır. Noulas ve ark. kullanıcıların konum rezervasyon geçmişinden faydalanarak bir

sonraki rezervasyon konumunu tahmin etmeye çalışmışlardır. Bu çalışmada doğrusal regresyon ve M5 model ağaç yapılarından oluşan iki denetimli öğrenme yöntemini bir arada kullanarak kullanıcıların rezervasyon tercihini etkileyen bireysel özelliklerin çıkarılmasını hedeflenmişlerdir [44]. KTÖS'lerde bir sonraki konum tahmininin performansını iyileştirmek için Likhyani ve ark. konum verilerini harita bilgileriyle birleştirerek mekanları doğru kategoriler altında ilişkilendirmeye çalışmıştır [45]. Model tabanlı yaklaşımlar arasında oldukça popüler olan Markov Modellerinin konum tabanlı öneride kullanıldığı ayrıca bilinmektedir. Li ve ark. nın Twitter ve Foursquare deki ilgili konum verileri üzerinden Markov Modellerini kullanarak kullanıcının bir sonraki ziyaret edeceği yeri tahmin etmeye yönelik çalışması buna örnektir [46]. Aynı problemin çözümüne yönelik bir başka çalışmada Ye ve ark. kullanıcıların coğrafi işaretli reklam ve kuponların yanında rezervasyon verilerini kategori bilgileriyle beraber kullanmışlardır. Daha sonra bu verinin modellenmesi Saklı Markov Modelleri üzerinden gerçekleştirilmiş ve kullanıcının en yüksek olasılığa sahip bir sonraki konumu hesaplanmıştır [47].

1.1. Problem Tanımı ve Motivasyon

Bu tez çalışması konum tabanlı öneri sistemleri için kullanıcıların bir sonraki davranışlarını rezervasyon verileri üzerinden tahmin etmeyi hedeflemektedir. Bu konu literatürde sonraki rezervasyon tahmin problemi (predicting next check-in problem) olarak adlandırılmaktadır. [48]. Bir çok sosyal içerikli uygulamalardaki rezervasyon kayıtlarında, bu problemin çözümüne yönelik olarak kullanıcı eğilimlerinin analizini kolaylaştırmak için ilgili mekan, kullanıcı ve kayıt özelliklerine ilaveten ait olduğu kategori bilgisi de tutulmaktadır. Rezervasyon kategori bilgisi kullanıcı tercihlerinin sınıflandırılmasında ve davranış özelliklerinin çıkarılmasında oldukça kullanışlıdır. Bu nedenle sonraki rezervasyon tahmin problemi rezervasyon kategori verileri üzerinden aşağıdaki gibi tanımlanabilir.

N adet kullanıcıdan oluşan bir sistemdeki kullanıcıların kümesine U ve tüm kullanıcıların yapmış olduğu rezervasyonların kümesine de R diyebiliriz. Böylece $U = \{u_1, u_2, \ldots, u_n\}$ için kullanıcıların rezervasyonlarından oluşan veri kümesi $R = \{r_1, r_2, r_3, \ldots r_n\}$ olarak gösterilebilir. Burada r_i, bir kullanıcının farklı zamanlarda yapmış olduğu rezervasyonların sıralı bir dizisine karşılık gelmektedir. Ayrıca kullanıcı rezervasyon dizilerini onları oluşturan kategori bilgisiyle ifade

edecek olursak her bir kullanıcı için bu diziler $r_i = \{(k_1, t_1), (k_2, t_2), \ldots, (k_m, t_m)\}$ şeklinde yazılır. Burada k, rezervasyonun ait olduğu kategoriyi ve t ise rezervasyonun yapıldığı zamanı temsil etmektedir. Son olarak bu ifadeyi sadeleştirmek amacıyla zamansal bilgileri çıkarıp rezervasyon yapılan yerlerin kategorilerini ziyaret edilme sırasına göre kronolojik olarak sıraladığımızda bir kullanıcı rezervasyon dizisi için $r_i = (k_1 -> k_2 -> k_3 -> \ldots -> k_m)$ genel ifadesini elde ederiz.

Sonraki rezervasyon probleminin çözümü tezde dört aşamada gerçekleşmektedir. Birinci aşamada R kümesi içinde yer alan r_i genel ifadesine göre düzenlenmiş tüm kullanıcı rezervasyon dizileri arasından hedef kullanıcı rezervasyon dizisine en benzer diğer kullanıcıların dizileri seçilir. İkinci aşamada ise hedef kullanıcının tercihlerini daha iyi yansıtabilmek ve profil özelliklerinin belirginleştirilmesi amacıyla birinci aşamada seçilen kullanıcı rezervasyon dizilerinin modelleme öncesinde birbiriyle en çok örtüşen dizilime ulaştırılması amaçlanır. Tezin odak noktasını oluşturan 1. ve 2. aşamalarda biyoinformatik algoritmalardan yararlanılmaktadır. Üçüncü aşama modellemeye hazır hale getirilen kullanıcı dizilerinin modellenerek olasılık değerlerinin çıkarılmasını içerir. Son ve 4. aşamada ise hazır model üzerindeki hedef kullanıcı dizisi için en yüksek olasılığa sahip kategori değeri bulunur.

2. BİYOİNFORMATİK ALGORİTMALAR

Biyoinformatik, bilişim teknolojilerinden yararlanarak özellikle biyo-moleküler verilerin elde edilmesi, depolanması, organizasyonu, analizi, görselleştirilmesi ve bu görevlere yönelik yeni yöntemlerin geliştirilmesini amaçlayan disiplinler arası bir bilim dalı olarak tanımlanabilir. Biyoinformatiğin motivasyonunun iki ayağı bulunmaktadır. Bunlardan birincisi biyolojide özellikle de moleküler seviyede yapılan çalışmalarda ortaya çıkan verilerin giderek çok daha büyük bir hacme ulaşması ve bunun sonucunda bu verilerin analiz edilmesinin insan gücüne dayalı klasik yöntemlerin yetersiz kalmaya başlamasıdır. İkincisi ise biyo-moleküler çalışmaların yoğunlaştığı aynı dönemlerde bilgisayarlı hesaplama sistemlerindeki ilk uygulamaların başlamasıyla birlikte insanlar için çok uzun süreler gerektiren sıralı işlemleri çok kısa sürelerde yapabilen bu sistemlerdeki potansiyelin fark edilmesidir. Dolayısıyla biyoloji alanında meydana gelen ihtiyaç ve bilgisayarlı hesaplamanın sağladığı büyük kabiliyet bu iki disiplini birbirine yaklaştırarak yeni bir araştırma alanının doğmasına yol açmıştır.

Biyoinformatiğin ilk ortaya çıkışı DNA moleküllerinin genetik bilgiyi taşıyan ve aktaran en küçük yapılar olduğunun keşfedilmesinden sonradır. 1953'de keşfedilen DNA'nın sarmallarındaki genetik kod yapısının deşifre edilmesi bu tarihten 13 yıl sonra gerçekleşmiş ve ilk DNA sıralama yöntemlerinin uygulanması ise yaklaşık 25 yıl sonra mümkün olmuştur [49]. Moleküler biyolojideki araştırmalar 1950'lilerden 1960'ların başına kadar protein dizilerinin sıra yapılarının belirlenmesine yani dizi hizalama (sekanslama) işlemine odaklanmıştır. Bir dizi hizalaması, incelenen diziler arasındaki işlevsel, yapısal, evrimsel ve diğer ilgi alanları hakkında ek bilgi sağlayabilen "benzerlik" bölgelerini bulmak amacıyla DNA, RNA ve protein dizilerinin düzenlenmesini içerir [50]. Protein sekanslamadaki ilk yöntemlerden Edman sekanslama yönteminin tek seferde en fazla 50-60 aminoasiti sekanslayabilmesi ve daha fazla aminoasitten oluşan proteinlerde bu yöntemin pratik olmaması 1960'ların başında ilk biyoinformatik yazılımı "Comprotein" in geliştirilmesini sağlamıştır. Biyoinformatikteki ikinci büyük sıçrama Emile

Zuckerkandl ve Linus Pauling'in biyomoleküler dizilerin birer bilgi taşıyıcı oldukları fikrini öne sürmeleriyle başlamıştır. Bu fikre göre harflerden oluşan bir kelimenin belli bir anlamı karşıladığı gibi aminoasitler de belli bir moleküler fonksiyonu karşılamak için bir düzen içerisinde bir araya gelmekteydiler. Böylelikle protein dizilerinin atalardan sonraki nesillere küçük farklılıklarla aktarıldığı düşüncesi mevcut protein yapılarından hareketle ilk ata protein dizi yapılarına ulaşılabileceği ve yine bu sayede türler arasındaki ilişkinin çözülebileceği varsayımlarını ortaya çıkarmıştır. Bu durum biyoinformatikte dizi benzerliklerinin bulunması için tekrarlanabilir dinamik hizalama algoritmalarına olan ihtiyacı doğurmuştur. Böylece 1970 yılında ilk hizalama algoritması Needleman-Wunch algoritması geliştirilmiştir. 1980'lerden sonra ise ikili hizalama algoritmayı temel alarak çoklu hizalama algoritmaları geliştirilmeye başlanmış ve bu algoritmalar günümüzde de farklı versiyonlarıyla yoğun olarak kullanılmaktadır [49].

Hizalama algoritmalarının geliştirilmesinden sonra biyolojik sekans veritabanları, keşfedilen yeni sekanslar nedeniyle gün geçtikte hızla genişlemeye başlamıştır. Bu durum mevcut dinamik hizalama algoritmalarında artan dizi sayısı dolayısıyla işlem sürelerinin çok uzamasına yol açmıştır. Böylece bu büyük veritabanları üzerinde çok daha hızlı çalışabilecek yeni hizalama algoritmalarına ihtiyaç oluşmuştur. Bu problem BLAST ve FASTA gibi sezgisel hizalama algoritmalarının geliştirilmesini motive etmiştir. En optimal sonucu elde eden önceki hizalama algoritmaların aksine sezgisel hizalama algoritmalarında en optimal sonucun bulunması kesin değildir. Bunun yerine sezgisel hizalama algoritmaları çok daha kısa süreler içerisinde kabul edilebilir sonuçlar elde etmeye odaklanır. Bu yeni yaklaşımın dizi hizalama probleminin çözümünde kullanılmasıyla birlikte hizalama algoritmalarının literatürde optimal ve sezgisel olarak iki ayrı sınıfta değerlendirilmesi de biyoinformatik için ayrıca önem taşımaktadır.

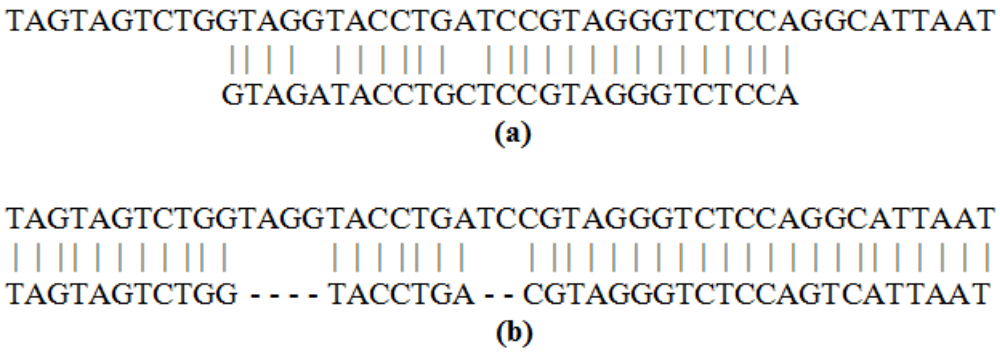
Bu bölümün devamında hizalama algoritmaları ikili ve çoklu hizalama olarak iki alt başlık halinde ele alınmış olup bu algoritmalar hakkında bilgi verilmektedir.

2.1. İkili Dizi Hizalama

İkili hizalama algoritmaları belli dizi elemanlardan oluşan iki dizi arasındaki en iyi eşleşmeyi bulmayı amaçlar. İkili hizalama işlemi, birbirine paralel şekilde sıralanmış

iki dizinin mümkün olan en fazla sayıda özdeş elemanlarının karşı karşıya gelebilmesi için dizinin belirli sıra pozisyonlarına boşlukların (-) eklenmesi ve bu sayede boşlukların aynı tarafındaki dizi elemanlarının tek yönde kaydırılması mantığına dayanmaktadır. İki dizinin hizalandığı birden fazla durum bulunabilir. Hizalama sırasında boşlukların yerleştirileceği dizilerdeki en optimal pozisyonların belirlenebilmesi için genellikle bir puanlama yöntemi kullanılır. Bu yöntemde iki dizinin farklı hizalama durumlarını karşılık gelen puanların yer aldığı bir matris üzerinden en yüksek puanı verecek sıralama pozisyonu tercih edilerek hizalama gerçekleştirilir.

Dizi hizalamada iki farklı yaklaşım söz konusudur. Bunlar global dizi hizalama ve lokal dizi hizalama yaklaşımlarıdır. Global dizi hizalamada iki dizinin tamamında bir bütün olarak en fazla benzerliğin bulunması hedeflenmektedir. Bu hizalama sonrasında iki dizinin farklı bölümlerinde eşleşmeyen kısımlar olsa bile iki dizi bir uçtan öbür uca eşleştirilmeye çalışılır. Lokal dizi hizalamada ise dizilerin belirli bir bölümünde en yüksek benzerliğin elde edilmesi yeterlidir. Bu iki hizalama türü için Şekil 2.1'de örnek bir hizalama sonucu gösterilmiştir. Her iki hizalama yaklaşımı da farklı durum ve problemler için tercih edilebilmektedir.



Şekil 2.1. İki dizinin a) lokal hizalama sonucu b) global hizalama sonucu

İlk ikili hizalama algoritması 1970'te S. A. Needleman ve C. D. Wunsch tarafından geliştirilmiş olan Needleman-Wunsch algoritmasıdır. Bu algoritma global hizalama yaklaşımını benimser. En çok bilinen bir diğer ikili hizalama algoritması ise T. F. Smith ve M. S. Waterman tarafından 1981 yılında geliştirilen Smith-Waterman bölgesel hizalama algoritmasıdır. Bu algoritmaların ikisi de en optimal hizalama sonucunu elde etmeye çalışır. Tezin yönteminde de başvurulan Needleman-Wunsch algoritmasının işleyişi aşağıda ayrıca detaylı olarak anlatılmaktadır.

2.1.1. Needleman-Wunsch algoritması

İki dizinin tamamında en iyi eşleşmeyi elde etmek için global bir hizalama yöntemi sunan NWA (Needleman-Wunsch Algoritması) bu işlemi kullandığı bir puanlama tekniği yardımıyla gerçekleştirir. Burada amaç iki dizinin birbiriyle farklı şekilde hizalanabilecek tüm durumlar arasından en iyisini seçmektir. Bu nedenle her hizalama durumunun belli bir puan değerinin bulunması gerekir. Böylece tüm olası hizalama durumları için benzerlik miktarları puanlandıktan sonra en yüksek puana sahip olan hizalama biçimi algoritma tarafından seçilebilir.

Başlangıçta farklı uzunluktaki iki dizi, hizalama sonrasında aynı dizi uzunluğuna ulaşırlar. Çünkü hizalanmış bir durumdaki dizilerin karşılaştırılabilmesi ve puanlanabilmesi için her iki dizideki tüm elemanlar karşılıklı olarak eşleşebilmelidir. Bununla birlikte bir hizalama durumunun puanlanması işlemi iki dizinin tüm karşılıklı elemanları ayrı ayrı değerlendirilerek yapılır. Burada hizalanmış iki dizinin herhangi bir pozisyonunda bulunan karşılıklı eleman çiftleri için 3 farklı olasılık söz konusudur. Birincisi karşılaştırılan bu gözlem değerleri birbirleriyle özdeş olabilir, ikincisi bunlar farklı olabilir ya da üçüncü olarak bir gözlem değerinin karşısında boşluk (-) değeri bulunabilir. Bu üç durum için farklı puan değerleri en başta kullanıcı tarafından belirlenerek algoritmanın bu parametre değerlerine göre işlem yapması sağlanır. Daha sonra iki dizinin tüm pozisyonlarındaki her eleman çiftinin karşılaştırma puanları bir araya getirilerek hizalamanın genel skoru elde edilir. Böylece iki dizinin olası tüm hizalama durumları arasından sadece biri için benzerlik puanı elde edilmiş olur. Benzer şekilde diğer tüm olası hizalanma durumları için de benzerlik puanları hesaplanarak ve bunlar arasından en yüksek değere sahip olan hizalamanın seçilmesiyle işlem tamamlanır.

NWA hizalama işleminin adımlarını daha detaylı anlatmak amacıyla elimizde A ve B isimli iki dizinin bulunduğunu varsayalım. Bunlardan A dizisi n adet ve B dizisi ise m adet gözlem değerine sahip olsun. Burada A dizisini Eşitlik (2.1) ve B dizisini ise Eşitlik (2.2)'deki gibi matematiksel olarak ifade edebiliriz:

$$A = a_1\ a_2\ a_3 \ldots a_i \ldots a_n \quad \Leftrightarrow 1 \leq i \leq n \ \text{ve} \ i,n \in Z^+ \text{ve } a_i \in G \qquad (2.1)$$

$$B = b_1\ b_2\ b_3 \ldots\ b_j \ldots\ b_m \Leftrightarrow 1 \leq j \leq m \text{ ve } j,m \in Z^+ \text{ ve } b_j \in G \quad (2.2)$$

Burada a_i, A dizisinin; b_j, B dizisinin genel dizi elemanını ve G ise dizi elemanlarının alabileceği tüm gözlem değerlerinin kümesini gösterir. Ayrıca karşılaştırılan her dizi eleman çifti için puanlama kuralı Eşitlik (2.3)'teki gibi yazılabilir:

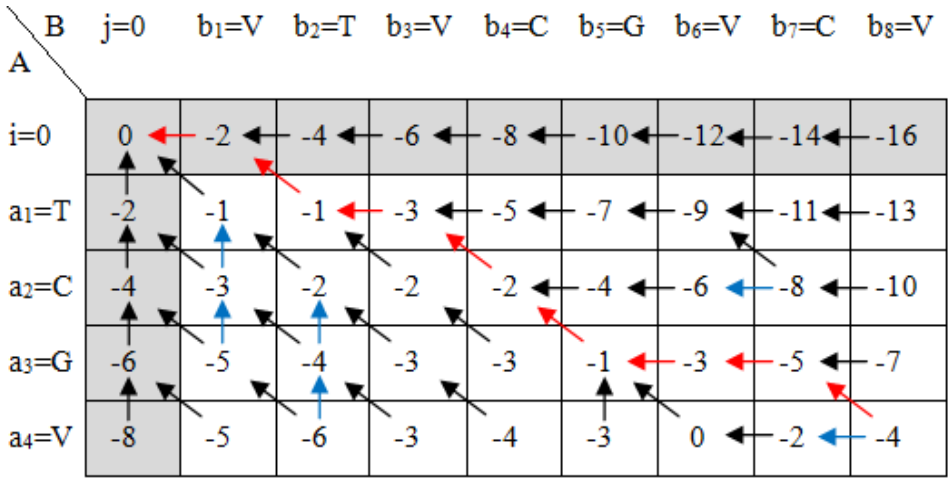
$$P(a_i,b_j) = \begin{cases} S_1 & \text{Eğer } a_i = b_j \text{ ve } i = j \\ S_2 & \text{Eğer } a_i \neq b_j \text{ ve } i = j \\ S_3 & \text{Eğer } (a_i = \text{'-'} \text{ veya } b_j = \text{'-'}) \text{ ve } i = j \end{cases} \quad (2.3)$$

Burada P, aynı dizi sıra pozisyonunda bulunan a_i ve b_j dizi elemanları çifti için karşılaştırma puan değerini temsil etmektedir. S_1 değeri a_i ve b_j nin aynı olması (eşleşme) durumunda verilecek puanı, S_2 değeri a_i ve b_j nin farklı olması (yanlış eşleşme) durumunda verilecek puanı ve S_3 ise a_i ve b_j dizi elemanlarından herhangi birinin boşlukla eşleştiği durumda verilen puana (ki buna boşluk ceza puanı da denilmektedir) karşılık gelmektedir. Ayrıca farklı değerler alabilmesinin yanı sıra S_1, S_2, S_3 değişkenleri için sırasıyla +1, -1 ve -2 değerleri çoğunlukla tercih edilmektedir. Yine aşağıda gösterilen örnek hizalamanın puan değerlerinin hesaplanmasında bu değerler kullanılmıştır.

NWA hizalama işlemi için ilk olarak (n+1) satır ve (m+1) sütun sayısına sahip boş bir M puanlama matrisi oluşturulur. İkinci olarak bu M matrisinin ilk satır ve ilk sütunundaki değerler boşluk ceza puanı kullanılarak Eşitlik (2.4)'te verilen kurala göre doldurulur:

$$M(x, y) = \begin{cases} 0 & \text{Eğer } x = 0 \text{ ve } y = 0 \\ y \times S_3 & \text{Eğer } x = 0 \text{ ve } y \neq 0 \\ x \times S_3 & \text{Eğer } x \neq 0\ y = 0 \end{cases} \quad \text{ve} \quad x,y \in N \quad (2.4)$$

Burada x değeri M matrisinin satır indeksini ve y değeri de sütun indeksini göstermektedir. Şekil 2.2'de n = 4 ve m = 8 için M matrisinin 1. satır ve 1. sütunundaki başlangıç değerleri gri renkle gösterilmiştir.

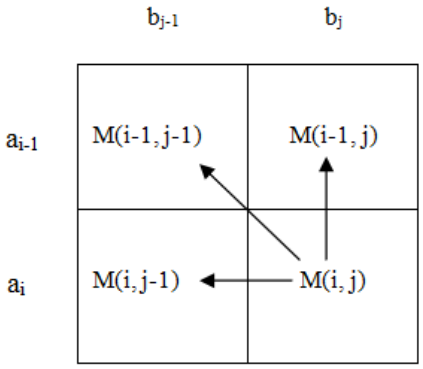


Şekil 2.2. A ve B dizilerinin n=4 ve m=8 için M puan matrisi

Bir sonraki adımda M matrisinin diğer satır ve sütunlarındaki alanlar aşağıda gösterilen Eşitlik (2.5)'deki kurala göre doldurulur;

$$M(x, y) = \max \begin{cases} M(x\text{-}1, y\text{-}1) + P(a_i, b_j) \\ M(x\text{-}1, y) + S_3 \\ M(x, y\text{-}1) + S_3 \end{cases} \quad \text{ve} \quad (x,y) \Rightarrow x = i,\ y = j \qquad (2.5)$$

burada M(x, y), $a_1\ a_2\ a_3 \ldots a_i$ ve $b_1\ b_2\ b_3 \ldots b_j$ dizilerinin en optimal hizalanma puanını gösterir. Benzer şekilde M(n, m) de bize A ve B dizilerinin global hizalama puanını verir.



Şekil 2.3. M(x, y) puan değerini bulmak için yapılabilecek olası üç geçiş

Eşitlik (2.5)'te görüleceği gibi M(x, y) üç farklı geçişe karşılık gelen puanlar arasından en büyüğünü alır. Ayrıca Şekil 2.3'de görüldüğü gibi bir alandan bir başka alana en fazla 3 farklı geçiş yapılabilir. Eğer M(x, y), (i, j) alanından (i, j-1) alanına doğru sola geçiş için maksimum değere sahipse bu durumda B dizisinin b_j elemanı A

dizisindeki boşluk (-) karakteriyle eşleşmiş yani A dizisinin j. pozisyonuna boşluk eklenmiş olur. Aynı şekilde eğer M(x, y), (i, j) alanından (i-1, j) alanına doğru yukarı yönlü geçiş için maksimum değere sahipse bu sefer de A dizisinin a_i elemanı B dizisindeki boşluk karakteriyle karşılaşır yani B dizisinin i. pozisyonuna boşluk karakteri eklenmiş olur. Üçüncü olası durum ise M(x, y)'nin hem sola hem de yukarı yönlü çapraz geçişi için maksimum değere sahip olmasıdır ki bu durumda da A dizisinin a_i elemanıyla B dizisinin b_j elemanını eşleşmiş olur. NWA, M matrisindeki tüm alanların puan değerlerini bu yolla hesaplar ve bu geçişlerin yönleriyle birlikte bu puanları hizalama işlemi bitinceye kadar bellekte tutar.

NWA artık son adımda M matrisinin sağ alt köşesinde yer alan (n, m) noktasındaki alandan başlayarak (0, 0) noktasına doğru önceki adımda oluşturulan geçiş oklarını takip eder ve çapraz geçişler dışındaki geçişler için dizilerin uygun pozisyonlarına boşluklar ekleyerek hizalama işlemini tamamlar. Şekil 2.2'deki M matris tablosunda verilen A = TCGV ve B = VTVCGVCV örnek gözlem değerleri için bu iki dizinin hizalanmasıyla birlikte nihai geçiş yolu kırmızıyla gösterilmiştir. Ayrıca siyah ve mavi oklar hesaplanan tüm alanlardaki geçiş yollarını göstermektedir. Dizilerin hizalama sonrası birbirine göre pozisyonları Şekil 2.4'te gösterildiği gibi olur ve benzerlik puan değeri ise M(4, 8) = -4 olarak bulunur.

A) – T – C G – – V

B) V T V C G V C V

Şekil 2.4. A ve B dizilerinin hizalama sonrası birbirine göre durumları

NWA'nın iyi bir hizalama sonucu vermesi S_1, S_2, S_3 parametre değerlerinin uygun aralıklarda seçilmesine bağlıdır. Dizilerin ulaşması istenen durum parametreleri pozitif istenmeyen durumlar için ise negatif değerler kullanılmalıdır. Burada istenen durum iki dizideki özdeş elemanların aynı hizaya gelmesi olduğu için eşleşme durumunu karşılayan S_1 parametresi pozitif değer almalıdır. Benzer şekilde eşleşmeme durumuna karşılık gelen S_2 ile boşluk durumunu karşılayan S_3 parametrelerine istenmeyen durumlar olduğu için negatif değerler verilmelidir. Ayrıca S_2 ve S_3 arasında da bir denge bulunmaktadır. Eğer S_2, S_3'den çok daha büyük bir parametre değeri alırsa bu durumda algoritma kaydırma için boşluk koymak

yerine yanlış eşleşmenin olduğu durumları daha fazla tercih etmeye başlar. Bu da algoritmanın hizalama etkisinin azalarak dizilerin başlangıç durumlarına yakın sonuçlar elde etmesi anlamına gelir. Yine eğer S_3, S_2'den daha büyük bir parametre değerine sahip olursa bu kez de algoritma yanlış eşleştirmelerden sakınacak ve daha fazla boşluk koymayı teşvik edecektir.

2.2. Çoklu Dizi Hizalama

Çoklu dizi hizalama, ikiden daha fazla sayıda dizinin birlikte hizalanması işlemini tanımlar. Sırayla n ve m uzunluklarına sahip iki farklı dizinin hizalanması probleminde optimal çözüm için hesaplama karmaşıklığı O(nm) ile gösterilmektedir. Bu gösterimi sadeleştirmek amacıyla her iki dizi uzunluğunun aynı olduğu varsayılırsa bu kez karmaşıklık ifadesi $O(n.n) = O(n^2)$ olacaktır. Benzer şekilde üç dizinin hizalanması işleminde bu karmaşıklık $O(7n^3)$ iken dört dizinin hizalanmasında ise $O(15n^4)$ olur. Genelleştirecek olursak n uzunluklu k adet dizinin hizalanmasında optimal çözümün bulunması için hesaplama karmaşıklığı $O((2^k-1)n^k) = O((2^k)n^k)$ olarak yazılır. Bu ifade literatürde "NP-complete" olarak geçen önemli zorluk derecesine sahip problem sınıfına karşılık gelmektedir [51]. Bu problemin doğrudan ikili hizalama algoritmalarıyla çözümü dizi sayısındaki her bir artışın üssel olarak artan miktarda fazla zaman ve donanım kaynağını gerektirdiği için pratiğe uygulanması mümkün olmamaktadır. Bu durum çoklu hizalamanın ikili hizalamadan ayrı bir problem olarak ele alınmasını gerekli kılmıştır. Bu sebeple çoklu dizi hizalama probleminin çözümü için birçok farklı yaklaşım ortaya konulmuştur. Mevcut bu yaklaşımlar kendi içerisinde tam kapsamlı (exhaustive) yaklaşımlar ve sezgisel (heuristic) yaklaşımlar olarak ikiye ayrılabilir.

Tam kapsamlı hizalama yaklaşımında en optimal çözümün elde edilmesi için tüm karşılaştırma olasılıkları değerlendirilir. Bu yöntemin ilk örnekleri n sayıdaki diziyi aynı anda hizalamak amacıyla ikili dinamik hizalama algoritmalarını kullanmışlardır. Buradaki temel mantık karşılaştırma matrisinin boyutunu dizi sayısı kadar artırmak ve böylece bu algoritmaları ikili hizalamada olduğu gibi doğrudan uygulamaktır. Ancak bu yöntem daha önce de açıklanan $O(n^k)$ hesaplama karmaşıklığı problemine sahiptir. Daha sonra benzer bir yaklaşımla çalışma süresini azaltmak için iki ayrı karşılaştırma matrisinin kullanılması düşünülse de bu kez zamandan elde edilen kazanç donanımdan kaybedilmiş ve ayrıca çalışma süresi için istenilen sonuç

alınamamıştır. Bu yaklaşımla ilgili en önemli gelişme Lipman ve ark. nın hizalama maliyetlerinin azaltılması için SP (Sum of Pairs) puanlama tekniğini kullanmaları olmuştur [52]. Bu yöntemde n adet dizi için ilk başta n(n-1)/2 miktarda dizi ikililerinin maliyetleri hesaplanmış sonrasında ise n boyutlu bir karşılaştırma matrisinde belirlenen bir üst maliyet sınırına göre çoklu hizalamaların yalnızca kısıtlı bir alanda optimal şekilde yapılması sağlanmıştır. SP puanlama tekniği daha sonra birçok farklı hizalama yönteminde dizi benzerliklerinin ölçülmesinde kullanılmıştır. Tam kapsamlı yaklaşım içerisinde bir diğer yöntem dizi maliyetlerinin belirlenmesinde ağaç yapılarının kullanılmasıdır. Bu yöntemde hizalanacak her dizi yapısal ilişkisine göre yapılandırılmış ağaç dallarının uç kısımlarında temsil edilir. Burada iki dizi arasındaki hizalama maliyeti bu dizilerin bulundukları dallar arasındaki uzaklıkla ölçülür. Bu sayede çoklu hizalamada maliyet etkinliğinin sağlanması hedeflenir [52].

Sezgisel hizalama yaklaşımı kesin yaklaşımlara göre çok daha kısa sürelerde mantıklı ve yeterli olabilecek sonuçlar elde etmeyi amaçlar. Böylece bu yaklaşımı benimseyen yöntemler en optimal sonucu bulmayı garanti etmez fakat çalışma süresini azaltarak güçlü bir performans etkinliği sunar. En iyi bilinen sezgisel hizalama yaklaşımlarını kademeli (progressive) hizalama, yinelemeli (iterative) hizalama, yaklaşık (approximate) hizalama olarak sınıflandırılabilir.

Kademeli hizalama dizilerin aynı anda değil belli bir öncelik sırasına göre aşama aşama hizalanması mantığına dayanmaktadır. Bunun için ikili hizalama algoritmaları kullanılarak diziler öncelikle kendi arasında ikili olarak hizalanır. Sonra bu ikili hizalamaların her biri için benzerlik dereceleri SP puanlama gibi puanlama teknikleri yardımıyla hesaplanır. Bulunan bu puanlara göre bir hizalama klavuz ağacı oluşturularak dizilerin çoklu hizalamaya katılma öncelikleri belirlenir. Son olarak birbirine benzerliği en yakın olandan başlayarak diziler adım adım çoklu hizalamaya dahil edilir ve tüm diziler tamamlanana kadar bu işlem devam eder. Bu yöntem hızlı olmasının yanı sıra optimal çözüme yakın mantıklı sonuçlar elde etmeyi başarsa da en büyük dezavantajı lokal hizalamalara takılmaya duyarlı olmasıdır [51].

Yinelemeli hizalama yaklaşımı kademeli hizalamadaki lokal hizalama problemiyle başa çıkmak için hizalama işleminin tekrar edilmesi esasına dayanır. Bunun için bu yaklaşımda öncelikle tüm diziler kademeli hizalamadakine benzer şekilde çoklu

hizalanır. İkinci olarak dizilerden biri çıkarılarak geriye kalan dizi profili bu çıkarılan diziyle tekrar hizalanır. Bu işlem sırayla tüm diziler için ayrı ayrı yapılır ve böylece bir tekrar (iteration) tamamlanmış olur. Her tekrar sonunda elde edilen dizi profili bir tekrar öncesindeki profille karşılaştırılarak benzerlikleri kontrol edilir. Eğer her iki profil arasındaki benzerlik miktarı belli bir eşik değerinin üstündeyse ya da başta belirlenen bir tekrar üst sınırına ulaşıldıysa hizalama işlemi durdurularak tamamlanmış olur. Eğer iki profil arasında eşik değerinden fazla bir fark varsa ve maksimum tekrar sayısına ulaşılmadıysa hizalama işlemine bu koşullardan biri sağlanana kadar devam edilir. Bu hizalama yaklaşımın ana avantajı kademeli hizalamada olduğu gibi çok fazla lokal hizalama sorunuyla karşılaşmamasıdır ancak hizalamanın başarısı büyük oranda ilk baştaki hizalama çözümüne bağlıdır.

Yaklaşık hizalama stratejisi dizilerin belirlenen bir modele göre hizalanmasını benimser. Burada belirlenen model seçilen bir veya daha fazla diziyi içerebileceği gibi başka matematiksel modellerden de oluşabilir. Bu yaklaşımın en büyük avantajı polinom süre içerisinde hizalama yaparak hızlı sonuç verebilmesidir. Bu yaklaşımda yöntemin başarısı büyük ölçüde belirlenen modele ve dizilerin bu modelle ilişkisine bağlıdır. Yaklaşık hizalama stratejisini uygulayan algoritmalar arasında en çok bilinenlerden biri merkez yıldız algoritmasıdır. Bu algoritmanın oldukça kolay bir uygulama tekniğinin olması kullanımını cazip hale getirmektedir. Tezin yönteminde de yararlanılan bu algoritmanın işleyişi hakkında detaylı bilgi aşağıda verilmektedir.

2.2.1. Merkez yıldız (Center star) algoritması

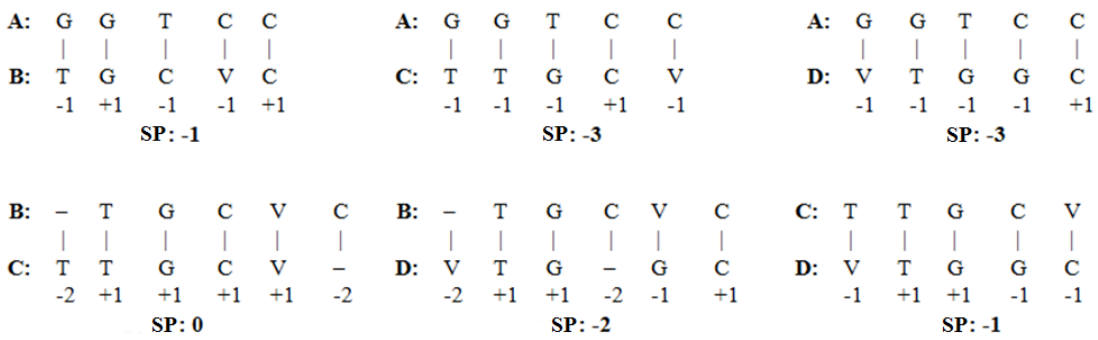
Merkez yıldız algoritması sezgisel bir çoklu dizi hizalama algoritmasıdır. Yaklaşık hizalama yaklaşımını benimseyen bu algoritma en basit ifadeyle çoklu hizalamaları ikili hizalama seviyesine indirgeyen bir yöntem sunar. Bu algoritma iki aşamadan oluşmaktadır. İlk aşama tüm diziler içerisinden diğer dizilerle en fazla benzerliğe sahip olan dizinin seçilmesi işlemidir. Seçilen bu diziye yıldız dizisi adı verilir. İkinci aşama ise geri kalan tüm dizilerin yıldız dizisine göre hizalanmasını içerir.

Hizalanan diziler arasındaki benzerliklerin puanlanabilmesi için SP puanlama, entropy puanlama ve tutarlılık (consistency) puanlama gibi farklı puanlama ölçütleri geliştirilmiştir. Bunlar arasında en çok kullanılan puanlama çeşidi SP puanlama yöntemidir. SP puanlama tekniğinin hızlı ve etkili olmasının yanı sıra kolay

uygulanabilir olması bu tekniğin popüler olmasını sağlayan başlıca faktörler arasındadır. Bununla birlikte merkez yıldız algoritmasında bu puanlama yöntemi yoğun olarak kullanılmaktadır. Bu yöntemde benzerlik puanı hesaplanmak istenen hizalı durumdaki bir dizi grubu için SP puanı her sütundaki gözlem değerlerinin ayrı ayrı karşılaştırılıp puanlanmasıyla elde edilir. İki gözlem değerinin karşılaştırılması sırasında dört farklı durum bulunabilir. Bunlar eşleşme, yanlış eşleşme, tek boşluk ve çift boşluk durumlarıdır. Çift boşluk durumu ikiden fazla dizinin hizalanması sonrasında çıkabilecek bir durumdur ve bu karşılaşma durumları için SP puanlamada 0 değeri verilir ve hesaplamada ihmal edilir. Ancak merkez yıldız algoritmasında SP puanlama yapılırken yalnızca iki dizi karşılaştırıldığından dolayı bu duruma rastlanmaz. Diğer karşılaştırma durumlarından eşleşme, yanlış eşleşme ve tek boşluk durumları için genellikle sırasıyla 1, -1, -2 değerleri çokça kullanılmaktadır. Tüm sütunların puan değerleri bulunduktan sonra bunların tamamının toplanmasıyla hizalamanın toplam puan değeri elde edilir.

Merkez yıldız algoritmasının her iki aşamasında da Needleman-Wunsch algoritması kullanılmaktadır. Birinci aşamada tüm diziler kendi arasında her dizi diğer tüm dizilerle eşleşecek şekilde n(n-1)/2 adet ikili hizalanır. Sonrasında bu hizalamaların ayrı ayrı SP puanları hesaplanır. Her dizi için diğer tüm dizilerle yapmış olduğu (n-1) tane hizalamanın SP puanları toplanarak o dizinin toplam benzerlik puanı bulunur. En son bulunan toplam puanlardan en yüksek değere sahip olan dizi yıldız dizi olarak seçilerek birinci aşama tamamlanmış olur.

Örnek olarak verilen A:(GGTCC), B:(TGCVC), C:(TTGCV), D:(VTGGC) dizileri için birinci aşamanın ikili hizalamaları sonrasındaki pozisyonları ve benzerlik puanları Şekil 2.5'te gösterilmiştir.



Şekil 2.5. Merkez yıldız algoritmasının birinci aşamasındaki ikili hizalama sonrası A, B, C, D dizilerinin puan durumu

Hizalama sonrası elde edilen SP değerleri toplanarak A, B, C, D dizilerinin her biri için toplam puan değerleri Eşitlik (2.6)'daki gibi bulunur ve B dizisi en yüksek SP değerine sahip olduğu için yıldız dizisi olarak seçilir.

$$SP_{Total}(A) = SP(A,B) + SP(A,C) + SP(A,D) \Rightarrow SP_{Total}(A)=(-1)+(-3)+(-3)= -7$$

$$SP_{Total}(B) = SP(A,B) + SP(B,C) + SP(B,D) \Rightarrow SP_{Total}(B)=(-1)+0+(-2)= -3 \quad (2.6)$$

$$SP_{Total}(C) = SP(A,C) + SP(B,C) + SP(C,D) \Rightarrow SP_{Total}(C)=(-3)+0+(-1)= -4$$

$$SP_{Total}(D) = SP(A,D) + SP(B,D) + SP(C,D) \Rightarrow SP_{Total}(D)=(-3)+(-2)+(-1)= -6$$

İkinci aşamada belirlenen yıldız dizisi ile geriye kalan tüm diziler sırayla tekrar hizalanarak çoklu hizalamaya dahil edilir. Ancak bu aşamada hizalama yapılırken özel bir durumu söz konusudur. Bu özel durum yıldız diziyle sıradaki dizi ikili olarak normal hizalanırken yıldız dizide bir boşluk eklenmesi gerektiğinde ortaya çıkar. Bu noktada eklenmesi gereken boşluk sadece yıldız dizisinin ilgili pozisyonuna değil aynı zamanda o ana kadar hizalanmış tüm dizilerin de aynı pozisyonlarına boşluk eklenir. Bu kurala "Bir kere bir boşluk devamlı bir boşluk" prensibi de denilmektedir [53]. Böylece o zamana kadar hizalanmış dizilerin hizalı pozisyonları korunmuş olur. Yıldız diziyle hizalanan sıradaki diziye bir boşluk eklenmesi gerektiğinde ise normal ikili hizalamada olduğu gibi sadece o diziye eklenir. Şekil 2.6'da A, B, C, D dizilerinin merkez yıldız algoritmasıyla hizalamasının ikinci aşamadaki adımları ve hizalama sonucu gösterilmiştir.

Şekil 2.6. A, B, C, D dizilerinin merkez yıldız hizalamasındaki ikinci aşama adımları ve hizalanmış son durumları

Yukarıdaki örnekte dikkat edilirse önce yıldız seçilen B dizisiyle sıradaki A dizisi normal bir şekilde ikili hizalanmıştır. Sonrasında sıradaki C dizisi yıldız diziyle hizalanırken Şekil 2.6'da kırmızıyla gösterilen ve yıldız dizisinde kaydırma yapmak için gerekli olan boşluklar hem B hem de A dizilerine eklenmiştir. Böylece A dizisindeki ilk hizalama pozisyonları bu sayede korunmuştur. Diğer yandan C

dizisinde kaydırma yapmak için gereken boşluk ise sadece bu diziye eklenerek üç dizinin hizalanması tamamlanmıştır. Son olarak D dizisi de C dizisine benzer şekilde yıldız dizisiyle hizalanarak çoklu hizalama tamamlanmış olur.

3. MARKOV MODELLERİ

Adını Rus Matematikçi Andrey Andreyevich Markov'dan alan Markov yöntemi zaman içerisinde oluşan ve her biri sistemin bir t anındaki çıktısına karşılık gelen sıralı gözlem değerlerinin ilişkisini tanımlamak için bize stokastik bir model sunar. Bu yöntem modellenecek veri kümesindeki ardışık gözlem değerlerinin birbirinden tamamen bağımsız olmadığını farzederek onların olasılıksal bir süreç içerisinde meydana geldiğini kabul eder [54]. Burada bir veri dizisi için her gözlem değerinin yalnızca kendinden bir önceki gözlem değerine göre belirlendiği ancak daha önceki gözlem değerlerinin bir etkisinin olmadığı varsayımı temel alınır. Buna sürecin Markov özelliği denildiği gibi bir durumun kendinden daha önceki durumları hatırlamaması nedeniyle hafızasızlık özelliği olarak da adlandırılmaktadır. Dolayısıyla her gözlem değeri kendinden bir sonraki gözlem değerini belirleyen bilgiye sahip olacak şekilde tıpkı bir zincirin bağımsız halkaları gibi ikili bağlantılarla birbirine bağlanırlar. Bu yüzden sayılabilen durum elemanlarından oluşan kesikli durumlu süreçlere aynı zamanda Markov zincirleri adı verilmiştir. Öte yandan sayılamayan durum elemanlarından oluşan Markov süreçlerine ise sürekli durumlu Markov süreci denilmektedir [55].

Markov modellerinde bir veri dizisinin gözlem değerleri birbirine bağlı durum yapıları ve bunlar arasındaki olasılık fonksiyonlarıyla ilişkilendirilir. Buna göre modeldeki her gözlem durumu Eşitlik (3.1)'de gösterildiği gibi yalnızca kendinden önceki duruma bağlıdır.

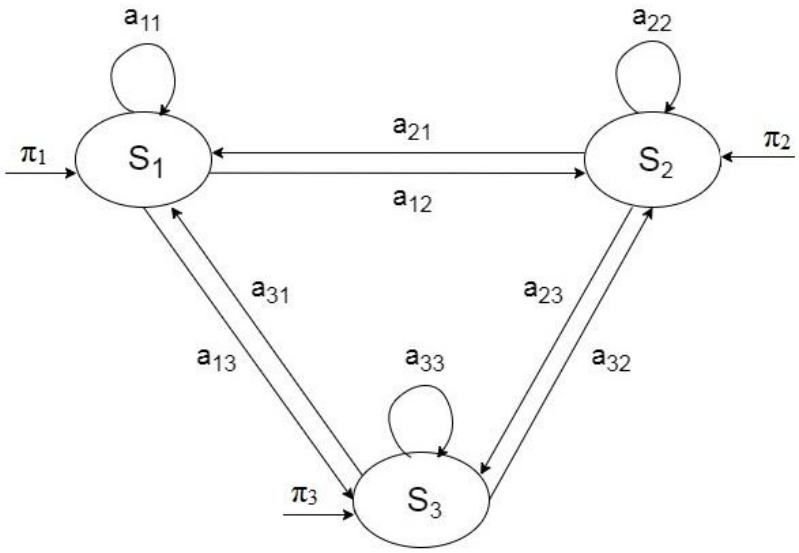
$$P\left(q_{t+1}= S_j \mid q_t= S_i, q_{t-1}= S_k, \ldots\right) = P\left(q_{t+1}= S_j \mid q_t= S_i\right) \tag{3.1}$$

Burada S_i, N tane sonlu elemandan oluşan bir $S=\{S_1, S_2, S_3, \ldots, S_N\}$ durum kümesinin i. durumunu ve q_t ise bir t anında bulunulan durumu ($q_t = S_i$) temsil etmektedir. Ayrıca $P(q_t)$, t anında q_t durumunun gerçekleşme olasılığını ifade eder.

$P(q_t)$, t anındaki S_i durumundan t+1 anında bulunan S_j durumuna geçiş olasılık değerini bize vermektedir ve a_{ij} ile gösterilir. Geçiş olasılık değerleri Eşitlik (3.2)'deki gibi hesaplanmaktadır;

$$a_{ij} = P(q_{t+1} = S_{i+1} \mid q_t = S_i) = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T-1} (q_t^k = S_i \text{ and } q_{t+1}^k = S_j)}{\sum_{k=1}^{K} \sum_{t=1}^{T-1} (q_t^k = S_i)} \quad (3.2)$$

Burada a_{ij} olasılık değeri model üzerinde S_i durumundan S_j durumuna geçiş sayısının S_i durumundan diğer tüm durumlara yapılan toplam geçiş sayısına bölünmesiyle elde edilir. Şekil 3.1'de verilen üç durumlu bir Markov zincir yapısı için örnek diyagram üzerinde durumlar arası geçişler gösterilmiştir.



Şekil 3.1. Üç durumlu Markov zincirinde durumlar arası geçişleri ve başlangıç olasılıklarını gösteren örnek model diyagramı

Diyagramdaki her düğüm bir durumu, durumları birbirine bağlayan tek yönlü oklar ise geçişleri temsil eder. Durumlar arasındaki geçiş olasılıkları a_{ij} ile gösterilirken π ise bulunduğu durumun başlangıç olasılık değerini belirtmektedir. Dikkat edilirse üç durumlu bu modelde her durum için toplamda üç geçiş bulunmaktadır. S_1 için bu durumdan S_1, S_2, S_3 durumlarına geçiş olasılıkları sırasıyla a_{11}, a_{12}, a_{13} değerlerinden oluşurken; S_2 için S_1, S_2, S_3 durumlarına geçiş olasılıkları sırasıyla a_{21}, a_{22}, a_{23} değerlerinden; S_3 için ise S_1, S_2, S_3 durumlarına geçiş olasılıkları sırasıyla a_{31}, a_{32} ve a_{33} değerlerinden oluşmaktadır. Yalnızca 0 veya pozitif bir değere sahip olabilen geçiş olasılıkları için aynı durumdan çıkan tüm olasılık değerlerinin toplamı Eşitlik (3.3)'de gösterildiği gibi 1'e eşit olmalıdır.

$$\sum_{j=1}^{N} a_{ij} = 1 \text{ ve } a_{ij} \geq 0 \quad (3.3)$$

Benzer şekilde N durumlu bir modelde başlangıç olasılıklarının toplamı Eşitlik (3.4)'de belirtildiği gibi 1' e eşit olmalıdır.

$$\sum_{i=1}^{N} \pi_i = 1 \text{ ve } \pi_i = P(q_1 = S_i) \tag{3.4}$$

Modellenmesi istenen M adet veri dizisi için her durumun başlangıç olasılıkları bu dizilerden kaç tanesinde ilk gözlem değerinin ilgili durumdan başladığı sayılarak elde edilen değerlerin toplam dizi sayısına bölünmesiyle Eşitlik (3.5)'teki gibi hesaplanır.

$$\pi_i = \frac{\sum_{m=1}^{M} (q_{m_1} = S_i)}{M} \tag{3.5}$$

Markov modellerinin bir bölümünde durumlar doğrudan gözlemlenebilir olmasına karşın Markov modellerinin tümü için aynı şey söylenemez. Durumların doğrudan gözlemlenemediği yapıdaki verilerin modellenmesi için SMM (Saklı Markov Model) leri geliştirilmiştir. Bu bölümün devamında SMM'ler hakkında bilgi verilmektedir.

3.1. Saklı Markov Modeli

Markov modellerinin özelleşmiş bir biçimi olan Saklı Markov Modelleri'nde bir t anında üzerinde olunan durum bilinmemektedir. Bunun yerine yalnızca sistemin çıktıları gözlem değerleri olarak izlenebilir. Bir başka deyişle SMM'de bir gözlem değerinin hangi duruma karşılık geldiğini klasik Markov modellerinde olduğu gibi sadece gözlem değerlerine bakarak doğrudan anlamak mümkün değildir. Çünkü SMM'de aynı durum yapısında farklı gözlem değerleri görülebilir (Şekil 3.2). Dolayısıyla SMM'lerde aynı gözlem dizisini karşılayan birden farklı durum dizileri bulunabilir.

Kesikli yapıdaki bir Saklı Markov Modeli aşağıdaki yedi maddeyle ifade edilebilir:

- Modelde kullanılan farklı durumların kümesi ve modeldeki toplam durum sayısı;
  $S = \{S_1, S_2, \ldots, S_N\}$, $s(S) = N$
- Farklı gözlem (çıktı) değişkenlerinin kümesi ve toplam değişken sayısı;
  $V = \{v_1, v_2, ..., v_M\}$, $s(V) = M$
- Bir T süreci boyunca gerçekleşen ardışık durumlar dizisi;
  $Q = q_1\, q_2 \ldots\, q_T \Rightarrow q_t \in S$ ve $t \in T$

- Bir T sürecinde izlenen gözlem elemanları dizisi;

  $O = o_1\ o_2 \ldots\ o_T \Rightarrow o_t = v_k\ ;\ 0< k < M;\ k \in Z^+;\ t \in T$

- Başlangıç durum olasılıklarının kümesi;

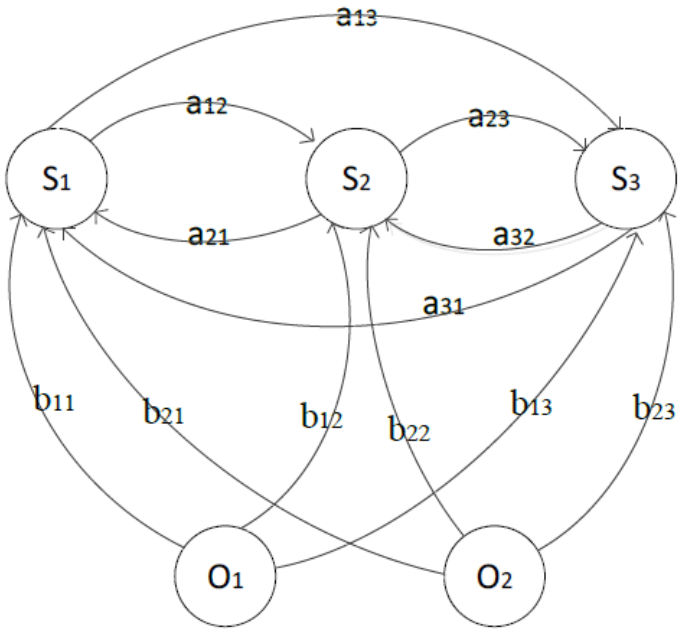
  $\Pi = \{\ \pi_i \mid 0< i < N,\ i \in Z^+ \text{ve } \pi_i \equiv P(q_1 = S_i)\}$

- Geçiş olasılıkları dağılımı;

  $A = \{a_{ij} \mid a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i) \text{ ve } 0< i, j < N;\ i, j \in Z^+\}$

- Durumlarda salınan gözlem olasılıkları kümesi;

  $B = \{b_j(k) \mid b_j(k) \equiv P(o_t = v_k \mid q_t = S_j) \text{ ve } 0< k < M;\ k \in Z^+\ \}$



Şekil 3.2. Örnek bir SMM modelinde durumlar arası geçişler ve gözlem değerlerinin durumlarla ilişkisi

Bir SMM yapısı, $\lambda = (A, B, \pi)$ genel ifadesi ile gösterilebilir. Burada A, geçiş olasılıklarını; B, varılan durumlardan salınan gözlem olasılıklarını ve π ise başlangıç olasılıklarını temsil etmektedir.

SMM'nin çözüm aradığı üç temel problem vardır. Bunlar;

1. Bir gözlem dizisini gerçekleyecek en yüksek olasılıklı durum dizisinin elde edilmesi,
2. Modelde bir gözlem dizisi için gerçekleşme olasılığının hesaplanması,
3. Bir eğitim veri kümesi aracılığıyla model parametrelerinin eğitilmesi problemleridir.

Bu problemlerden ilkinin çözümünde Viterbi algoritması, ikinci problem için Forward algoritması, üçüncü problemin çözümü içinse Forward-Backward algoritması, Expectation Maximization ya da Baum-Welch algoritması ile birlikte genellikle kullanılmaktadır [3]. Tezde birinci probleme çözüm bulunmaya çalışıldığı için ikinci ve üçüncü problemin çözümüne fazla değinilmeyecektir.

En yüksek olasılıklı durum dizisinin elde edilmesi probleminde T süresi boyunca bir $O = o_1\ o_2\ o_3 \ldots\ o_T$ gözlem dizisinin görülebilmesi için en yüksek olasılığa sahip $Q = q_1\ q_2\ q_3 \ldots\ q_T$ durum dizisinin ve bu dizinin gerçekleşme olasılığının bulunması hedeflenir. Bu olasılık değeri Viterbi algoritması kullanılarak ve koşullu olasılık formülü $P(O \mid \lambda)$ yardımıyla elde edilir [56]. Bunun için belli bir duruma kadar hesaplanan olasılık değerlerinin tutulması amacıyla $\delta_t(i)$ olarak gösterilen bir ara değişken tanımlanır. Bundan sonra bir gözlem elemanının kısmi olasılık değerini bulmak için gözlem dizisinde bulunması istenen gözlem elemanına ulaşıncaya kadar izlenen yolların olasılıkları ile bu yollar boyunca durumlardan salınan gözlem elemanlarının salınma olasılık değerleri öz tekrarlı (recursive) bir şekilde çarpılır. Burada kısmi olasılıkların hesaplanmasında iki durum mevcuttur. Bunlardan birincisi $t = 1$ anındaki olasılık değeridir ki bu noktada gözlenen bir yol olmadığı için $\delta_0(i)$ değeri 1'e eşittir ve bu yüzden sadece başlangıç olasılık değeri ile gözlem elemanının olasılık değerleri çarpılarak Eşitlik (3.6)'daki gibi hesaplama yapılır.

$$\delta_1(i) = \pi_i b_i(O_1) \tag{3.6}$$

$t > 1$ koşulunu sağlayan ikinci durumda ise daha önce gözlenen bir yol bulunduğu için $\delta_{t-1}(i)$ değeri de çarpımlara dahil edilerek Eşitlik (3.7)'deki gibi olasılık değerleri hesaplanarak maksimum olasılık değeri alınır.

$$\delta_t(j) = \max_i(\delta_{t-1}(i)a_{ij})b_j(O_t) \tag{3.7}$$

N uzunluğundaki bir gözlem dizisinin son gözlem değerinin olasılık değeri $\delta_T(i)$ Eşitlik (3.8)'deki gibi hesaplandığında bu dizideki tüm gözlem değerleri için görülme olasılığı bulunmuş olur.

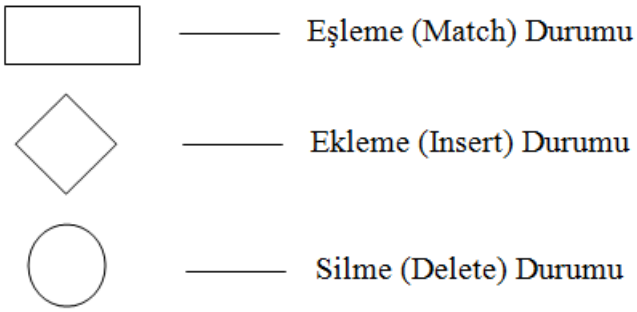
$$P(O \mid \lambda) = \max(\delta_T(i)) \tag{3.8}$$

İkinci problemin çözümünde kullanılan Forward algoritmasının Viterbi algoritmasından tek farkı belli bir gözlem değerine kadar olan önceki yolların olasılıklarını hesaplarken maksimum olanı seçmek yerine bu muhtemel yolların olasılık değerleri için tümünün toplamını almasıdır.

Biyoinformatikte ve zaman serileri özelliği gösteren verilerin modellenmesinde çokça kullanılan ve SMM'nin özel bir uzantısı durumundaki Profile Hidden Markov Model (PHMM)'ler tez yönteminde de kullanılmıştır. Bir sonraki bölümde PHMM'lerin yapısı ve Viterbi algoritmasının bu modele uygulanması hakkında bilgi verilmektedir.

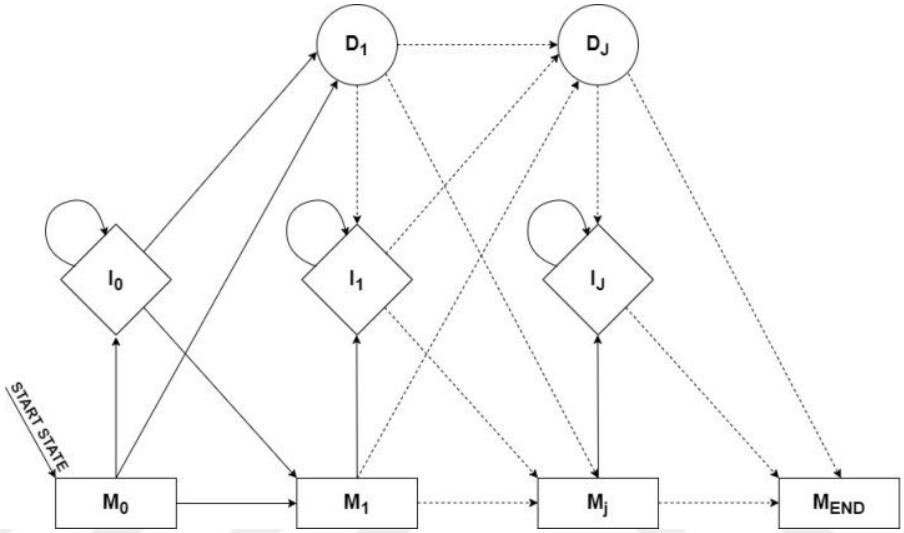
3.2. Profile Hidden Markov Model

Çoklu dizilerin modellenmesinde diğer SMM'lere göre daha avantajlı bir yöntem olarak Profile Hidden Markov Modeli ilk defa Krogh ve ark. tarafından önerilmiştir. PHMM çok sayıdaki dizilerin bir ortak diziden türediği varsayımına dayanarak bu ata diziyi elde etmeyi amaçlayan bir yaklaşıma sahiptir. PHMM'nin SMM'den en önemli farkı PHMM'nin durumları arasındaki geçişlerin her zaman ileri yönlü olmasıdır, geriye gidiş yoktur. Bu yüzden PHMM'deki diziler gözlem elemanlarının sırasına bağımlıdırlar. Bu durum PHMM'nin SMM'den farklı olarak bir başlangıç ve bir de bitiş durumlarına sahip olmasının da nedenini oluşturur. Ayrıca PHMM'de her adım için üç farklı durum yapısı bulunmaktadır. Bunlar ekleme, eşleme ve silme durumlarıdır. Bu durumları temsil eden diyagram gösterimleri Şekil 3.3'de verilmiştir.



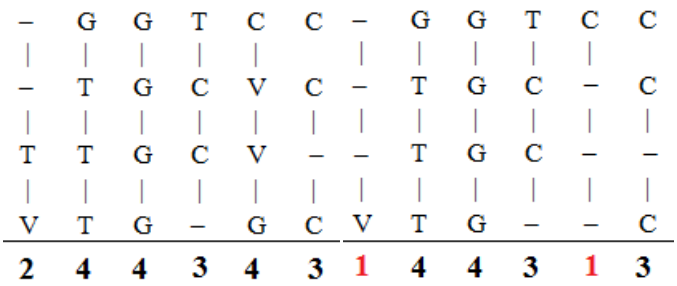
Şekil 3.3. PHMM'deki farklı türdeki durumların şekilsel gösterimleri

Eşleme durumu hizalanmış dizilerin belli bir kritere göre seçilen sütunlardaki gözlem elemanları için kullanılır ve her durumda yalnızca bir gözlem elemanı bu duruma karşılık gelebilir. Ekleme durumu eşleme durumunun dışındaki sütunlarda bulunan gözlem elemanları için gereklidir. Silme durumu ise eşleme durumunda herhangi bir gözlem değerinin gelmemesini yani boşluk (-) değerini almasını ifade eder. Bu durumda herhangi bir çıktı gözlenmez ancak diğer durumlara geçiş yapmaya izin verir. Şekil 3.4'de HMM'nin durum yapısını gösteren bir diyagram verilmiştir.



Şekil 3.4. PHMM'deki durum yapıları ve bunlar arasındaki geçişler

Kullanılacak PHMM modelini tasarlarken ilk karar verilmesi gereken parametre modelin uzunluğudur. Burada eşleme durumlarının sayısı aynı zamanda bize modelin uzunluğunu verir. Çünkü PHMM'de eşleme durumları arasındaki geçişler başlangıç durumu (M_0) ile başlar ve bitiş durumu (M_{End}) ile sonlanacak biçimde tek yönlü olarak ilerler. Eşleme durumlarının sayısını belirlerken öncelikle modellenmesi istenen hizalanmış diziler Şekil 3.5'de gösterildiği gibi alt alta satırlar halinde yazılır.



Şekil 3.5. PHMM'de eşleme durum sayısı için dizi sütunlarının örnek bir görüntüsü

Sonrasında her dizinin aynı pozisyonundaki gözlem değerlerini içeren sütunlar ayrı ayrı değerlendirilir. Daha önce belirlenen bir eşik değerine göre bu sütunlardan az sayıda boşluk değeri içerenler eşleme durumu olarak kabul edilir. Farklı değerler alabilmesine karşın genellikle 0,5 boşluk oran değeri eşik değeri olarak kullanılmaktadır. Şekil 3.5'de 0,5 eşik değerine göre soldan 7. ve 11. sütundaki boşluk sayıları ikiden az olduğu için eşleme durumlarına dahil edilmez ve bu sütundaki gözlem değerleri ekleme durumlarında kullanılırlar.

Durum yapıları oluşturulduktan sonra modellenecek dizilerin PHMM'ye eğitim verisi olarak verilip modele öğretilmesi gerekmektedir. Bu aşamada her dizi için gözlem elemanları sırayla modelin ilgili durum yapısındaki olasılık hesabına dahil edilerek tüm durumların olasılık değerleri güncellenir ve bu işlem dizilerin tamamı modele aktarılana kadar devam eder. PHMM'de bu işlem yapılırken Eşitlik (3.9) ve Eşitlik (3.10)'da formülleri verilen iki ayrı olasılık hesaplaması kullanılır (Durbin, 1998).

$$a_{ij}=\frac{A_{ij}}{\sum_{j'} A_{ij'}} \tag{3.9}$$

$$e_i(x)=\frac{E_i(x)}{\sum_{x'} E_i(x')} \tag{3.10}$$

Eşitlik (3.9)'da i durumundan j durumuna geçiş olasılığı hesaplanmaktadır. Burada a_{ij} geçiş olasılık değerini, $A_{ij,}$ i durumundan j durumuna yapılan geçiş sayısını, j' gösterimi ise i durumundan kendisine geçiş yapılabilen muhtemel tüm durumları temsil etmektedir. Böylece i durumundan j durumuna yapılan geçişlerin sayısı, i durumundan yapılan toplam geçiş sayısına bölünerek geçiş olasılık değeri bulunmuş olur. Eşitlik (3.10)'da x gözlem değerinin i durumundaki çıktı olasılık değeri hesaplanır. Burada $E_i(x)$, x gözlem değerinin i durumundan görülme sayısına, $\sum_{x'} E_i(x')$ ifadesi ise i durumunda gerçekleşen toplam gözlem sayısını karşılık gelir.

PHMM'de N adet durum için geçiş olasılık değerlerinin tutulması ve böylece gereksiz tekrarlı hesaplamaların yapılmasını önlemek amacıyla N x N boyutlu bir durum matrisi kullanılır. İlk satır ve sütununda durum isimlerinin yer aldığı bu matris üzerinde olasılık değerleri dinamik olarak hesaplanarak saklanır.

Durumlarda geçiş ve gözlem olasılık değerleri hesaplanıp modele öğretildikten sonra belli bir gözlem sıralısına karşılık gelen en yüksek olasılığa sahip durumu bulmak için Viterbi algoritması kullanılır. Viterbi algoritmasının PHMM'de nasıl uygulandığı sonraki bölümde anlatılmaktadır.

3.2.1. PHMM'de Viterbi algoritması

Viterbi algoritması, bir veri modeli üzerinde belli bir çıktıya karşılık gelen farklı durum dizilerinin içerdiği yollar arasından en yüksek olasılığa sahip olanı bulmayı amaçlayan bir çözümleme algoritmasıdır [57]. Genel olasılık denklemleri Bölüm 3.1'de verilmiş olan Viterbi algoritmasının PHMM'de üç farklı türdeki durum yapısı için uygulama adımları Eşitlik (3.11)'de gösterilmiştir:

$$\delta_{M_i}(t)=e_{M_i}(o_t)\max\begin{cases}\delta_{M_{i-1}}(t-1)a_{M_{i-1}M_i}\\ \delta_{I_{i-1}}(t-1)a_{I_{i-1}M_i}\\ \delta_{D_{i-1}}(t-1)a_{D_{i-1}M_i}\end{cases}$$

$$\delta_{I_i}(t)=e_{I_i}(o_t)\max\begin{cases}\delta_{M_i}(t-1)a_{M_iI_i}\\ \delta_{I_i}(t-1)a_{I_iI_i}\\ \delta_{D_i}(t-1)a_{D_iI_i}\end{cases} \qquad (3.11)$$

$$\delta_{D_i}(t)=\max\begin{cases}\delta_{M_{i-1}}(t)a_{M_{i-1}D_i}\\ \delta_{I_{i-1}}(t)a_{I_{i-1}D_i}\\ \delta_{D_{i-1}}(t)a_{D_{i-1}D_i}\end{cases}$$

Burada $\delta_{M_i}(t)$, t anındaki M_i eşleşme durumuyla biten en yüksek olasılıklı yolu ve olasılık değerini göstermektedir. Benzer şekilde $\delta_{I_i}(t)$, t anındaki I_i ekleme durumuyla biten en yüksek olasılıklı yolu ve olasılık değerini; $\delta_{D_i}(t)$ ise t anındaki D_i silme durumuyla biten en yüksek olasılıklı yolu ve olasılık değerini göstermektedir.

PHMM'de Viterbi algoritması uygulanırken t = 1 için başlangıç durumunun (M_0) olasılık değeri 1 alınır. Bu PHMM'nin tek yönlü model yapısından dolayı her zaman yalnızca bir başlangıç ve bir bitiş durumuna sahip olmasından ileri gelmektedir. Dolayısıyla π başlangıç olasılık değeri PHMM'de sadece başlangıç durumu M_0 için 1 olarak diğer durumlar için 0 kabul edilir. t > 1 için Eşitlik (3.11) öz yinelemeli bir

şekilde uygulanarak tüm yollar için olasılık değerleri çıkarılır ve maksimum olasılık değerine sahip olan yol seçilerek sonuç elde edilir.

4. VERİ VE YÖNTEM

Tezin önceki bölümlerinde yöntemde yararlanılan algoritmalar ve yaklaşımlar tanıtılarak detaylı bir şekilde ele alınmıştı. Bu bölümde öncelikle kullanılan veri kümesi ve özellikleri hakkında bilgiler verilmekte sonrasında ise kullanılan yöntemin uygulama aşamaları anlatılmaktadır.

4.1. Veri

Tez yönteminin gerçeklenmesi amacıyla yapılan testlerde Weeplaces veri kümesi kullanılmıştır. Weeplaces; Facebook Places, Foursquare ve Gowalla gibi diğer konum bazlı sosyal ağ servislerinin uygulama API'lerine entegre edilerek kullanıcı aktivitelerinin görselleştirilmesini amaçlayan çevrimiçi bir platformdur. Bu veri kümesi 15.799 kullanıcı tarafından 971.309 konum üzerinden oluşturulmuş ve 200 ü aşkın farklı seviyedeki kategoriler altında etiketlenmiş 7.658.368 rezervasyon kayıt verisine sahiptir. Her kayıt satırı 7 farklı sütuna ayrılmış özellik bilgisine sahiptir. Özellikler kullanıcı adı (userid), mekân adı (placeid), rezervasyon zamanı (datetime), rezervasyon yerinin enlemi (lat), rezervasyon yerinin boylamı (lon), rezervasyon yerinin bulunduğu şehir (city) ve rezervasyonun kategorisi (category) bilgilerinden oluşmaktadır. Weeplaces veri kümesindeki kayıt satırlarının örnek ekran görüntüsü Şekil 4.1'de verilmiştir.

Weeplace_checkins

userid	placeid	datetime	lat	lon	city	category
joselin-mane	apple-store-boylston-street-boston	2009-08-13T18:58:36	42.348821	-71.082369	Boston	Shops:Technology
ryan-romero	thai-select-new-york	2009-08-13T18:58:44	40.754769	-73.994896	New York	Food:Thai
jennifer-newell	noodles-company-pearl-boulder	2009-08-13T18:59:08	40.021828	-105.258939	Boulder	
craig-villamor	roli-roti-san-francisco	2009-08-13T18:59:57	37.7955	-122.3937	San Francis	Food:Food Truck / Street Food
joselin-mane	hynes-convention-center-boston	2009-08-13T19:00:23	42.3474120215	-71.0840320587	Boston	Home / Work / Other:Convention Center
joselin-mane	prudential-center-tower-boston	2009-08-13T19:00:58	42.3471424212	-71.0825300217	Boston	Home / Work / Other:Corporate / Office
andrew-cafourek	bryant-park-new-york	2009-08-13T19:01:44	40.7537591455	-73.9836072922	New York	Parks & Outdoors:Plaza / Square
joselin-mane	shops-at-prudential-center-boston	2009-08-13T19:02:02	42.3472375744	-71.0813498497	Boston	Shops:Mall
sanjay-kairam	parc-palo-alto-research-center-palo-alto	2009-08-13T19:03:07	37.402483	-122.147372	Palo Alto	Home / Work / Other:Corporate / Office
nick-cupo	duboce-park-cafe-san-francisco	2009-08-13T19:04:32	37.7693	-122.431	San Francis	Food:Coffee Shop
benjamin-kudria	mexico-au-parc-san-francisco	2009-08-13T19:05:20	37.7821798958	-122.393586785	San Francis	Food:Burritos
drew-christien	hrp-new-york	2009-08-13T19:05:41	40.7236	-74.008	New York	
powen-shiah	la-colombe-torrefaction-philadelphia	2009-08-13T19:06:00	39.9507280407	-75.1722550392	Philadelphi	Food:Coffee Shop
patrick-haney9	starbucks-new-york	2009-08-13T19:06:42	40.7416496	-73.9902791	New York	Food:Coffee Shop

Şekil 4.1. Weeplaces veri kümesinden rezervasyon kayıtlarını gösteren örnek bir ekran görüntüsü

Uygulama sürelerinin fazla olması ve çok büyük verilerin yönetimindeki zorluklar sebebiyle deneysel testler Weeplaces veri kümesinin ilk 1.000.000 kayıt verisi

üzerinde yapılmıştır. Ayrıca kullanıcı davranış ve rutinlerinin günün farklı zamanlarına göre değişiklik göstermesi nedeniyle bu kayıtlar günün belli saat aralıklarına göre ayrılarak 6 ayrı alt veri kümesi elde edilmiştir. Bu veri kümeleri bir günlük zaman periyodu içerisindeki 00:00–08:00, 08:00–12:00, 12:00–14:00, 14:00–18:00, 18:00–20:00, 20:00–24:00 saat dilimlerini karşılayan rezervasyon kayıtlarını içerecek şekilde oluşturuldu. Oluşturulan tüm veri kümelerinde rezervasyonlar 10 farklı kategoriden birine ait olup her kullanıcının rezervasyon sayısı 100 olarak belirlenmiştir. Bunun için her kullanıcının sadece o veri kümesindeki zamana göre ilk 100 kayıt verisi alınmıştır. Yöntemin veri önişlemi adımında gerçekleştirilen bu işlemi bir sonraki bölümde anlatılmaktadır. Bu veri kümesinin sayısal özellikler Tablo 4.1'de gösterilmektedir.

Tablo 4.1. Testlerde kullanılan veri kümelerinin toplam kayıt ve kullanıcı sayıları

Veri Kümesi	Kullanıcı Sayısı	Kayıt sayısı (check-in)
w_00-08	822	82200
w_08-12	195	19500
w_12-14	177	17700
w_14-18	666	66600
w_18-20	359	35900
w_20-24	716	71600

4.2. Yöntem

Yöntemin amacı kullanıcıların bir sonraki rezervasyon tercihinin hangi kategoriye ait olacağını tahmin etmektir. Tez yönteminin uygulama aşamaları ve algoritma yapısının örnek gösterimi Şekil 4.2'de verilmiştir.

Yöntemin uygulanması için tasarlanan uygulamanın altı adet ana girdisi bulunmaktadır. Bunlar;

- Kullanıcıların sıralı rezervasyon verileri (V),
- İkili dizi hizalama işlemindeki sırasıyla eşleşme, yanlış eşleşme ve boşluk durumlarına karşılık gelen karşılaştırma puan parametreleri (s1, s2, s3),

- Sonraki kategori değeri tahmin edilecek olan test kullanıcısının tercih dizisi (td),
- Test kullanıcı dizisinin modele öğretilmesi için verilen parçasını içeren eğitim yüzde oranı (ey),
- Modelleme için seçilen maksimum kullanıcı sayısı (ks),
- Test kullanıcısının sonraki tercihini tahmin etmek için kullanılan son tercih sayısı, (ts) değerleridir.

Algorithm Yöntem

```
1: function KullaniciSonrakiTercihTahmini(V, S(s1, s2, s3), td, ey, ks, ts)
2:     F = VeriÖnişlem (V)
3:     K = KullanıcıSeçimi (F, S, td, ey, ks)
4:     H = ÇokluHizalama (K, S)
5:     M = PSMModelleme (H)
6:     T = TercihTahmini (M, td, ts)
7:     return T
```

Şekil 4.2. Yöntem aşamalarının genel algoritma gösterimi

Kullanılan yöntem beş temel aşamadan oluşmaktadır:

1- Veri Önişlemi: Bu aşamada kullanılan verinin uygulanacak yöntem için en uygun hale getirilmesi amaçlanmaktadır. Bu aşama kendi içerisinde üç adımdan oluşmaktadır.

a- Veri temizleme ve restorasyon: Bu adımda ilk başta Weeplaces veri kümesindeki her rezervasyon kaydının kategori bilgisine sahip olup olmadığı kontrol edilir. Kategori bilgisi bulunmayan kayıtların test sonuçları için doğruluklarının değerlendirilmesi mümkün olamayacağı için öncelikle bu kayıtlar temizlenir. İkinci olarak kayıtlardan bazı özellik bilgileri kaybolmuş veya farklı biçimlerde yazılmış olanlar kurtarılmaya çalışılarak kullanılabilecek ortak bir forma dönüştürülür.

b- Kategori organizasyonu: Burada farklı isme sahip fakat aynı içerikteki kategorilerin tek bir kategori etiketi altında birleştirilmesi işlemi yapılır. Ayrıca aynı üst kategorinin altında yer alıp farklı seviyelerde farklı alt kategorilere ait olan tüm kayıtlar sadece tek bir üst kategori altında birleştirilerek tek seviyede kategorilendirilir. Bu işlem sonucunda (Home/Work/Other, Arts & Entertainment, Food, Travel, Parks & Outdoors, Nightlife, Shops, College & Education, Great Outdoors, Colleges & Universities) 10 kategori altında tüm kayıtlar etiketlenmiştir.

c- Sıralama: Önişlemin bu bölümünde her kullanıcının rezervasyon kayıtları günün belli periyotlarına göre filtrelenerek 6 farklı alt veri kümesi oluşturulur ve her bir veri kümelerindeki kullanıcı kayıtları kronolojik olarak sıralanır. Daha sonra sıralanmış rezervasyon kayıtlarının ilk 100 tanesi için sadece kategori değerleri çekilerek aynı sırayla tek bir satırda birleştirilir. Böylece her satır farklı bir kullanıcıya ait olan ve eşit sayıdaki kategori gözlem değerlerinden oluşan kullanıcı dizileri elde edilir. Bu işlem 6 ayrı veri kümesi için gerçekleştirilir.

2- Kullanıcı Seçimi: Bu aşamada model için test kullanıcısına en benzer belli sayıdaki kullanıcıların seçilmesi hedeflenir. Bu amaçla önce tüm kullanıcı dizileri test kullanıcısıyla ayrı ayrı ikili hizalanarak her bir kullanıcı ile test kullanıcı arasındaki hizalamanın benzerlik puanları hesaplanır. Ardından bu diziler sahip oldukları puanlara göre en yüksekten en düşüğe doğru sıralanırlar. En yüksek puanlı diziden başlayarak belirlenen maksimum kullanıcı sayısı (ks) parametre değeri kadar kullanıcı dizisi seçilir. Diğer taraftan test kullanıcı dizisi, eğitim yüzdesi (ey) parametresine göre iki parçaya bölünür. Bu parametre değerinin yüzdelik kısmını oluşturan dizinin ilk parçası seçilen kullanıcı dizilerinin listesine eklenir. Test kullanıcı dizisinin geriye kalan ve aynı zamanda test kullanıcısının en son tercihlerini içeren diğer ikinci parçası ise test sonuçlarının doğruluğunu karşılaştırmak için kullanılır.

3- Çoklu Hizalama: Önceki aşamada seçilen kullanıcı dizileri listesindeki tüm diziler sırasıyla eşleşme, yanlış eşleşme ve boşluk durumları için belirlenen (s1, s2, s3) parametre değerleriyle Merkez Yıldız Algoritması kullanılarak bu aşamada çoklu hizalanır. Bu hizalama sonrasında birbirine en uyumlu pozisyonlara sahip eşit uzunlukta diziler elde edilmiş olur. Böylece diziler modellemeye hazır hale gelmiş olurlar.

4- PHMM ile Modelleme: Bu aşamada hizalanmış kullanıcı dizileri PHMM yapısı kullanılarak modellenir. Bu amaçla hizalı diziler sırayla modele tanıtılarak modelin durumlarındaki olasılık değerleri hesaplanır. Bu işlemle birlikte hizalı dizilerin her pozisyonda yer alan gözlem elemanlarının istatistiksel değerleri elde edilerek modelin durum yapılarında saklanır. Bu sayede mevcut hizalı dizi kümesinin tüm özelliklerini taşıyan genel bir dizi modeli oluşturulmuş olur.

5- Sonraki Kullanıcı Tercihi Tahmini: Yöntemin son ayağını oluşturan bu aşamada daha önce oluşturulmuş olan model üzerinden Viterbi algoritması uygulanarak kullanıcının bir sonraki kategori tercihi tahmin edilmeye çalışılır. Bunun için

algoritmanın Tercih Sayısı (ts) parametre değeri kadar test kullanıcısının en son yaptığı tercihler dikkate alınır. Bu tercih sırasını karşılayan modeldeki muhtemel çözüm yolları üzerinden bir sonraki tercih için gerçekleşmesi en yüksek olasılıklı gözlem değeri yani rezervasyon kategorisi bulunur. Bu işlem test kullanıcı dizisinin ey parametresi ile belirlenen ve test için ayrılan parçasındaki her gözlem değeri için sırayla tekrarlanır. İşlem sonunda test kullanıcı dizisi için doğru tahmin edilen gözlem değerlerinin yüzdelik değerleri çıktı olarak verilir.

5. DENEYSEL SONUÇLAR

Bu bölümde yöntemin başarısını ölçmek için yapılan deneysel testler hakkında bilgi verilerek test sonuçları değerlendirilmektedir. Test sonuçlarının başarı ölçümünde doğruluk (accuracy) metriği kullanılmıştır. Bunun nedeni tezde kullanılan verinin olumlu (positive) ve olumsuz (negative) şeklinde iki sınıfa ayrılmamasıdır. Çünkü kullanıcının bir sonraki rezervasyon kategorisini tahmin edilirken kullanıcının gerçekte yapmış olduğu seçimi referans alındığı için kullanıcının yanlış karar verme durumu söz konusu değildir. Bu yüzden verinin tamamı yalnızca olumlu (positive) veri olarak değerlendirilir. Tablo 5.1'de gri renkteki alanla gösterildiği gibi bu durumda yöntemin yapacağı tahminin kullanıcının gerçekte yaptığı seçimle aynı olması (true) veya farklı olması (false) biçiminde yalnızca iki ihtimal bulunmaktadır.

Tablo 5.1. Test verisinin karışıklık matrisi üzerinde karşılık gelen alanlar

Gerçekleşen / Tahminlenen	Positive (P)	Negative (N)
Positive	TP	FP
Negative (N)	FN	TN

Dolayısıyla doğruluk metriği bu noktada Eşitlik (5.1) ve Eşitlik (5.2)'de gösterildiği gibi kesinlik (precision) metriğiyle aynı sonucu vermektedir.

$$\text{Doğruluk (Accuracy)} = \frac{TP+TN}{TP+FP+TN+FN} = \frac{TP+0}{TP+FP+0+0} = \frac{TP}{TP+FP} \quad (5.1)$$

$$\text{Kesinlik (Precision)} = \frac{TP}{TP+FP} \quad (5.2)$$

Herhangi bir veri kümesinde belirli parametre değerleriyle yapılan bir test işlemi sırasında o veri kümesi içerisinde yer alan tüm kullanıcılar için tahmin sonuçları elde edilir. Her kullanıcı dizisi 100 adet kategori verisinden oluşması ve kullanıcı dizisinin yüzde 30'luk bölümünün test için ayrılması nedeniyle bir kullanıcı için tek

seferde 30 gözlem değeri tahmin edilmeye çalışılır. Dolayısıyla tek bir test işleminde toplam (30 x kullanıcı sayısı) kadar tahmin yapılır. Bir testin doğruluk değeri bulunurken önce her kullanıcı için doğru tahmin edilen gözlem değerlerinin sayısı kullanıcıdaki test gözlemlerinin sayısına (30) bölünür ve yüzdelik değeri hesaplanır. Sonrasında veri kümesindeki tüm kullanıcıların doğruluk yüzdelerinin ortalaması alınarak veri kümesinin belirli parametre değerleri için genel doğruluk yüzdesi elde edilir.

Bu bölümün ilk kısmında yapılan deneysel testler için seçilen parametre değerleri açıklanmakta sonrasında ise test sonuçları incelenerek yöntemin başarısı ölçülmektedir.

5.1. Deneysel Testler için Parametre Seçimi

Yapılan testlerde ikili hizalamalar için kullanılan Skor parametrelerindeki eşleme durumu için 1,0 yanlış eşleme için -1,0 ve boşluk durumu için ise -0,5 değerleri kullanıldı. Test kullanıcı dizisinin eğitim için ayrılan yüzdelik oran (ey) değeri ise yüzde 70 belirlenerek tüm testlerde sabit tutuldu. Tüm veri kümelerinde tercih sayısı (ts) parametresinin 5, 10, 20, 40 değerleri için ve yine kullanıcı sayısı (ks) parametresinin de 5, 10, 20, 40 değerleri için test yapılması düşünüldü. Ancak bu durumda sadece bir veri kümesi için 4 x 4 = 16 test yapılması gerekmektedir. Her veri kümelerinde aynı sayıda testler gerçekleştirilmesinin fazla zaman alması nedeniyle ilk başta tek bir veri kümesinde bu testlerin yapılması kararlaştırıldı. Sonrasında elde edilen sonuçlara göre parametrelerin sonuç dağılımına etkisini ölçmek için istatistiksel bir yöntem olan T testi uygulanmıştır.

T testi normal olarak dağılan iki sütunlu bir veride sütunlar arasında istatistiksel yönden önemli bir farkın olup olmadığını bize verir. Bu testin önem değeri (p) 0,005 değerinden küçük olduğu durumlar bize iki sütun arasında dikkate değer bir fark olduğunu gösterir. Bu değerden büyük olması ise bu sütunlara karşılık gelen parametre değerlerinin istatistiksel sonuçlar bakımından birbirine çok yakın olduğunu ifade eder. T testinin matematiksel formülü Eşitlik (5.3)'de gösterilmektedir:

$$t = \frac{\sum_{i=1}^{N} D_i}{\sqrt{\frac{(N(\sum_{i=1}^{N} D_i^2)-(\sum_{i=1}^{N} D_i)^2)}{N-1}}} \quad (5.3)$$

Burada D ifadesi iki ayrı veri grubunun (sütunun) aynı kayıt satırlarında bulunan değerlerin farkını, N ise gruplar için satır sayısını temsil eder.

En uygun parametrelerin seçilmesi için yapılacak ön test işleminde w_20-24 veri kümesi kullanıldı. Bu veri kümesi için yöntemin ts ve ks parametrelerinin her ikisinde 5, 10, 20 ve 40 değerleri için toplamda 16 test yapıldı ve sonuçlar üzerinde T testi uygulandı. Elde edilen sonuçlar Tablo 5.2 ve Tablo 5.3'de verilmiştir.

Tablo 5.2. w_20-24 veri kümesinde yapılan test sonuçlarının farklı kullanıcı sayısı (ks) ikililerine göre T testi analiz değerleri

ks / ts	5-10	5-20	5-40	10-20	10-40	20-40
5	1,93E-08	1,34E-09	2,11E-07	0,855024	0,565884	0,401346
10	5,6E-08	1,89E-09	3,01E-07	0,555248	0,76886	0,353206
20	1,36E-06	3,31E-07	1,74E-06	0,532317	0,722523	0,811496
40	2,71E-07	3,2E-10	1,14E-07	0,164030	0,629826	0,382858

Tablo 5.3. w_20-24 veri kümesinde yapılan test sonuçlarının farklı tercih sayısı (ts) ikililerine göre T testi analiz değerleri

ts / ks	5-10	5-20	5-40	10-20	10-40	20-40
5	0,640399	0,244628	0,551621	0,240459	0,640066	0,445832
10	0,218906	0,131727	0,092286	0,295442	0,192449	0,507024
20	0,911485	0,408556	0,857187	0,315881	0,803202	0,459272
40	0,892102	0,840443	0,898966	0,752333	0,94397	0,648035

Tablo 5.2'de w_20-24 verisi için tercih sayısı parametre değerinin sabit tutularak değişen kullanıcı sayısı parametre değerleriyle yapılan testlerin sonuçlarına ikili gruplar halinde T testi uygulanmış çıkan sonuçlar tabloda gösterilmiştir. Bu sonuçlara göre 5 kullanıcılı ile diğer kullanıcı sayısına sahip test sonuçlarının ikili karşılaştırılmasından (5-10, 5-20, 5-40) oluşan değerlerin 0,005 den küçük olduğu görülmekte buna karşın diğer kullanıcı sayısı ikilileri (10-20, 10-40, 20-40) için elde edilen değerler 0,005'den büyük çıkmıştır. Bu durum 5 kullanıcı sayısıyla yapılan testlerle diğer kullanıcı sayısına sahip test sonuçları arasında istatistiksel yönden bir fark oluştuğunu ve ayrı parametre değerleri olarak alınmasının anlamlı olduğunu bize gösterir. Diğer taraftan 10, 20 ve 40 kullanıcı sayısı değerli test sonuçlarının kendi

içindeki karşılaştırma sonuçlarından da istatistiksel bakımdan aralarında önemli bir fark oluşmadığı için bu değerlerden yalnızca birinin karşılaştırma grubuna dahil edilmesinin yeterli olduğu anlaşılır.

Tablo 5.3'deki değerler incelendiğinde 0,005'den küçük bir değer gözlemlenemediği için farklı tercih sayısı parametre değerleriyle yapılan test sonuçları arasında istatistiksel bir fark oluşmadığı ve sonuçların birbirine yakın dağılımlar gösterdiği görülmektedir. Elde edilen bu değerler sonucunda yapılacak deneysel testlerde kullanıcı sayısı (ks) parametresi için 5 ve 20 değerleri seçilmiştir. Tercih sayısı parametresi için de yine aynı değerler (5 ve 20) kullanılmıştır.

Parametre değerlerinin seçilmesinde ikinci adım hizalama işlemleri için kullanılan benzerlik puan parametre değerlerinin belirlenmesidir. Bu amaçla T testi sonrası yukarıda belirlenen 5 ve 20 kullanıcı sayısı değerlerinden 20 kullanıcı sabit tutularak w_20-24 veri kümesi üzerinde eşleşme, yanlış eşleşme ve boşluk puan parametrelerinin her biri için -0,5, -1,0 ve -2,0 değerleriyle testler yapılmıştır. Test sonuçlarının ortalama başarı yüzdeleri Tablo 5.4'te gösterilmektedir.

Tablo 5.4. w_20-24 veri kümesinde 20 kullanıcı sayısıyla ve farklı skor parametre değerleriyle yapılan testlerin başarı yüzdeleri

Puan (eşleşme (s1), yanlış eşleşme (s2), boşluk(s3))			**Ortalama başarı yüzdesi (%)**
s1	**s2**	**s3**	
1	-0,5	-0,5	31,72
1	-0,5	-1	33,50
1	-0,5	-2	35,25
1	-1	-0,5	31,60
1	-1	-1	31,63
1	-1	-2	33,57
1	-2	-0,5	33,98
1	-2	-1	31,43
1	-2	-2	31,90

Tablo 5.4'deki sonuçlara göre yapılan testlerde en yüksek başarı yüzde değeri eşleşme, yanlış eşleşme ve boşluk puan parametrelerinin sırasıyla 1, -0,5, -2 değerleri için elde edilmiştir. Bu ön test sonuçlarına göre yapılacak sonraki testlerde bu puan parametre değerleri seçilerek sabit tutulmuştur. Ayrıca kullanıcı sayısı için seçilen

parametre değerlerinin birbirleriyle 4 farklı kombinasyonu 5 ayrı veri kümesi için testlerde sırayla uygulanmıştır. Böylece ilk testler sonrasında toplamda 5 x 4 = 20 yeni test yapılarak sonuçlar tablolaştırılmıştır. Sonraki bölümde test sonuçlarının doğruluk değerleri verilerek sonuçlar incelenmektedir.

5.2. Sonuçlar

Yöntemin başarısını ölçmek için hem gerekli test sayısını en aza indirebilmek hem de test sonuçlarında parametrik değişimlerin gözlenebileceği en anlamlı parametre değerlerini seçilebilmesi amacıyla ilk olarak w_20-24 veri kümesi üzerinde ön testler yapılmıştır. Ön testler kullanıcı sayısı ve tercih sayısı parametrelerinin her ikisi için 5, 10, 20 ve 40 değerleriyle gerçekleştirilmiş ve elde edilen sonuçlar Tablo 5.5'de gösterilmiştir.

Tablo 5.5. w_20-24 veri kümesi üzerinde farklı parametre değerleri için yapılan ön test sonuçlarının en küçük, en büyük ve ortalama başarı yüzdeleri

ts \ ks	5			10			20			40		
	ort	**eb**	**ek**	**ort**	**eb**	**ek**	**ort**	**eb**	**ek**	**ort**	**eb**	**ek**
5	28,48	100,00	0,00	31,52	100,00	0,00	31,60	100,00	3,33	31,25	100,00	3,33
10	28,55	100,00	0,00	31,36	100,00	0,00	31,62	100,00	3,33	31,23	100,00	3,33
20	28,87	100,00	0,00	31,14	100,00	0,00	31,40	100,00	3,33	31,30	100,00	0,00
40	28,70	100,00	0,00	31,08	100,00	3,33	31,55	100,00	0,00	31,21	100,00	0,00

Tablo 5.5'de aynı kullanıcı sayısı değerine sahip test sonuçları karşılaştırıldığında farklı tercih sayısına göre elde edilen sonuçlarda değişmeler görülmesiyle birlikte ortalama değerlerinin birbirine çok yakın oldukları fark edilir. Diğer taraftan aynı tercih sayısı parametre değerleri için sonuçlar karşılaştırıldığında 20 kullanıcı sayısına sahip testlerin en yüksek değere ve 5 kullanıcı sayısıyla yapılan test sonuçlarının ise en düşük değere sahip olduğu görülmektedir. Kullanıcı sayısının arttıkça önce ortalama başarı yüzdesinin artması kullanıcı sayısının belli bir değerinden sonra başarı yüzdesinin azalarak bir çan eğrisi oluşturması sonuçların normal olarak dağıldığını bize göstermektedir. Kullanıcı sayısının optimal değerden az olması tahmin edilen test dizisinin profil yapısını modellemede modelin zayıf kalmasına yol açar. Her ne kadar yöntemde kullanıcı dizilerinden test kullanıcısına en benzer olanlar modele daha önce tanıtılsa bile bu durum modelin bu az sayıdaki kullanıcıya göre genel davranışı yansıtmayan çok daha özel kararlar almasına neden olmaktadır. Benzer şekilde kullanıcı sayısının optimal değerden fazla olması bu sefer

de modelin test veri profil özelliklerinden uzaklaşarak çok fazla genel sonuçlar üretmesine sebep olacaktır.

Kullanıcı sayısı için 5 ve 10 değerleri diğer veri kümeleri için yapılacak sonraki testlerde parametre olarak alınmış ve sonuçların başarı yüzdeleri Tablo 5.6'da gösterilmiştir.

Tablo 5.6. 6 farklı veri kümesi üzerinde farklı parametre değerleri için yapılan test sonuçlarının en küçük, en büyük ve ortalama başarı yüzdeleri

Veri kümesi	**ks** / **ts**	**5**			**20**		
		ort	**eb**	**ek**	**ort**	**eb**	**ek**
w_00-08	5	33,26	93,33	0,00	34,43	90,00	0,00
	20	33,13	86,67	0,00	35,03	86,67	3,33
w_08-12	5	30,44	100,00	0,00	32,53	100,00	0,00
	20	31,13	100,00	0,00	32,60	100,00	0,00
w_12-14	5	37,19	100,00	0,00	37,65	100,00	3,33
	20	37,27	93,33	6,67	37,02	100,00	3,33
w_14-18	5	34,84	93,33	0,00	36,74	96,67	0,00
	20	35,38	96,67	0,00	37,70	96,67	0,00
w_18-20	5	30,55	90,00	0,00	33,64	96,67	3,33
	20	31,31	86,67	0,00	33,89	96,67	6,67
w_20-24	5	28,48	100,00	0,00	31,60	100,00	3,33
	20	28,87	100,00	0,00	31,40	100,00	3,33

Tablo 5.6'daki sonuçlara göre en yüksek doğruluk oranlarına sahip sonuçların w_12-14 veri kümesi için elde edilmiş en düşük doğruluk oranlarına sahip sonuçların ise w_20-24 veri kümesi üzerinde yapılan testler için gözlenmiştir. Ayrıca veri kümeleri için en büyük doğruluk oranlarına bakıldığında en düşük değerin w_18-20 ve w_00-08 veri kümeleri için 86,67 olduğu görülür. Yine en küçük doğruluk değerleri içerisinde en yüksek değerin w_18-20 ve w_12-14 veri kümeleri için 6,67 değerinin bulunduğu gözükmektedir. Buradan testlerden elde edilen doğruluk değerlerinin genel anlamda normal bir dağılım gösterdiği ve veri kümeleri arasında alt ve üst sınırlar bakımından büyük farkların oluşmadığı anlaşılır. Diğer taraftan kullanıcı sayısı parametrelerine göre sonuçlar karşılaştırıldığında tüm veri kümelerinde 20 kullanıcıyla yapılan testlerin 5 kullanıcıyla yapılanlara göre daha iyi sonuç verdiği açıkça görülmektedir. Bununla birlikte tercih sayılarına göre sonuçlar kıyaslandığında w_00-08 veri kümesinin 5 kullanıcılı testlerinde ve w_12-14 ile

w_20-24 veri kümelerinin ise 20 kullanıcılı testlerdeki ortalama başarı değerleri 20 tercih sayısı değeri için daha düşük çıkmıştır. Ancak geri kalan tüm testlerde 20 tercih sayısı parametreli test sonuçlarının daha iyi olduğu görülür. Buna göre tercih sayısı parametresinin başarı yüzdesine etkisinin oldukça düşük seviyede olduğu buna karşın tercih sayısı artışının sonucun başarısına genel olarak olumlu bir etkide bulunduğu fark edilir.

Kullanılan yöntem uygulanan farklı kullanıcı sayısına sahip tüm veri kümelerinde ortalama başarı yüzdesi için birbirine çok yakın değerlere sahiptir. Ayrıca kullanıcıların kategori tercih dizilerinde art arda aynı kategori gözlem değerlerinin çokça sıralandığı ve belli kategori sayısının diğerlerine göre daha fazla olduğu durumlar için genel olarak tahmin başarı yüzdesinin daha yüksek olduğu gözlenmiştir. Bunun yöntemde kullanılan PHMM'nin olasılık hesabına dayanan ve hafızasız yapısıyla örtüşen bir durum olduğu söylenebilir.

Tez yönteminin kullanılan öneri sistemi verisindeki kullanıcı davranışını tahmin etmede ortalama başarı yüzdelerinin %30'lar seviyesinde olduğu tespit edilmiştir. Buna göre kullanılan yöntemin Weeplaces veri kümesinde kullanıcı tahmini için olumlu sonuç verdiği ve konum tabanlı öneri sistemlerinde kullanıcı tercih tahmininde pozitif katkı sağlayabileceği görülmüştür.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışması konum tabanlı öneri sistemleri için kullanıcıların bir sonraki davranışlarını rezervasyon kategori verileri üzerinden tahmin etmeyi hedeflemektedir. Tezde konum tabanlı öneri sistemindeki kullanıcı tercihlerini tahmin etmek amacıyla SMM'nin özelleşmiş bir uzantısı olan Profile Hidden Markov Model ve biyoinformatik algoritmaların birlikte kullanıldığı bir yöntem sunulmuştur.

Kullanılan yöntemin üç ana aşaması bulunmaktadır. Bunlar tahmin edilecek kullanıcı tercih verisi için en benzer profildeki başka kullanıcıların seçilmesi, seçilen kullanıcı verilerinin PHMM ile modellenmesi ve son olarak Viterbi algoritması kullanılarak kullanıcı kategori tercihinin tahmin edilmesidir.

Test kullanıcısına benzer profildeki kullanıcıların seçilmesi için biyoinformatik hizalama algoritmalarından yararlanılmış ve modelleme öncesinde kullanıcı tercih dizilerinin oluşturulmasında çoklu hizalama algoritmaları kullanılmıştır.

Yöntemin uygulanması ve test edilmesi işlemleri Weeplaces verisinin bir günün farklı saat dilimlerine göre düzenlenmiş 6 ayrı veri kümesi üzerinde gerçeklenmiştir. Genel testler öncesinde modellenecek kullanıcı sayısı ve test kullanıcısının en son tercih sayısı parametrelerine göre ön testler yapılmış olup elde edilen sonuçlar üzerinde en uygun parametre değerlerinin belirlenmesi için istatistiksel analizler yapılmıştır. Daha sonra seçilen parametre değerleriyle farklı periyotlarda oluşturulmuş veri kümeleri üzerinde testler gerçekleştirilmiştir.

Elde edilen test sonuçlarına göre kullanıcı sayısı artışının belli bir optimal değere kadar tahmin başarısını artırdığı optimal değerden daha fazla kullanıcı sayıları için başarı yüzdesinin tekrar azaldığı gözlenmiştir. Böylece kullanıcı sayısının belli bir optimal değer için en iyi sonucu verdiği gösterilmiştir. Tercih sayısı parametre değişiminin yapılan tahmin başarısı yüzdelerine oldukça düşük bir etkisinin olduğu ve genellikle tercih sayısı arttıkça tahmin başarı ortalama yüzdesinde çok az bir artış

olduğu görülmüştür. Ayrıca oluşturulan veri kümeleri üzerindeki test sonuçlarının ortalama başarı yüzdeleri arasında dikkate değer bir fark izlenmemiştir.

Sonuç olarak konum tabanlı öneri sistem verilerinde kullanıcı tercih tahmini için yöntem başarısının olumlu sonuçlar verdiği ve farklı çözümlerle birlikte kullanılmasının bu sistemlerdeki verime katkı sağlayacağı öngörülmektedir.

KAYNAKLAR

[1] Bao J., Zheng Y., Location-Based Recommendation Systems, Editors: Shekhar S., Xiong H., Zhou X, *Encyclopedia of GIS*, 2nd Ed., Springer, Switzerland,1145–1153, 2017.

[2] Doğaner A., Enformasyon Sistemlerinde Saklı Markov Modeli ve Bayes Tabanlı Sınıflandırıcılar ile Bilgi Modellerinin Geliştirilmesi, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ, 2015, 386141.

[3] Kaya Gülağız F., İçerik Dağıtım Ağlarında Senkronizasyon Zamanının Profile Hidden Markov Model ile Kestirimi, Doktora Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2018, 518829.

[4] Bobadilla J., Ortega F., Hernando A., Gutiarrez A., Recommender Systems Survey, *Knowledge-Based Systems*, 2013, **46**, 109-132.

[5] Carrer-Neto W., Hernandez-Alcaraz M. L., Valencia-Garcia R., Garcia-Sanchez F., Social Knowledge-Based Recommender System, Application to the Movies Domain, *Expert Systems with Applications*, 2012, **39**(12), 10990–11000.

[6] Winoto P., Tang T. Y., The Role of User Mood in Movie Recommendations, *Expert Systems with Applications*, 2010, **37**(8), 6086–6092.

[7] Szomszor M., Cattuto C., Alani H., O'Hara K., Baldassarri A., Loreto V., Servedio V. D. P., Folksonomies, the Semantic Web, and Movie Recommendation, *4th European Semantic Web Conference*, Innsbruck, Austria, 07 June 2007.

[8] Lee S. K., Cho Y. H., Kim S. H., Collaborative Filtering with Ordinal Scale-Based Implicit Ratings for Mobile Music Recommendations, *Information Sciences*, 2010, **180**(11), 2142–2155.

[9] Nanolopoulus A., Rafailidis D., Symeonidis P., Manolopoulus Y., Music Box: Personalizad Music Recommendation Based on Cubic Analysis of Social Tags, *IEEE Transactions on Audio, Speech and Language Processing*, 2010, **18**(2), 407–412.

[10] Tan S., Bu J., Chen C. H., He X., Using Rich Social Media Information for Music Recommendation via Hypergraph Model, Editors: Hoi S., Luo J., Boll S., Xu D., Jin R., King I., *Social Media Modeling and Computing*, 1st Ed., Springer, London, 213-237, 2011.

[11] Chen C., Meng X., Xu Z., Lukasiewicz T., Location-Aware Personalized News Recommendation with Deep Semantic Analysis, *IEEE* Access, 2017, **5**, 1624–1638.

[12] Li L., Zheng L., Yang F., Li T., Modeling and Broadening Temporal User Interest in Personalized News Recommendation. *Expert Systems with Applications*, 2014, **41**(7), 3168-3177.

[13] Yu Z., Zhou X., Hao Y., Gu J., TV Program Recommendation for Multiple Viewers Based on User Profile Merging, *User Modeling and User-Adapted Interaction*, 2006, **16**(1), 63–82.

[14] Barragans-Martinez A. B., Costa-Montenegro E., Burguillo J. C., Rey-Lopez M., Mikic-Fonte F. A., Peleteiro A., A Hybrid Content-Based and Item-Based Collaborative Filtering Approach to Recommend TV Programs Enhanced with Singular Value Decomposition, *Information Sciences*, 2010, **180**(22), 4290–4311.

[15] Gonzalez-Crespo R., Sanjuan-Martinez O., Manuel-Cueva J., Cristina-Pelayo B., Labra-Gayo J. E., Ordonez P., Recommendation System Based on User Interaction Data Applied to Intelligent Electronic Books, *Computers in Human Behavior*, 2011, **27**(4), 1445–1449.

[16] Nunez-Valdez E. R., Cueva-Lovelle J. M., Sanjuan-Martinez O., Garcia-Diaz V., Ordonez P., Montenegro-Marin C. E., Implicit Feedback Techniques on Recommender Systems Applied to Electronic Books, *Computers in Human Behavior*, 2012, **28**(4), 1186–1193.

[17] Bobadilla J., Serradilla F., Hernando A., Collaborative Filtering Adapted to Recommender Systems of E-learning, *Knowledge Based Systems*, 2009, **22**, 261–265.

[18] Zaiane O., Building a Recommender Agent for E-learning Systems, *Proceedings of the International Conference on Computers Education (ICCE'02)*, Auckland, New Zealand, 03-06 December 2002.

[19] Liu D., Shih Y., Integrating AHP and Data Mining for Product Recommendation Based on Customer Lifetime Value, *Information & Management*, 2004, **42**(3), 387-400.

[20] Park Y., Chang K., Individual and Group Behavior-Based Customer Profile Model for Personalized Product Recommendation, *Expert Systems with Applications*, 2009, **36**(2), 1932-1939.

[21] Ye M., Yin P., Lee W., Location Recommendation for Location-Based Social Networks, *The 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA-United States, 02-05 November 2010.

[22] Gao H., Tang J., Hu X., Liu H., Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks, T*he 7th ACM Conference on Recommender Systems*, Hong Kong, China, 12-16 October 2013.

[23] Wang H., Terrovitis M., Mamoulis N., Location Recommendation in Location-Based Social Networks Using User Check-in Data, *The 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Orlando, Florida, 05-08 November 2013.

[24] Ekstrand M., Riedl J., Konstan J., Collaborative Filtering Recommender Systems, *Foundations and Trends in Human–Computer Interaction,* 2011, **4**(2), 81–173.

[25] Ricci F., Rokach L., Shapira B., *Recommender Systems Handbook*, DOI: 10.1007/978-0-387-85820-3_1.

[26] Burke R., Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction*, 2002, **12**(4), 331–370.

[27] Adomavicius G., Tuzhilin A., Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on Knowledge and Data Engineering*, 2005, **17**(6), 734-749.

[28] Schafer J. B., Frankowski D., Herlocker J., Sen S., Collaborative Filtering Recommender Systems. Editors: Brusilovsky P., Kobsa A., Nejdl W., *The Adaptive Web Lecture Notes in Computer Science*, 1st Ed., Springer, Berlin, 2007.

[29] Su X., Khoshgoftaar T., A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, DOI:10.1155/2009/421425.

[30] Ekstrand M. D., Riedl J. T., Konstan J. A., Collaborative Filtering Recommender Systems, *Foundations and Trends in Human-Computer Interaction*, DOI: 10.1561/1100000009.

[31] Gong S. J., Ye H. W., Tan H. S., Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System, *2009 Pacific-Asia Conference on Circuits,Communications and System*, Chengdu, China, 16-17 May 2009.

[32] Breese J. S., Heckerman D., Kadie C., Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *14th Conferance Uncertainty in Artificial Intelligence*, Madison, Wisconsin-USA, 24-26 July 1998.

[33] Saraee M., Khan S., Yamaner S., Data Mining Approach to Implement a Recommendation System for Electronic Tour Guides, *The 2005 International Conference on E-Business, Enterprise Information Systems, E-Government, and Outsourcing(EEE 2005)*, Las Vegas, Nevada, 20-23 June 2005.

[34] Koren Y., Bell R., Volinsky C., Matrix Factorization Techniques for Recommendation Systems, *Computer*, IEEE Computer Society, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.147.8295&rep=rep1&type=pdf (Access Date: 05 June 2012).

[35] Zukerman I., Albrecht D., Nicholson A., Predicting Users' Requests on the WWW, Editor: Kay J., *UM99 User Modeling-Proceedings of the Seventh International Conference*, 1st Ed., Springer,Verlag, 275–284, 1999.

[36] Yang D., Zhang Y., Yu Z., Wang Z., A Sentiment Enhanced Personalized Location Recommendation System, *HT '13: 24th ACM Conference on Hypertext and Social Media*, Paris, France, 01-03 May, 2013.

[37] Zheng Y., Zhang L., Ma Z., Xie X., Ma W. Y., Recommending Friends and Locations Based on Individual Location History, *ACM Transactions on the Web*, 2011, **5**(1), 1-44.

[38] Park M. H., Hong J. H., Cho S. B., Location-Based Recommendation System Using Bayesian Users Preference Model in Mobile Devices, Editors: Indulska J., Ma J., Yang L.T., Ungerer T., Cao J., *Ubiquitous Intelligence and Computing*, 1st Ed., Springer, Berlin, 1130–1139, 2007.

[39] Horozov T., Narasimhan N., Vasudevan V., Using Location for Personalized POI Recommendations in Mobile Environments, *International Symposium on Applications and the Internet (SAINT'06)*, Phoenix, USA, 23-27 January 2006.

[40] Ye M., Yin P., Lee W. C., Lee D. L., Exploiting Geographical Influence for Collaborative Point of Interest Recommendation, *SIGIR '11: The 34th International ACM* SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24-28 July 2011.

[41] Takeuchi Y. Sugimoto M., CityVoyager: An Outdoor Recommendation System Based on User Location History, *Third International Conference-UIC 2006*, Wuhan, China, 03-06 September 2006.

[42] Zheng Y., Zhang L., Xie X., Ma W. Y., Mining Interesting Locations and Travel Sequences from GPS Trajectories, *WWW '09: The 18th International World Wide Web Conference*, Madrid, Spain, 20-24 April 2009.

[43] Yu X., Pan A., Tang L. A., Li Z., Han J., Geo-Friends Recommendation in GPS-Based Cyber-Physical Social Network, *The 2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Kaohsiung, Taiwan, 25-27 July 2011.

[44] Noulas A., Scellato S., Lathia N., Mascolo C., Mining User Mobility Features for Next Place Prediction in Location-Based Services, *12th International Conference on Data Mining*, Brussels, Belgium, 10-13 December 2012.

[45] Likhyani A., Padmanabhan D., Bedathur S., Mehta S., 2015. Inferring and Exploiting Categories for Next Location Prediction, *The 24th International Conference on World Wide Web*, Florence, Italy, 18-22 May 2015.

[46] Li W., Eickhoff C., de Vries A. P., Want a Coffee? Predicting Users' Trails, *The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Oregon, USA, 12-16 August 2012.

[47] Cheng H., Ye J., Zhu Z., What's Your Next Move: User Activity Prediction in Location-based Social Networks, *The 13th SIAM International Conference on Data Mining*, Austin, Texas-USA, 2-4 May 2013.

[48] Cao J., Xu S., Zhu X., Lv R., Liu B., Effective Fine-Grained Location Prediction Based on User Check-in Pattern in LBSNs, *Journal of Network and Computer Applications*, 2018, **108**, 64–75.

[49] Jeff G., Vincent A. T., Charette S. J., Derome N., A Brief History of Bioinformatics, *Briefings in Bioinformatics*, 2019, **20**(6), 1981–1996.

[50] Haque W., Aravind A., Reddy B., Pairwise Sequence Alignment Algorithms–A Survey, *ISTA '09: Information Science, Technology and Applications*, Kuwait, Kuwait, 20-22 March 2009.

[51] Taheri J., Zomaya A., A Novel Metaheuristic for Solving the Multiple Sequence Alignment Problem, *2008 International Conference on Bioinformatics & Computational Biology (BIOCOMP'08*, Las Vegas, NV-USA, 14-17 July 2008.

[52] Chan S. C., Wong A. K. C., Chiu D. K. Y., A Survey of Multiple Sequence Comparison Methods, *Bulletin of Mathematical Biology*, 1992, **54**(4), 563-598.

[53] Feng D. F., Doolittle R. F., Progressive Sequence Alignment as a Prerequisite to Correct Pylogenetic Trees, *Journal of Molecular Evolution*, 1987, **25**, 351–360.

[54] Alpaydın E., *Yapay Öğrenme*, 2. Baskı , Boğaziçi Üniversitesi Yayınevi, İstanbul, 2013.

[55] Can C. E., Saklı Markov Modelinin Farklı Dağılımlar İçin İncelenmesi, Doktora Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2016, 430922.

[56] Büyüktatlı F., Şirketlerdeki Erken Uyarı Göstergeleri İle Saklı Markov Modeli Üzerine Bir Uygulama, Yüksek Lisans Tezi, Akdeniz Üniversitesi, Sosyal Bilimler Enstitüsü, Antalya, 2013, 344447.

[57] Viterbi A., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, *IEEE Trans. Inf. Theory,*.1967, **13**(2), 260-269.

KİŞİSEL YAYIN VE ESERLER

Göz F., Mutlu A., Küçük K., Temur M., **Gün A.,** Türkçe Metinlerden Anahtar Kelime Çıkarma için Merkezilik Ölçütlerinin İncelenmesi, *29. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Çevrimiçi, 9-11 Haziran 2021.

ÖZGEÇMİŞ

Abdurrahman Gün, lise öğrenimini Ankara Fen Lisesi'nde tamamladı. 2007 yılında girdiği Anadolu Üniversitesi Bilgisayar Mühendisliği Bölümü'nden hazırlık eğitimi ile birlikte 2014 yılında mezun oldu. 2016 yılında Aksaray Üniversitesi Elektrik Elektronik ve Bilgisayar Bölümü'nde yüksek lisans eğitimine başladı. 2017'de Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nde başladığı araştırma görevlisi görevini halen sürdürmektedir.