INF224 Veri Yapısı ve Algoritmalar Dönem Sonu Projesi

Muhammed Ali Karslı

12 Aralık 2024

Özet

A project focused on constructing a graph from raw data and developing a recommendation system leveraging the graph structure. The system combines traditional algorithms with neural network-based methods to provide and evaluate recommendations.

1 Giriş

Projenin amacı C veya C++ dilinde iki taraflı, yönsüz bağlantılı graf yapısının kurulup ardından çeşitli öneri sistemlerinin tasarlanmasıdır. Bu öneri sistemleri 5'e bölünebilir:

1. Rassal izlenmemiş film önerme
2. İzlenmemiş en iyi filmleri önerme
3. Hedef kullanıcıya en çok benzeyen kullanıcının en beğendiği filmleri önerme
4. Hedef kullanıcıya en yakın ağırlıklı izlenmemiş filmleri önerme
5. Özgün bir metod geliştirme: Belirli bir filmi beğenmiş kullanıcı grubuna o grupta en çok sevilen filmleri önerme

Proje için GroupLens sitesindeki MovieLens 100 bin satırlık veri setinin kullanılması gerekmektedir.

Rassal yürüyüş yöntemiyle film önerme, başarı performans ölçümü, zaman ve mekan karmaşıklığı ölçümü analizi de bonus isterlerdir.

2 Yöntem

2.1 Veri Seti

Projenin veri setinde filmlerin türü, açıklaması; kullanıcıların meslekleri, yaşları, cinsiyetleri ve hangi kullanıcının hangi filme kaç puan verdiği yer almaktadır. Bunlar arasından proje isteri gereği sadece "u.data" isimli; kullanıcı numarası, filme verilen puan, filmin numarası ve zaman bilgisi bilgilerini barındıran veri seti kullanılacaktır. Bu bilgilerden zaman bilgisi gene projenin isterleri doğrultusunda harici tutulacaktır.

Dolayısıyla kullanılacak veri setinde kullanıcı numarası, film numarası ve kullanıcının o filme kaç puan verdiğini gösteren bilgiler vardır. Veri setinde toplam 100000 değerlendirme, 943 kullanıcı, 1682 film vardır. Her bir puan 1 ile 5 arasındadır ve 5 en yüksek, 1 ise en düşük puan olarak kabul edilmiştir.

Sonuç olarak veri setinde kullanıcı ve film tipinde iki düğüm ve bunlar arasındaki skorları temsil eden bağlantılar vardır. Proje isteri gereği her bir skor, temsil edildiği bağlantının ağırlığı olarak kabul edilmiştir. Bu yöntem daha iyi skorun daha yüksek bir ağırlık oluşturmasıyla sonuçlandığından ötürü öneri algoritmalarında yeri geldiğinde bu ağırlık tersine çevrilip veya bir sayıdan çıkartılıp işlem yapılmıştır.

2.2 Graf Yapısı

Graf yapısı; Graf, Düğüm, Bağlantı isimli üç yapı üzerinden inşa edilmiştir. Graf, kullanıcılar ve nesneler isimli iki Düğüm bağlı listesinden oluşmaktadır. Düğüm; numara isimli bir tam sayı değerden, Bağlantı bağlı listesinden ve "sonraki" isimli Düğüm işaretçisinden oluşmaktadır. Her bir kullanıcı ve film Düğüm tipinde soyutlanmaktadır. Bağlantı; kullanıcı numarası, film numarası, skor isimli 3 tam sayı değerden ve "sonraki bağlantı" isimli bir Bağlantı işaretçisinden oluşmaktadır.

Graf, iki kümeli bir graf olduğundan ötürü sadece kullanıcılar ve filmler kümesi oluşturulmuş, bağlantılar da sadece kullanıcılar ve filmler arasında olmak üzere yönsüz bir şekilde oluşturulmuştur.

Graf, veri setinden satır satır okunup bu yapılara dayanan temel fonksiyonlar ile inşa edilmiştir. Tüm graf yapısı ve öneri sistemi C dili kullanılarak yazılmıştır.

2.3 Öneri Sistemleri

Öneri sistemlerinin açıklanmasında ve yazı boyunca filmi izlemek ifadesi filme skor vermek anlamında kullanılmıştır. Hedef kullanıcı ifadesi ise önerinin yapılacağı kullanıcıyı kastetmektedir.

Kullanılan öneri sistemlerini beş temel ve iki ilave algoritma olmak üzere yediye ayırabiliriz:

2.3.1 Rastgele Film Önerme

Rastgele film önerme algoritmasında kullanıcının izlemediği tüm filmler oluşturulan geçici bir listede kaydedilir, ardından rastgele bir dizin seçilir ve seçilen film kullanıcıya önerilir.

Tüm filmler teker teker ziyaret edilmekte, her bir filmin tüm bağlantıları gezilip edilip kullanıcının daha önce o filme bir skor verilip verilmediği kontrol edilmektedir. Algoritma, n toplam film sayısı olmak üzere, en kötü durumda $O(n^2)$ zaman karmaşıklığına ve her durumda O(n) mekan karmaşıklığına sahip olacaktır.

2.3.2 Derecesi Yüksek Film Önerme

Derecesi yüksek film önerme algoritmasında kullanıcının izlemediği filmlerden skoru en yüksek olan filmler önerilir. Kullanıcıya önerilen film sayısı önceden belirlenmiş olup m kadardır.

Bunun için her bir film gezilip kullanıcının o filmi izleyip izlemediği kontrol edilir, izlemediyse o filme ait toplam skor hesaplanır. m uzunluğunda iki boyutlu geçici bir liste oluşturulur. Eğer toplam skor o listedeki filmlerin toplam skorlarının en az birinden fazlaysa listeye o alınır, en düşük toplam skorlu olan film listeden çıkartılır. Son olarak listedeki filmler kullanıcıya önerilir.

n toplam film sayısı, m önerilecek film sayısı olmak üzere; her bir film ve filme ait tüm bağlantılar gezileceği, ardından skora ve izlenmeye bağlı olarak $m * 2$ boyutlu liste gezileceği ve liste üzerinde eleman kaydırma işlemi yapılacağı için en kötü durumda zaman karmaşıklığı $O(n(n + m^2))$ olacaktır. Burada m, n'nin yanında önemsiz derecede küçük kalacağı için ve teori gereğince en kötü durumda zaman karmaşıklığının $O(n^2)$ olacağı söylenebilir. Mekan karmaşıklığı ise her durumda $O(2m)$ olacaktır.

2.3.3 Benzer Kullanıcı Esaslı Film Önerme

Benzer kullanıcı esaslı film önerme algoritmasında kullanıcıya en çok benzeyen kullanıcı bulunur. Ardından benzeyen kullanıcının en çok beğendiği filmler arasından öneri yapılacak kullanıcının izlemediği filmler kullanıcıya önerilir. Burada benzerlik birçok farklı şekilde yorumlanabilir. Bu projede benzerlik tespiti için bir benzerlik skoru tanımlanmıştır. Bu skor, aynı filme puan vermiş iki kullanıcının verdikleri puanların birbirine olan yakınlığıyla ölçülür. Dolayısıyla, bu algoritmada, aynı filmlere puan vermiş ve verdikleri puanlar birbirine en yakın olan iki kullanıcı birbirine benzer bulunur.

Bu algoritmada her bir kullanıcı dolaşılır, her bir kullanıcının her bir bağlantısı ziyaret edilir ve hedef kullanıcının her bir bağlantısıyla karşılaştırılır. Eğer aynı film puanlanmışsa (6 - (puanlar arasındaki fark) kadar) benzerlik skoru arttırılır. Son olarak en yüksek benzerlik skoruna sahip kullanıcı benzer kullanıcı ilan edilir. Ardından benzer kullanıcının izlediği filmler arasından hedef kullanıcının izlemediği ve benzer kullanıcının en yüksek puan verdiği filmler geçici bir iki boyutlu listede depolanır. Son olarak bu filmler kullanıcıya önerilir.

k toplam kullanıcı sayısı, m önerilecek film sayısı olmak üzere; en kötü durumda zaman karmaşıklığı $O(k^2)$, her durumda mekan karmaşıklığı $O(2m)$ değerindedir.

2.3.4 Ağırlıklı Uzaklığa Dayalı Film Önerme

Ağırlıklı uzaklığa dayalı film önerisi algoritmasında; hedef kullanıcı düğümünün diğer film düğümlerine olan uzaklığı Dijkstra'nın algoritması ile bulunmuş ve film sayısı uzunluğundaki bir geçici listede depolanmıştır. Ardından düğümü en yakın olan filmler, önerilecek film sayısı uzunluğunda geçici bir listede depolanmış ve kullanıcıya önerilmiştir.

Dijkstra'nın algoritmasının zaman karmaşıklığı $O(n^2)$'dir. Algoritmada, film düğümlerini ve düğüm bağlantılarını dolaşma gibi diğer işlemlerde de en fazla $O(n^2)$ karmaşıklığında çözümler kullanılmıştır. Dolayısıyla, n toplam film sayısı, m önerilecek film sayısı olmak üzere; en kötü durumda zaman karmaşıklığı $O(n^2)$, her durumda mekan karmaşıklığı $O(m)$ değerindedir.

2.3.5 Kullanıcı Gruplama Esaslı Film Önerme

Kullanıcı gruplama esaslı film önerme algoritmasında aynı filme beş puan vermiş bütün kullanıcılar bir grup olarak kabul edilir. Bu kullanıcıların en çok beş puan verdiği filmler bulunur ve toplam film sayısı boyutundaki geçici bir listede saklanır. Kullanıcıdan bir x değişkeni alınır. Listedeki ilk x film, kullanıcı grubundaki daha önce bu x filmleri izlememiş olan kullanıcılara önerilir.

Bu algoritmadaki fikir, aynı filme beş puan vermiş kişilerin, beş puan verdikleri başka bir filmi birbirlerine önerebileceğidir. Aynı filmleri beğendiğiniz bir arkadaşınızın size önereceği bir filmi muhtemelen beğenecek olmanız beklenmiş ve bu iddia temel alınmıştır.

x, ilk x film, y kullanıcı grubundaki kişi sayısı, z bir kullanıcının sahip olduğu bağlantı sayısı, n toplam film sayısı olmak üzere; zaman karmaşıklığı $O(xyz)$, mekan karmaşıklığı ise $O(n)$'dir.

2.3.6 Rassal Yürüyüşle Film Önerme

Rassal yürüyüşle film önerme algoritmasında başlangıç düğümünden itibaren önceden belirlenmiş w değişkeni kadar düğüm değiştirilir. Düğüm değiştirilirken bağlantı seçimi rastgele yapılır. İki kümeli grafta öneri sistemi bağlamında, eğer varılan son düğüm bir kullanıcı düğümü değilse bir kez daha rastgele ilerlenir. Varılan film kullanıcıya önerilir.

Bağlantılar arası geçiş bağlı listede ilerleme yöntemiyle yapıldığı için; l, film ve kullanıcı sayısının maksimumu olmak üzere, en kötü durumda zaman karmaşıklığı $O(l)$ şeklinde ifade edilebilir. Mekan karmaşıklığı $O(1)$'dir.

2.3.7 Nöral Ağla Film Önerme

Nöral ağla film önerme algoritmasında, önceden eğitilmiş modelle, hedef kullanıcının daha önce izlemediği tüm filmler için skor tahmini yapılır, önerilmesi istenilen film sayısı uzunluğunda geçici bir listede saklanır. Ardından en yüksekten en düşüğe olmak üzere istenilen sayıda film kullanıcıya önerilir.

l, film ve kullanıcı sayısının maksimumu, s veri setindeki her bir satır olmak üzere; model eğitim süreci dahil olmak üzere algoritmanın zaman karmaşıklığı $O(s)$, mekan karmaşıklığı $O(l)$'dir.

2.4 Nöral Ağ

Nöral ağı geliştirmek için matriks faktorizasyonlu nöral ağ eğitimi yapılmıştır. Konuyla ilgili detaylı bilgiye bu adresten ulaşılabilir.

2.5 Başarı Metriği

Yapılan önerilerin başarılarının ölçümü nöral ağla yapılan tahminler üzerinden hesaplanmıştır. Her bir öneri, aynı kullanıcı-film ikilisi için eğitilmiş modele tahmin ettirilmiş, kullanıcıya önerilen filmin tahmini skoruna göre hata oranı hesaplanmıştır. Buna göre tahmini 5 puanlı bir film önerilmişse bunun hata oranı 0, tahmini 1 puanlı bir film önerilmişse bunun hata oranı 1'dir. Hata oranı yapılan her önerinin sonunda kullanıcıya çıktı olarak verilmektedir.

$$\textbf{Hata Oranı:} \quad f : \mathbf{R} \to \mathbf{R}, \quad f(y) = \frac{5-y}{4}$$

y : Modelle tahmin edilen skor

3 Bulgular

Oluşturulan proje çeşitli girdilerle birçok kez sınanmış ve tutarlı sonuçlar elde edildiği gözlemlenmiştir. Başarı metriği incelendiğinde rastgele film önerme ve rassal yürüyüşle film önerme algoritmalarının, doğalarından anlaşılacağı üzerine, çoğu durumda diğer algoritmalardan daha yüksek hata oranına sahip olduğu gözlemlenmiştir. Derecesi yüksek film önerme algoritması genelde başarılı sonuçlar vermiştir. Benzer kullanıcı esaslı film önerme algoritması ortalama bir performans sergilemektedir. Ağırlıklı uzaklığa dayalı film önerme ise genelde ortalama altı bir başarı oranına sahiptir. Kullanıcı gruplama esaslı film önerme ise projedeki en başarılı algoritmalardan birisi olarak göze çarpmaktadır. Nöral ağla film önerme algoritmasına, başarı metriği kendisi olduğundan ötürü yorum yapılamamaktadır.

Projedeki en başarılı iki algoritmanın seçilmesi gerekirse derecesi yüksek film önerme algoritmasıyla kullanıcı gruplama esaslı film önerme algoritmasının seçilmesinin gerektiği yapılan gözlemler sonucu iddia edilebilir.

4 Tartışma

Öneri sistemlerinin genel performansını arttırmak için projede çeşitli iyileştirmeler ve güncellemeler yapılabilir. Süre açısından faydalı olabilecek bir geliştirme, Bellman-Ford algoritmasının Dijkstra'nın algoritmasına bir alternatif olarak kullanılıp karşılaştırmalı sonuçlar elde edilmesi olabilir. Bununla birlikte, kullanılan çeşitli hiper parametrelerin hassas ayarı için çeşitli yöntemler geliştilip nöral ağ modelinin ve öneri algoritmalarının performansı arttırılabilir. Bunun yanı sıra, benzer kullanıcıları bulma algoritmasında, kullanıcıların benzerliklerini bulmada yeni ölçütler belirlenebilir.

Projenin ilerleyen aşamalarında veri setindeki diğer bilgilerin de öneri sistemlerine dahil edilmesi düşünülebilir. İçerik, tür, yıl, açıklama gibi filme ait ve yaş, meslek gibi kullanıcıya ait çeşitli verilerle bütün öneri algoritmaları çok yönlü bir şekilde geliştirilebilir. Ayrıca bu verilerle inovatif ve daha kapsayıcı algoritmalar üretilebilir.

Yeni bir başarı metriği algoritması türetilebilir. Bu aynı zamanda nöral ağ modelinin kalitesinin testine imkan sağlar.

Son olarak, kullanılan matriks faktorizasyonlu nöral ağ modeli yerine graf nöral ağı ya da nöral işbirlikçi filtreleme modelleri kullanılabilir. Bunun sonucunda elde edilen modeller sadece yeni tahminler üretmekle kalmayıp birbirlerinin hata oranını tespit etmekte de çarpraz bir şekilde kullanılabilir.