

Her Komşu Aynı Dili Konuşmaz

Sheaf Neural Networks ile GNN'lerin Sınırlarını Aşmak

MEHMET ÜNLÜ | Kategori: Yapay Zekâ & Derin Öğrenme

Birbirine bağlı verilerle çalışırken ilk aklımıza gelen şeylerden biri Graph Neural Networks, yani **GNN'ler** oluyor. Sosyal ağlardan moleküllere, ulaşım sistemlerinden bilgi ağlarına kadar pek çok problemde GNN'ler, düğümler arasındaki ilişkileri kullanarak güçlü temsiller öğrenebiliyor.

Ama burada gözden kaçan küçük bir varsayım var:

Komşu düğümlerin bilgilerini doğrudan birbirleriyle karşılaştırabileceğimizi varsayıyoruz.

Peki ya bu doğru değilse?

Bir düşünün: Bir grafikteki her düğüm kendi yerel koordinat sistemine, kendi özellik uzayına veya kendi bilgi yorumlama biçimine sahip olsaydı ne olurdu? Bir düğümden diğerine bilgi gönderirken bu bilgiyi olduğu gibi taşımak yerine, önce dönüştürmek gerekseydi?

İşte tam bu noktada **Sheaf Neural Networks (SNN)** devreye giriyor.

Sheaf Neural Networks, grafik üzerindeki bilgiyi yalnızca “komşudan komşuya mesaj gönderme” problemi olarak görmek yerine, **farklı yerel bilgi uzayları arasındaki uyumu ve dönüşümü modellemeye** çalışıyor.

Başka bir ifadeyle, GNN'lerin sorduğu:

“Komşumdan ne öğrenebilirim?”

sorusuna ek olarak Sheaf Neural Networks şunu soruyor:

“Komşumun bilgisini kendi dünyamda nasıl yorumlamalıyım?”

GNN'ler Grafikleri Nasıl Anlar?

Önce temel noktaya geri dönelim.

Bir grafik düşünelim. Örneğin bir sosyal ağ:

- İnsanlar → **Node**
- Arkadaşlık ilişkileri → **Edge**
- İnsanların özellikleri → **Node Features**

Ali'nin Ayşe ile, Ayşe'nin de Mehmet ile bağlantısı olsun.

Bir GNN, Ali'nin yalnızca kendi özelliklerine bakmakla kalmaz. Komşularından gelen bilgileri de kullanır.

Temel fikir kabaca şöyledir:

Node → Komşularından mesaj alır → Mesajları birleştirir → Yeni temsil oluşturur

Matematiksel olarak bunu basitleştirerek şöyle ifade edebiliriz:

$$h_v^{(l+1)} = \text{UPDATE}(h_v^{(l)}, \text{AGGREGATE}(\{ h_u^{(l)} : u \in \mathcal{N}(v) \}))$$

Burada h_v bir node'un temsili, $\mathcal{N}(v)$ ise o node'un komşularını ifade ediyor.

Mantık oldukça sezgisel:

“Benim kim olduğumu anlamak için kimlerle bağlantılı olduğuma da bakmalıyım.”

Ve bu yaklaşım gerçekten güçlü. Fakat işin içinde küçük bir varsayım saklı.

Komşuların ürettiği bilgileri aynı temsil uzayında toplayabiliyor muyuz?

İşte Sheaf Neural Networks'ün hikâyesi tam burada başlıyor.

Bütün Node'lar Aynı Dili mi Konuşuyor?

Bir sınıfta üç farklı uzman olduğunu düşünelim: bir mühendis, bir doktor ve bir ekonomist aynı probleme bakıyor.

Üçünün de elinde değerli bilgiler var. Ancak kullandıkları kavramlar, ölçüm biçimleri ve olayları yorumlama şekilleri birbirinden farklı.

Şimdi bu üç kişinin bilgilerini olduğu gibi toplayıp ortalamasını aldığımızı düşünün. Sonuç büyük ihtimalle pek anlamlı olmayacaktır.

Çünkü problem bilginin eksik olması değil:

Bilginin farklı uzaylarda temsil edilmesi.

Grafiklerde de benzer bir durum ortaya çıkabilir. Klasik message-passing yaklaşımlarında node'lardan gelen embedding'leri toplamak oldukça doğal görünür:

“Komşuların bilgilerini al, aggregate et ve yeni node temsili oluştur.”

Fakat her node'un bilgisi aynı şekilde yorumlanmak zorunda değil. Bir node'dan diğerine bilgi aktarılırken:

- ölçek değişebilir,
- yön değişebilir,
- koordinat sistemi değişebilir,
- temsil uzayı değişebilir,

- ilişkinin kendisi bir dönüşüm gerektirebilir.

Dolayısıyla asıl soru artık sadece “**hangi bilgi geliyor?**” değildir.

“*Bu bilgi geldiğinde onu nasıl yorumlamalıyım?*”

Sheaf Nedir?

Burada matematiğin biraz daha derinlerine iniyoruz.

Sheaf, farklı yerel alanlarda bulunan bilgilerin birbiriyle nasıl ilişkilendirileceğini ve nasıl tutarlı hale getirileceğini modellemek için kullanılan matematiksel bir yapıdır.

Bu tanım ilk bakışta biraz soyut gelebilir. O yüzden bunu grafik üzerinden düşünelim.

Bir Sheaf yapısında temel olarak her node için bir **stalk** bulunur. Stalk’i, o node’un sahip olduğu **yerel bilgi uzayı** gibi düşünebiliriz. Yani:

Node → Stalk

Ama iş burada bitmiyor. Node’lar arasındaki edge’ler üzerinden bilgi aktarılırken, bu bilginin nasıl dönüştürüleceğini belirleyen **restriction map** adı verilen dönüşümler devreye giriyor.

Basitleştirelim:

$$\rho_{u \rightarrow e} : \mathcal{F}(u) \rightarrow \mathcal{F}(e)$$

ve

$$\rho_{v \rightarrow e} : \mathcal{F}(v) \rightarrow \mathcal{F}(e)$$

Burada $\mathcal{F}(u)$ ve $\mathcal{F}(v)$ node’ların yerel bilgi uzaylarını; $\mathcal{F}(e)$ ise edge üzerinde ortaklaştırılan bilgi uzayını temsil ediyor.

Bu noktada çok önemli bir fikir ortaya çıkıyor:

Bilgiyi doğrudan toplamak yerine, önce karşılaştırılabilir hale getiriyoruz.

GNN’den Sheaf Neural Network’e Geçiş

Şimdi iki yaklaşımı yan yana koyalım. Klasik GNN’in temel yaklaşımı:

Node → Message → Aggregation → Updated Node

Sheaf Neural Network yaklaşımında ise tablo biraz değişiyor:

Node → Yerel Bilgi → Edge Üzerinden Dönüşüm → Uyumlaştırma → Updated Node

Yani edge artık sadece “bu iki node arasında bağlantı var” anlamına gelmiyor. Aynı zamanda:

“Bu iki node’un bilgisi birbirine nasıl dönüştürülmeli?”

sorusuna da cevap verebiliyor.

Bu oldukça önemli bir fark. Çünkü grafik yapısının yalnızca **kim kime bağlı?** bilgisini değil, aynı zamanda **bu bağlantı üzerinden bilgi nasıl ilişkilendirilmeli?** bilgisini de taşımasına izin veriyoruz.

Restriction Map: Edge’ler Artık Sadece Bağlantı Değil

Sheaf yaklaşımının en önemli kavramlarından biri **restriction map**. Bunu bir “çeviri mekanizması” gibi düşünebiliriz.

Ali Türkçe konuşuyor, Ayşe İngilizce konuşuyor olsun. Ali’nin söylediğini Ayşe’nin dünyasına doğrudan aktarmak yerine önce bir çeviri yapıyoruz.

Grafikte de benzer bir mantık var. Bir node’un temsili x_u olsun. Bu bilgi edge üzerinden başka bir ortak uzaya taşınırken:

$$\rho_{u,e}(x_u)$$

haline geliyor. Diğer node için de:

$$\rho_{v,e}(x_v)$$

elde ediyoruz. Artık iki bilgi, edge’in tanımladığı ortak bağlam içerisinde karşılaştırılabilir.

Bu nedenle Sheaf Neural Networks’ü sezgisel olarak şöyle düşünebiliriz:

GNN komşular arasında bilgi taşır. Sheaf Neural Network ise bu bilginin taşınırken nasıl dönüştürüleceğini de modellemeye çalışır.

Peki Bunun GNN’lere Katkısı Ne?

Burada Sheaf Neural Networks’ün asıl değerini görebiliyoruz.

1. Yerel Temsilleri Modelleyebilmek

Klasik GNN’lerde node embedding’lerinin aynı genel temsil uzayında ele alınması oldukça yaygındır. Sheaf yaklaşımı ise her node’un kendi yerel bilgi uzayına sahip olmasına izin verir.

Bu, özellikle geometrik veya ilişkisel yapının karmaşık olduğu problemlerde önemli olabilir.

2. Edge’lere Daha Fazla Anlam Kazandırmak

Klasik bir grafikte edge çoğu zaman “u ile v arasında bağlantı var” der. Sheaf yaklaşımında edge aynı zamanda:

“u’daki bilgiyi v ile karşılaştırırken şu dönüşümü kullan.”

diyebilir. Bu, grafiğin geometrik yapısının daha zengin şekilde modellenmesini sağlayabilir.

3. Heterophily Problemlerine Farklı Bir Bakış

GNN’lerde önemli problemlerden biri **heterophily**. Basitçe ifade edersek: komşu node’lar her zaman birbirine benzemez.

Örneğin bir sosyal ağda iki kişinin arkadaş olması, onların aynı özelliklere sahip olduğu anlamına gelmez. Hatta bazı grafiklerde farklı özelliklere sahip node’ların birbirine bağlanması yapının önemli bir parçasıdır.

Klasik message passing yaklaşımı komşuların benzerliğine fazla yaslandığında problem yaşayabilir. Sheaf yaklaşımı ise daha esnek bir bakış sunar:

Komşuların aynı olması gerekmiyor. Önemli olan aralarındaki ilişkinin nasıl dönüştürüldüğünü modelleyebilmek.

Grafiklerde “Yerel Geometri” Neden Önemli?

Burada konu biraz daha ilginç hale geliyor.

Bir grafiği yalnızca düğümler ve bağlantılardan oluşan bir veri yapısı olarak düşünmek kolaydır. Ama gerçekte birçok grafik, kendi içerisinde farklı yerel yapılar barındırır.

Bir trafik ağı düşünün. İstanbul’daki iki yolun ilişkisi ile bir metro ağındaki iki istasyonun ilişkisi aynı şekilde yorumlanmayabilir. Bir molekülde iki atom arasındaki bağın anlamı, bir sosyal ağdaki iki kişi arasındaki bağlantıyla aynı değildir.

Yani “**edge = sadece bağlantı**” demek bazen yetersiz kalabilir.

Sheaf yaklaşımı bu bağlantının arkasındaki **yerel dönüşümü ve geometrik ilişkiyi** de modellemeye çalışır. Bu nedenle Sheaf Neural Networks, **Geometric Deep Learning** perspektifinde oldukça ilginç bir yere oturuyor.

Sheaf Laplacian: Grafiğin Matematikini Değiştirmek

Sheaf yaklaşımının yalnızca message passing mekanizmasına yeni bir katman eklemekten daha derin bir tarafı var. Burada **Sheaf Laplacian** kavramı devreye giriyor.

Klasik grafiklerde Laplacian, node’lar arasındaki ilişkilerin yapısını analiz etmek için temel araçlardan biridir. Sheaf Laplacian ise bu fikri, node’lar arasındaki bilgi dönüşümlerini de hesaba katacak şekilde genişletir.

Basitleştirilmiş olarak şu şekilde ifade edilebilir:

$$L_{\mathcal{F}} = \delta^T \delta$$

Buradaki δ , node'lar ile edge'ler arasındaki sheaf yapısının uyumsuzluklarını temsil eden bir operatör olarak düşünülebilir.

Bunun bize söylediği şey oldukça güzel:

Grafikte yalnızca “kim kime bağlı?” sorusunu değil, “bu bağlantı üzerindeki bilgiler ne kadar tutarlı?” sorusunu da sorabiliyoruz.

Bu da Sheaf Neural Networks'ün arkasındaki matematiksel fikrin neden klasik message passing'den daha zengin olduğunu gösteriyor.

Sheaf Neural Network'ler Nasıl Öğreniyor?

Buraya kadar sheaf'in ne olduğunu anlattık. Peki model bu dönüşümleri nereden biliyor?

İşte burada öğrenilebilir parametreler devreye giriyor. Restriction map'ler sabit olmak zorunda değil; model, eğitim sırasında bu dönüşümlerin parametrelerini öğrenebilir.

Örneğin bir edge için şöyle bir dönüşüm düşünülebilir:

$$\rho_{u,e}(x) = W_{u,e} x$$

Buradaki $W_{u,e}$ model tarafından öğrenilen bir dönüşüm matrisi olabilir.

Dolayısıyla ağ yalnızca “komşudan gelen bilgiyi nasıl birleştireyim?” sorusunu öğrenmiyor. Aynı zamanda:

“Komşunun bilgisini benim bilgi uzayımda nasıl dönüştürmeliyim?”

sorusuna da veri üzerinden cevap arıyor. Bu, Sheaf Neural Networks'ün en dikkat çekici taraflarından biri.

Her Şeyin Bir Bedeli Var

Buraya kadar Sheaf Neural Networks kulağa oldukça güçlü geliyor. Ancak bilimsel açıdan önemli olan nokta şu:

Sheaf Neural Networks, GNN'lerin her problemini sihirli şekilde çözmiyor.

Aksine, daha zengin bir modelleme gücü beraberinde yeni zorluklar getiriyor.

Daha Karmaşık Bir Yapı

Klasik bir GNN'e göre öğrenilecek dönüşümlerin sayısı ve yapısı artabilir. Node embedding'lerine ek olarak edge bazlı dönüşümlerin modellenmesi hesaplama ve optimizasyon maliyetini artırabilir.

Eğitim Kolay Değil

Modelin yalnızca node özelliklerini değil, aralarındaki dönüşümleri de öğrenmesi optimizasyon sürecini daha karmaşık hale getirebilir.

Her Problem Sheaf Gerektirmez

Eğer grafiğinizdeki node'lar zaten aynı temsil uzayında doğal biçimde karşılaştırılabiliriyorsa, sheaf yapısının getirdiği ek karmaşıklık gereksiz olabilir.

Yani mesele “Sheaf Neural Networks her zaman daha iyidir” değil. Asıl mesele:

“Problemimizde yerel temsil farklılıklarını ve edge bazlı dönüşümleri modellemek gerçekten gerekli mi?”

sorusudur.

GNN'den SNN'e Bakınca Ne Değişiyor?

Şimdi bütün hikâyeyi tek bir tabloda düşünelim.

Yaklaşım	Temel Soru
GNN	Komşularımdan hangi bilgileri öğrenebilirim?
Message Passing GNN	Komşuların mesajlarını nasıl aggregate edebilirim?
Sheaf Neural Network	Komşuların bilgilerini hangi dönüşümle karşılaştırmalıyım?

Bu fark küçük gibi görünse de aslında oldukça temel. GNN **graph structure** üzerinden öğrenir. Sheaf Neural Network ise buna **local representation + relation-dependent transformation** katmanını ekler.

Dolayısıyla Sheaf Neural Networks'ü GNN'lerin “bir sonraki versiyonu” olarak görmekten ziyade, **GNN'lerin temsil ve bilgi aktarımı konusundaki varsayımlarını genişleten bir yaklaşım** olarak düşünmek daha doğru olur.

Gelecek Nereye Gidiyor?

Yapay zekâ araştırmalarında son yıllarda önemli bir değişim yaşanıyor. Uzun süre modelleri büyütme odaklandık: daha fazla parametre, daha fazla veri, daha fazla katman, daha fazla hesaplama.

Ancak artık başka bir soru da giderek önem kazanıyor:

Bir model yalnızca daha fazla bilgi öğrenmek yerine, bilginin yapısını daha doğru anlayabilir mi?

Graph Neural Networks bize verinin bağlantılarından öğrenmeyi öğretti. Geometric Deep Learning, bu bağlantıların altında yatan yapıya daha fazla odaklandı. Sheaf Neural Networks ise bir adım daha ileri giderek şu soruyu gündeme getiriyor:

“Bağlantı yalnızca iki noktayı birbirine bağlayan bir çizgi mi, yoksa bilginin bir noktadan diğerine nasıl dönüştürüleceğini de belirleyen matematiksel bir yapı mı?”

Eğer ikinci seçenek doğruysa, grafikler artık yalnızca **node + edge** değildir. Aynı zamanda:

Node + Local Space + Relation + Transformation

haline gelir. Ve bu bakış açısı, özellikle heterojen, geometrik ve ilişkisel yapısı karmaşık veri kümelerinde yeni modelleme imkanları sunabilir.

Son Söz

Sheaf Neural Networks’ü anlamanın en kolay yolu aslında tek bir soruda yatıyor:

“Komşumun bilgisini neden olduğu gibi almak zorundayım?”

Klasik GNN’ler, grafik üzerindeki bağlantıları kullanarak komşulardan bilgi toplamamızı sağladı. Sheaf Neural Networks ise bu sürece daha ince bir soru ekledi:

“Peki bu bilgi benim bulunduğum yerde ne anlama geliyor?”

İşte sheaf yaklaşımının temel fikri burada yatıyor. Node’lar kendi yerel bilgi uzaylarına sahip olabilir. Edge’ler yalnızca bağlantı değil, dönüşüm bilgisi taşıyabilir. Ve bir grafiğin yapısı yalnızca bağlantıların varlığından değil, bu bağlantılar üzerinden bilginin nasıl ilişkilendirildiğinden de oluşabilir.

Belki de grafik tabanlı yapay zekânın geleceğinde asıl mesele daha fazla node veya daha fazla edge değil: **bağlantıların ne anlama geldiğini daha iyi anlayabilmek.**

Çünkü gerçek dünyada insanlar, şehirler, moleküller veya sistemler yalnızca birbirine bağlı değildir. **Birbirlerini dönüştürürler.**

Ve Sheaf Neural Networks’ün sorduğu soru tam olarak budur:

Bir bilgiyi başka bir yere taşıdığımızda, o bilgi hâlâ aynı bilgi midir?

Belki de cevap, yapay zekânın grafikleri nasıl gördüğünü değiştirecek kadar önemlidir.

AI DEEP- DIVE serisinde, yapay zekânın yalnızca ne yaptığını değil, bunu mümkün kılan matematiksel fikirleri de keşfetmeye devam edeceğiz.

Bir sonraki yazıda görüşmek üzere.