KARADENİZ TEKNİK ÜNİVERSİTESİ
OF TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YZM 4008
Veri Madenciliği Dersi Proje Raporu

MPSO-KNN ile Sezgisel Nitelik Ağırlıklandırma

369527 - Betül ÇALIK

2020 -2021 BAHAR DÖNEMİ

ÖZET

Karmaşık optimizasyon problemleri genel olarak doğrusal olmayan, süreksiz ve birçok tasarım parametresinden oluşur. Bu tür problemler geleneksel matematik yöntemleriyle çözülemediği için meta-sezgisel optimizasyon yöntemlerine başvurulur. Gerçek dünyadaki birçok problem çok boyutlu değişken barındırdığı için, doğadan ilham alınarak geliştirilmiş meta-sezgisel optimizasyon algoritmaları bu tür problemlerin çözümünde kullanılabilir.

Particle Swarm Optimization (PSO) algoritması, etkili ve basit olmasının yanı sıra birçok optimizasyon probleminin çözümünde yaygın bir şekilde kullanılmaktadır. Modified Particle Swarm Optimization (MPSO) algoritması, PSO algoritmasının dezavantajlarının üstesinden gelmek ve yeteneklerini güçlendirmek için uyarlanabilir bir strateji kullanmıştır.

MPSO algoritmasında, PSO'nun global keşif ve lokal kullanım yeteneklerini iyi dengelemek için kaos tabanlı doğrusal olmayan bir atalet ağırlığı (*inertia weight)* önermektedir. Aynı zamanda erken yakınsamayı önlemek için stokastik (*stochastic)* ve ana akım (*mainstream)* öğrenme stratejileri benimsenir. Son olarak, PSO'nun uzman sistemlerdeki karmaşık optimizasyon sorunlarını çözme kabiliyetini geliştirmek için uyarlanabilir bir konum güncelleme stratejisi ve terminal değiştirme mekanizması kullanılır.

Bu çalışmada MPSO algoritmasının KNN ile sezgisel ağırlıklandırılması konusu incelenecektir.

1. Giriş

Uzman sistemlerde herhangi bir problem; güç sistemi, yol planlama ve su tedarik sistemi tasarımı, operasyon yönetimi ve uydu ağı gibi optimizasyon yöntemleri olarak modellenebilir. Uzman sistem uygulamalarının oldukça geniş bir kullanım alanına sahip olduğu için karmaşık optimizasyon problemleri genellikle doğrusal olmayan, belirsiz, çok modlu, süreksiz ve yüksek boyutludur. Bu durum, yüksek performanslı bir optimizasyon algoritması geliştirmeyi etkin bir araştırma noktası haline getirmiştir.

PSO algoritması, araştırmacılardan giderek daha fazla ilgi gören ve uzman sistemlerde birçok gerçek optimizasyon problemini çözmek için başarıyla uygulanan (Kennedy & Eberhart, 1995) tarafından önerilen popülasyon tabanlı bir optimizasyon algoritmasıdır. Araç yönlendirme problemi, veri sınıflandırması, özellik seçimi, derin sinir ağı ve birçok gerçek problemin çözümünde kullanılmıştır.

PSO algoritması basit prensibine, kolay uygulamasına ve yüksek yakınsama oranına sahip olmasına rağmen erken yakınsama ve zayıf global arama yeteneği gibi bazı eksikliklere sahiptir. Araştırmacılar bu eksikliklerin üstesinden gelmek için PSO'nun çeşitli değiştirilmiş sürümlerini önermişlerdir.

Her ne kadar verilen PSO varyantları olgunlaşmış ve uzman sistemlerde gerçek optimizasyon problemlerini çözmek için başarılı bir şekilde uygulanmış olsa da, optimizasyon problemlerinde erken yakınsama ve zayıf performans problemleri hala mevcuttur. Ek olarak, insan toplumunun gelişmesiyle birlikte optimizasyon problemleri gittikçe karmaşıklaşmaya ve yüksek boyutlu, çok kısıtlamalı ve çok modlu olma eğiliminde olmaya devam etmektedir. Bu durum PSO'nun nasıl iyileştirileceği konusunda araştırmaların artmasını sağlamıştır.

MPSO algoritması, PSO'nun performansını güçlendirmek için modifiye edilmiştir. PSO algoritmasına ek olarak MPSO'da aşağıda açıklanan dört katkı vardır.

1) Arama adımını genişleterek veya daraltarak keşif kullanımını daha iyi dengelemek için kaos tabanlı doğrusal olmayan bir atalet ağırlığı önerilmiştir.

2) Stokastik ve ana akım öğrenme teknikleri, popülasyonun çeşitliliğini etkin bir şekilde arttırabilen ve erken yakınsamayı önleyebilen bir şekilde tasarlanmıştır. Ardından, algoritmanın uzman sistemlerde karmaşık optimizasyon problemlerini çözme yeteneği geliştirilmiştir.

3) Keşif ve kullanım sürecini dengelemek için adaptif bir konum güncelleme stratejisi geliştirilmiştir.

4) Doğadaki en güçlü olanın hayatta kalması kuralından esinlenilerek MPSO'nun yakınsama kesinliğini arttırmak için bir terminal değiştirme mekanizması benimsenmiştir. Böylece algoritmanın gerçek optimizasyon problemlerini çözme yeteneği daha da geliştirilmiştir.

Bu çalışmada güncel ve güçlü bir algoritma olan MPSO algoritması kullanılmıştır. Bu algoritmanın knn sınıflandırıcı için en iyi k değerinin belirlenmesi, probleme ait niteliklerin ağırlıklandırılması ve sezgisel sınıflandırma gibi gereksinimlere cevap vermesi amaçlanmıştır.

2. Materyal ve Yöntem (Material and Method)

Geliştirilen melez algoritmanın anlaşılmasına yardımcı olması için üç alt bölüm hazırlanmıştır. Takip eden bölümde k-nn sınıflandırıcı hakkında bilgi verilmektedir.

2.1. K-En Yakın Komşu Algoritması (k-Nearest Neigbour Algorithm)

KNN algoritması, denetimli öğrenme yöntemine sahip algoritmalardan biridir. Denetimli öğrenmede sınıflandırma ve regresyon amacı ile kullanılır. Temel olarak yeni noktaya en yakın olan noktalar aranır. K değeri yeni noktanın en yakın komşu sayısını belirtir.

Bağımsız değişkenleri sayısal veri tipi olmalıdır. Çünkü uzaklık esaslı bir algoritmadır. Bu algoritmada önemli olan nokta, eğitim aşamasının olmamasıdır. Önce veri seti oluşturulur, daha sonra uzaklık ölçülür. Veriler arasındaki mesafe ölçülürken farklı uzaklık metrikleri kullanılabilir.

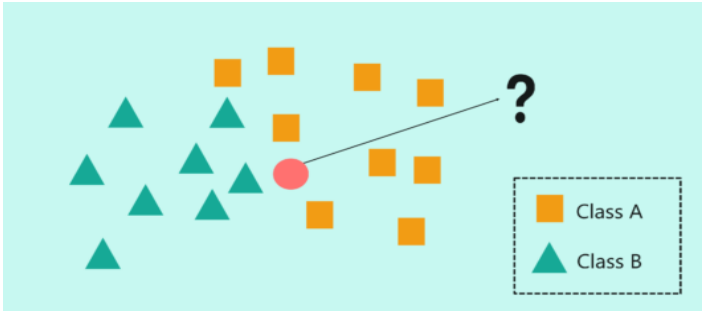
KNN algoritmasında k değeri sabit bir değer değildir. Problem tanımına göre k sayısı değişkenlik gösterebilir. K değeri seçilirken sorgu örneğine en yakın olan veriler dikkate

alınmalıdır. İdeal alt sınır ve ideal üst sınır belirlenmelidir. K değeri belirlenirken deneme/yanılma yöntemine başvurulabilir.

Algoritmanın adımları aşağıdaki gibi verilebilir:

1. Sınıf nitelikleri belli olan elemanlardan oluşan bir uzaya yeni bir örnek ekle.
2. K sayısını belirle.
3. Yeni gelen örnek ile elemanlar arasındaki uzaklığı hesapla.
4. En yakın olan k tane elemanın sınıf niteliğine bak.
5. En yakın olan k tane elemanlardan sınıf değeri sayıca fazla olan sınıfa örneği dahil et.

KNN sınıflandırma algoritmasına dair görsel bir örnek aşağıda verilmiştir.



Şekil 1. Temsili KNN

Knn algoritmasına ait sözde kod aşağıdaki gibi verilebilir.

Algoritma 1. k-nn Algoritmasının Sözde Kodu (Kahraman vd., 2013)(Pseudo Code of the k-nn Algorithm)

1. Veri setinin tanımlanması:probleme ait n-adet örnek gözlemleri içeren ve problem uzayını temsil etme kabiliyeti yüksek (gözlem uzayını homojen olarak örnekleyen) Xveri setini oluştur.
2. Uzaklık bağıntısının belirlenmesi: gözlemler arasındaki uzaklıkların hesaplanmasında kullanılacak yöntemi belirle.
3. k-değerinin belirlenmesi:gözlem sayısına ve veri setinin karakteristiğine bağlı olarak k-komşu sayısı için arama uzayının sınırlarını tanımla
4. for each kj
   kj için sınıflandırma performansını SPkj= fk-nn(kj)
   if(SPkj > SPkj-1)
      k=kj
   end if
   end
5. En iyi sınıflandırma performansı sağlayan k-değerini kaydet
6. Sınıf etiketi belirlenecek olan q sorgu gözlemini tanımla
7. for i=1:n
      D[i]=qile Xiarasındaki uzaklığı hesapla
   end
8. Xq[k]=D[i] uzaklık dizisinden q sorgu gözlemine en yakın k-adet gözlemi belirle
9. Xq[k] gözlemlerinin sınıflarını dikkate alarak çoğunluk oylaması/ağırlıklı oylama yöntemi kullanarak q-gözleminin sınıfını belirle

2.2.MPSO Sezgisel Arama Algoritması (MPSO Heuristic Search Algorithm)

MPSO algoritması, PSO algoritmasının iyileştirilmiş bir versiyonu olmak ile birlikte popülasyon tabanlı stochastic arama algoritmasıdır. Popülasyona swarm (sürü) ve popülasyondaki her bir bireye particle (parçacık) adı verilmektedir.

Popülasyonda her bir parçacığın konumu

(1)	$x_i^d = x_{min}^d + rand(x_{max}^d - x_{min}^d)$

ve hızı

(2)	$v_i^d = v_{min}^d + rand(v_{max}^d - v_{min}^d)$

bulunmaktadır.

MPSO algoritmasında, PSO algoritmasından farklı olarak dört adet yaklaşım sergilenmiştir:

1. Kaotik Tabanlı Atalet Ağırlığı (Chaotic Based Inertia Weight)

- PSO algoritmasında her bir parçacığın kendisine göre en iyi konumu Pbest, tüm parçacıkların en iyi konumu ise Gbest olarak adlandırılmaktadır.
- Atalet ağırlığı, parçacıkların farklı ortamlardaki dinamik ağırlığına katkıda bulunduğu için PSO algoritmasında önemli bir konuma sahiptir.
- Atalet ağırlığı, her nesildeki parçacıkların yerel en iyi (Pbest) ve global en iyi (Gbest) değerlerinin bir fonksiyonu olarak tanımlanır.
- MPSO algoritmasında atalet ağırlığı PSO algoritmasından farklı olarak kaotik tabanlıdır. Lineer olmayan bir atalet ağırlığı elde edilerek daha güçlü uyum ve simülasyon yeteneği kazanılmıştır.

(3)	$\omega(t) = r(t)\cdot\omega min + \frac{(\omega max - \omega min)\,.\,t}{Tmax}$

2. Stokastik ve Ana Akım Öğrenme Stratejileri (Stochastic and Mainstream Learning Strategies)

- Parçacıklar hızlarını ve konumlarını güncellemek için Pbest ve Gbest'i öğrenmelidirler.
- PSO algoritmasında bu öğrenme stratejisi PSO'nun hızlı yakınsama, yüksek güvenilirlik avantajlarına sahip olmasını sağlamaktadır. Ancak yine bu strateji, erken yakınsama ve karmaşık problemlerde düşük performans gibi eksikliklere yol açar.
- Bu yüzden MPSO algoritmasında stokastik ve ana akım öğrenme stratejileri önerilmiştir.

(4) $Vi(t+1) = \omega(t)Vi(t) + r1c1 \otimes (SPbesti(t) - Xi(t)) + r2c2 \otimes (Mbest(t) - Xi(t))$

Denklemin sağ tarafındaki ilk bileşen atalet ağırlığı, ikinci bileşen stokastik öğrenme bileşeni ve üçüncü bileşen de ana akım öğrenme bileşenidir.

- Böylece PSO'nun erken yakınsama probleminin aşılması amaçlanmıştır.

3. Adaptif Pozisyon Güncelleme Stratejisi (Adaptive Position Updating Strategy)

- Farklı konum güncelleme stratejilerinin farklı keşif ve sömürü yeteneklerine sahip olduğu iyi bilinmektedir. Yerel sömürü ve küresel keşfi dengelemek için, uyarlanabilir bir konum güncelleme mekanizması önerilmektedir.

(5) $$p_i = \frac{exp(fit(X_i(t)))}{exp(\frac{1}{N}\sum_{i=1}^{N} fit(X_i(t)))}$$

6

(6) $$Xi(t+1) = \begin{cases} \omega(t)Xi(t) + (1-\omega(t))Vi(t+1) + Gbest(t) & pi > rand \\ Xi(t) + Vi(t+1) & otherwise \end{cases}$$

- Yukarıda verilen denklemlere göre pi değeri küçük olduğunda lokal pozisyon güncellenir, pi değeri büyük olduğunda ise global pozisyon güncellenir.

4. Terminal Değiştirme Mekanizması (Terminal Replacement Mechanism)

- Popülasyonun karmaşık problemlerdeki performansını iyileştirmek için terminal değiştirme mekanizması kullanılmaktadır.
- Her iterasyonda global en kötü parçacık olan Gworst değiştirilmektedir.
- Daha sonra çaprazlama ile birlikte kişisel en iyi olan Nbest üretilir.
- Gworst ve Nbest karşılaştırılarak değiştirme mekanizması uygulanmaktadır.

(7)	$Gworst(t) = \max\{Pbest1(t), Pbest2(t), \cdots, PbestN(t)\}$
(8)	$Nbest(t) = Gbest(t) + rand \cdot (Pbest_j(t) - Pbest_k(t))\ j \neq k$
(9)	$Gworst(t) = \begin{cases} Nbest(t) & fit(Nbest(t)) < fit(Gworst(t)) \\ Gworst(t) & \text{otherwise} \end{cases}$

- Yukardaki denklemlere göre 7. denklemde Gworst bulunur, 8. denklemde Nbest üretilir ve 9. denklemde ise terminal değiştirme mekanizması uygulanır.

MPSO algoritmasına ait sözde kod aşağıda verilmiştir.

Algoritma 2. MPSO Algoritmasının Temel Adımları

i) Problemin oluşturulması
ii) Çözüm adayının tasarımı ve çözüm adayları topluluğunun oluşturulması
iii) Çözüm adayların uygunluk değerlerinin hesaplanmas
iv) İteratif süreç

- Ubest'I güncelle
- Adaptif konum güncelleme stratejisi
- Terminal değiştirme mekanizması
- If fit(Xi) < fit(Pbesti)
  Pbesti = Xi;
  Fit(Pbesti) = fit(Xi);
  End
  If fit(Pbesti) < fit(Gbest)
  Pbesti=Gbest;
  Fit(Gbest)=fit(Pbesti);
  end

2.3. Önerilen Yöntem: MPSO-KNN ile Sezgisel Nitelik Ağırlıklandırma (Proposed Method: Heuristics for Attribute Weighting with MPSO-KNN)

MPSO-KNN algoritması ile birlikte niteliklerin ağırlıklandırılması: Sezgisel k-nn algoritması ile niteliklerin probleme etkisi incelenip buna göre ağırlıklandırılması işlemi yapılmaktadır.

3. Deneysel Sonuçlar (Experimental Results)

Önerilen yöntemin sonuçlarını tespit etmek adına mevcut veri seti ile karşılaştırma yapılmıştır. Karşılaştırma kapsamında MPSO-KNN algoritması farklı iterasyon sayılarında çalıştırılarak kendi aralarında karşılaştırma işlemi gerçekleştirilmiştir.

3.1. Ayarlar (Settings)

Algoritma ayarları için Tablo-1'de gösterilen değerler aynı şekilde kullanılmıştır.

Algoritma	Parametre Değerleri
MPSO	N=50, Max_Fes=300, k=5
KNN	Uzaklık bağıntısı=Öklid ve Manhattan

Tablo 1. Ayarlar

3.2. Veri Seti (Dataset)

Algoritmada kullanılan veri setine ilişkin genel özellikler Tablo-2'de verilmiştir.

	Boyut	Eğitim Örnek Sayısı	Test Örnek Sayısı
Veri Seti	5	258	145

Tablo 2. Veri Seti

3.3. Önerilen Yöntemin Sonuçları (Results of Proposed Method)

Önerilen yöntemin sonuçları Tablo-3'te verilmiştir.

MPSO-KNN		
	Öklid	**Manhattan**
En İyi Değer	5 (%3.44)	4 (%2.75)
En Kötü Değer	45 (%31.03)	46 (%31.72)
Ortalama Değer	17.8 (%12.27)	9.7 (%6.68)
Standart Sapma	6.9 (%4.75)	5.6 (%3.86)