Metin Madenciliğinde Büyüyen Öz-Düzenlemeli Harita Ağlarının Kullanımı

Zafer İşcan
İstanbul Teknik Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümü, İstanbul
iscanz@itu.edu.tr

Özet: Çalışmada, metin madenciliğinde Büyüyen Öz-Düzenlemeli Harita (GSOM) Ağlarını temel alarak, demetleme zamanının azaltılmasına yönelik farklı yaklaşımlar sergileyen iki bildiri incelenmektedir. İlk bildiride, iki-seviyeli büyüyen öz-düzenlemeli harita kullanan bir demetleme yöntemi tanıtılmaktadır. Demetleme işleminin önemli bir kısmı, gelişen grid teknolojisi kullanarak farklı bilgisayarlarda gerçekleştirilebilecek alt-işlemlere bölünmektedir. Bu sayede, genel olarak toplanan bilginin hızlı analizi mümkün olmaktadır. Önerilen metodun başarımı geleneksel yaklaşımlara yakın olmakla birlikte, uygulama zamanını 15 kat iyileştirmektedir. İkinci bildiride ise, HDGSOMr adı verilen, SOM'un büyüyen bir çeşidine dayalı yeni bir algoritma tanıtılmaktadır. Bu yeni algoritma, daha küçük komşuluk boyutları kullanarak, az sayıda iterasyon içerisinde daha kaliteli demetler üretmek için rastgeleliği öz-düzenleme işlemine dahil etmektedir. Böylelikle, toplam işlem zamanı önemli ölçüde azalmaktadır.

Anahtar Sözcükler: Öz-Düzenlemeli Harita, Veri Madenciliği, Metin Madenciliği.

Use of Growing Self Organizing Maps in Text Mining

Abstract: In the study, growing self-organizing map (GSOM) based two papers that perform different approaches for decreasing clustering time in text mining are examined. In the first paper, a clustering method using growing self organizing map in two steps is presented. Most significant amount of the clustering process is divided into sub-processes which can be performed in different computers by using the evolving grid technology. Therefore, a quick analysis of the acquired information becomes possible. Performance of the proposed method is close to the traditional approaches. However, application time is improving 15 times. In the second paper, a new algorithm (HDGSOMr) based on a growing version of SOM is introduced. The new algorithm adds randomness to the self-organizing process in order to generate more qualified clusters in a few iterations by using smaller neighborhood. Thus, total time of the operation decreases significantly.

Keywords: Self-Organizing Map, Data Mining, Text Mining.

1. Giriş

Metin verilerinin madenciliği, veri hacimlerinin artmasına paralel olarak günümüzde bir gereklilik haline gelmiştir [1]. Verilerin çok büyük hacimlerde olması, onları bulmada, işaretlemede ve gruplamada sıkıntı oluşturmaktadır. Yeni ve gelişen teknolojiler (Anlamsal Web, W4 (World Wide Wisdom Web) ve web hizmetleri çerçeveleri) gelecekte bilgiyi düzenleme ve geri elde etmenin daha iyi yollarını sağlayacak olsalar da, bu teknolojilerle iyileştirilemeyecek milyonlarca web sayfası mevcut olacaktır ve bu verilerin çoğunluğu az yapılı ve düzenli metinsel bilgi olacaktır [2].

Metin bilgisi topluluklarına bir miktar düzen getirmenin bir yolu, üzerlerinde metin maden-

ciliği yaparak belgeler arasındaki benzerliği tanımlayıp, ona göre bunları gruplamaktır. Zaman içerisinde farklı teknolojilere dayalı birçok metin madenciliği algoritması geliştirilmiştir. Bu algoritmalardan en popüler olanları, genelde öznitelik haritası algoritmaları olarak bilinen Öz-Düzenlemeli Harita'ya (SOM) [3] dayanan algoritmalardır. Bu algoritmaların popüler olmasının nedeni, oluşturulan demet dağılımlarının görselliğini daha iyi ifade etme yetenekleridir. SOM'un değişik sürümlerinin en çok konu edilen problemlerinden biri, haritanın optimal yüksekliğini ve genişliğini bulmaktır. GHSOM [4], Büyüyen Grid [5], Artımsal Grid Büyüme [6], DASH [7] ve GSOM [8] gibi SOM'un birçok büyüyen sürümünde bu konu ele alınmıştır. Ayrıca, SOM tabanlı algoritmaların genetik algoritmalar [9], K-ortalamalar [10], daha hızlı kazanan araması [11] ve büyük haritaların küçük haritalar ile [12] başlatılmasıyla, son haritaya çabucak yakınsatılarak optimize edilmesine çalışılmıştır.

İncelenen birinci bildiride [1], ölçeklenebilirliği sağlamak için Gelişen Grid (ızgara) Hesaplama teknolojisinden faydalanılmıştır. Grid, hesap ve veri yoğunluğu fazla olan problemlerin çözümünde, dağınık vaziyetteki ayrık kaynakların paylaşımını sağlamaktadır. Metin madenciliği uygulamalarının da, demetleme işlemi bağımsız alt-işlemlere bölünebildiği ve ayrı olarak icra edilebildiği takdirde, ölçeklenebilirliği arttırmak için grid'den yararlanabileceği aşikârdır.

İncelenen ikinci bildiride [2], SOM ve diğer öznitelik haritası algoritmalarının değişik adımlarında başarımı arttırmak için rastgelelik kullanılmıştır. Ağırlık vektörlerinin rastgele başlatılması, girişlerin öznitelik haritasına rastgele sırada verilmesi, bu uygulamalardan bazılarıdır. Rastgeleliğin HDGSOM [13, 14] adı verilen yüksek boyutlu büyüyen öz-düzenlemeli haritaya dahil edilme sonuçları gösterilmiştir. Değişen algoritma, rastgeleliği göstermek için HDGSOMr adını almıştır. Önerilen algoritmadan elde edilen sonuçlar, öz-düzenleme safhasındaki rastgeleliğin kullanımının, kullanılmadığı duruma göre daha kısa zaman içerisinde daha iyi sonuç verebileceğini göstermiştir.

2. Öz-Düzenlemeli Harita Algoritmaları

Bu bölümde temel SOM ve GSOM algoritmaları hakkında genel bilgiler verilmektedir.

2.1 SOM Algoritması

Öz-Düzenlemeli Harita (SOM) [4], yüksek boyutlu girişi harita üzerinde benzer demetler birbirine yakın olacak şekilde düşük boyutlu topolojiye çeviren bir deneticisiz sinir ağı modelidir. SOM'un her düğümü kendisine ait ve başlangıçta rastgele oluşturulan bir ağırlık vektörüne sahiptir. Her giriş vektörü ağa verilir ve giriş vektörüne en yakın düğüm, girişi izleyecek şekilde güncellenir. Genellikle Gauss komşuluğu fonksiyonuyla belirlenen kazanan (en yakın) düğümün çevresindeki düğümler de düzgün bir haritalama üretmek için girişe uyarlanırlar.

2.2 GSOM Algoritması

SOM'un bir uzantısı farklı şekillerde ve boyutlarda büyüyebilen Dinamik SOM'dur [5, 12, 15]. GSOM az sayıda düğümle başlar. Büyüme oranı, dolayısıyla son harita boyutu, 0 ile 1 arasında (1 maksimum büyümeyi verir) değişen yayılma çarpanı (Spread factor (SF)) ile kontrol edilir [14]. Bu, aşağıda gösterilen büyüme eşiğini (GT) belirlemekte kullanılır:

$$GT = -D \ln(SF) \qquad (1)$$

D, giriş vektörlerinin boyutunu göstermektedir. Veri analisti farklı SF değerleri kullanarak daha genel gruplamadan daha ince bölmelemeye kadar seçme imkânına sahiptir. Büyüme safhasında, kazanan düğüm belirlendiğinde, hata değeri - genelde Euclid mesafesi tarafından belirlenen giriş ve ağırlık vektörü arasındaki fark - o düğümün hata sayacına eklenir. Kazanan bir düğüm için toplam hata GT'den büyükse ve kazanan düğüm boş komşu alana sahipse (sınır düğümse), yeni bir düğüm kazananın yanında oluşturulur. Aksi halde, hatanın yarısı bütün komşu düğümlere dağıtılır.

3. Ölçeklenebilir Büyüyen Öz-Düzenlemeli Harita Algoritması (sGSOM) [1]

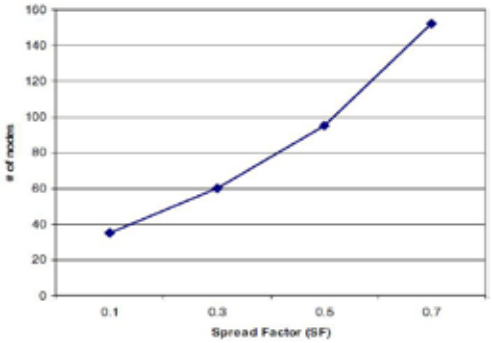
Önerilen çalışmada [1], SOM yerine GSOM kullanılmıştır. Bunun nedeni, SOM'un haritanın boyutunu ve şeklini önceden bilmeye ihtiyacı olmasıdır. GSOM'da ise, yeni bir düğüm ancak ihtiyaç olduğunda eklendiğinden, bu sorun aşılmaktadır. Bu dinamik yapı, GSOM'u özellikle ani olarak değişen belge toplulukları için uygun hale getirmektedir [1].

Demetleme işlemini ölçeklenebilir ve verimli yapmak için, 3 hedefin [10] başarılması amaçlanmıştır:

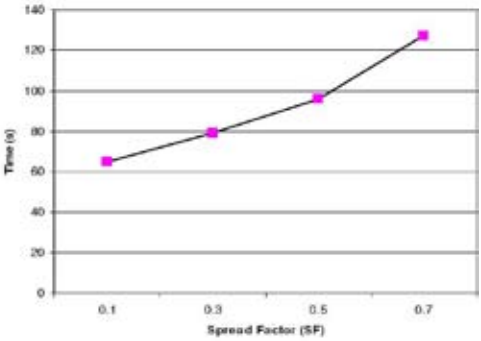
- Orjinal işi, her biri diğerlerinin sonuçlarından bağımsız alt-işlere bölmek.
- Alt-iş sayısı ve her alt-işe verilen giriş verisini iletişim sürelerini kompanze edebilecek kadar büyük yapmak.
- Merkezi bilgisayar ile işlenen her düğüm arasındaki veri transferini azaltmak.

Sonuç olarak, GSOM'un başlangıçtaki tek büyüme safhasının, iki farklı büyüme safhasına ayrılması önerilmiştir [1]. İlk büyüme safhası tek bir süreç olarak gerçekleştirilirken, çıkışı farklı bilgisayar düğümlerinde aynı anda koşan ikinci büyüme safhasını başlatmakta kullanılır. Öncelikle GSOM'la bütün giriş verisinin birinci büyüme safhasında düşük yayılma faktörüyle genel ve hızlı gruplaması elde edilmektedir [1]. Böylece, aralarında yüksek seviyede ayrım olan az sayıda yoğunlaşmış düğüme sahip öznitelik haritası üretilir. Her düğüm içinden elde edilen veriler farklı bilgisayarlara gönderilip bağımsız bir şekilde ayarlanabilir. Bunlar, daha iyi demetlemeyi sağlamak için daha yüksek yayılma katsayısına sahip başka bir GSOM'la işlenecektir ve bir büyüme, iki ayarlama safhalı normal prosedürü takip edecektir. Sonuçta üretilen bütün çıkışlar merkezi bilgisayara gönderilecek ve diğer işlemler için birleştirilecektir.

Uygun yayılma çarpanı (SF) değeri seçimi kritik bir konudur [1]. Şekil 1 ve Şekil 2'ye bakıldığında, yüksek yayılma çarpanının merkezi bilgisayar için daha uzun işlem zamanı gerektireceği ve ikinci hedefe uymayan aşırı sayıda düğüm üreteceği (düğüm başına düşen veri miktarını azaltacağı) görülmektedir. Ayrıca, öznitelik haritasını aşırı genel ya da iyileşmiş yapan bir SF, demetleme kalitesini de düşürebilir (Tablo 2). Bununla birlikte, SF'in kesin değerlerini seçmek, uygulama için gerekli genelleme seviyesine de bağlı olduğundan, deneyseldir [1].



Şekil 1. SF – Düğüm sayısı eğrisi [1]



Şekil 2. SF – Zaman eğrisi [1]

3.1 Belge Toplulukları

Bu çalışmada, iki metin topluluk kümesi kullanılmıştır [1]. İlk küme, farklı yaklaşımlar arasındaki demet kalitesini karşılaştırmak için ABC haber arşivindeki 5 alan içerisinden alınan 150 haber makalesi topluluğunu içermektedir. İkinci küme, Usenet tartışma grubundan alınan [3]'teki WEBSOM çalışmalarında kullanılan forum makalelerinin altkümesinden oluşturulmuştur.

3.2 Önişleme

Belgeler, HTML etiketleri, başlık bilgisi ve kullanıcı imzaları gibi metinsel-olmayan içeriği temizlemek için ön-işlemeye tabi tutulmuşlardır [1].

3.3 Demet Değerlendirme Yöntemi

SOM, GSOM ve ölçeklenebilir GSOM kullanarak farklı sonuçlar arasındaki kaliteyi karşılaştırmak için [9]'daki gibi standart F-ölçütü kullanılmıştır. Tablo 1'de, birinci veri kümesi için farklı yaklaşımların sonucu gösterilmektedir [1].

Yöntem	F-Ölçütü
SOM	0.49
GSOM	0.51
Ölçeklenebilir GSOM	0.50

Tablo 1. ABC haber arşivi için F-ölçütü [1]

Tablo 1'den görüldüğü gibi, elde edilen haritaların kalitesi her üç yöntem için de karşılaştırılabilirdir [1]. İki GSOM öznitelik haritası oluşturulacağından (bir tane kelime kategori haritası (w) için ve bir tane belge haritası (d) için), iki SF değeri seçilmesi gerekmektedir. Tablo 2'de her iki harita üzerinde farklı SF kullanma sonuçları gösterilmiştir [1]. Öznitelik haritası aşırı genelse (SF(w)=0.2, SF(d)=0.2) ya da detaylıysa (SF(w)=0.8, SF(d)=0.8), GSOM'un başarımı kötüleşmektedir.

SF(d)=0.2				
SF(w)	0.2	0.4	0.6	0.8
F-ölçütü	0.35	0.43	0.49	0.51
SF(w)=0.8				
SF(d)	0.2	0.4	0.6	0.8
F-ölçütü	0.51	0.52	0.43	0.39

Tablo 2. GSOM'da SF'in F-ölçütüne etkisi [1]

3.4 Uygulama Zamanını Karşılaştırma Ölçeklenebilir GSOM yöntemini kullanarak, öncelikle 0.1 SF değerli, 65 saniyede 35 düğüm üreten (Şekil 3) bir kaba kelime kategorisi oluşturulmuştur.

Daha sonra her düğüm ayrı olarak 0.3 SF değerli GSOM kullanarak iyileştirilmektedir.



Şekil 3. Kaba kelime kategori haritası [1]

Tablo 3'te kelime kategori haritası üretmek için tek bir GSOM ile, ölçeklenebilir GSOM kullanma sonuçları özetlenmektedir [1].

Yöntem	Toplam Düğüm	Zaman(s)
GSOM	**402**	**5407**
sGSOM	**752**	**347**

Tablo 3. Usenet makaleleri için kelime haritası üretimi uygulama zamanı [1]

Ölçeklenebilir yöntem için gerekli zamanın, geleneksel yöntemden 15 kat daha az olduğu açıktır [1]. Bununla birlikte, ölçeklenebilir yöntem sadece benzetim durumunda test edilmiştir ve Grid üzerinde çalıştığında oluşacak haberleşme gecikmeleri ve iş bekleme zamanlarını göz önüne almamaktadır. Bu idealleştirmeye rağmen, ölçeklenebilir durum yine de oldukça umut verici bir başarım gelişimi sergilemekte ve haberleşmeden kaynaklanan sorunlar için gereğinden fazla zaman bırakmaktadır [1].

4. HDGSOMr Algoritması [2]

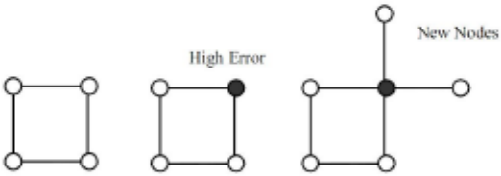
HDGSOM [13], GSOM'un [8] yüksek boyutlu veriye uygulandığındaki durumuna işaret eden bir uzantısıdır [2]. Minimum sayıda düğüm sayısından (genelde 4) başlayarak dışarıya doğru büyüyen orijinal GSOM algoritması metin madenciliğindeki gibi yüksek boyutlu verilere uygulandığında hızlı büyüyerek, eğilmiş ve bükülmüş haritalar oluşturan düğümlerle sonuçlanır. Bunun temel nedeni çok büyük boyutlarda oluşan daha büyük hatalardır.

HGDSOM algoritması, büyüme eşik değerini,

birkaç düzeltme safhasıyla birlikte birkaç adımda küçültür ve daha düzgün bir büyüme sağlar [13]. Sonrasında büyüme safhası SOM'un işlemlerine benzeyen daha fazla düzgünleştirme safhalarını takip ederek orjinal GSOM algoritmasından daha iyi öznitelik haritası oluşturur. HGDSOM algoritması aşağıda kısaca açıklanmıştır [13].

4.1 Başlangıç Safhası

HDGSOM Şekil 4'teki gibi dikdörtgen yapıda birbirine bağlı 4 düğümden oluşturulur ve rastgele ağırlık vektörleri bu düğümlere atanır.



Şekil 4. HGDSOM algoritmasındaki yeni düğüm büyümesi [2]

İki büyüme eşik değeri (GT1 ve GT2) algoritmanın büyüme safhasında kullanmak üzere hesaplanmaktadır [2]:

$$GT_1 = -50\ (\mu+2\sigma)\ \ln(SF) \qquad (2)$$
$$GT_2 = -50\ (\mu+2\sigma)\ \ln(SF)$$

SF, 0 ile 1 arasında değişen, veri analistinin haritanın yayılımını kontrol etmesini sağlayan yayılma çarpanı, μ ve σ ise girişlerin sıfır-olmayan boyutlarının ortalama ve standart sapmasını ifade etmektedir. SF'in yüksek değerleri, ayrıntılı demetler veren geniş haritalar oluştururken, küçük SF değerleri daha yoğun demetler oluşturmaktadır. SF genelde 0.1 olarak alınmaktadır [2].

4.2 Büyüme Safhası

HDGSOM'un büyüme safhasında, girişler SOM'daki gibi düğümlere verilir ve kazanan düğüm belirlenir. Giriş ve ağırlık vektörü arasındaki hata SOM'daki gibi kazanan düğümün ve komşularının ağırlıklarını güncellemekte kullanılır. Eğer bir düğümde hesaplanan hata, büyüme eşiği (GT) adı verilen eşikten büyükse ve o düğüm sınır düğümüyse, yeni düğümler Şekil 4'deki gibi bütün boş alanlarda yetişecektir [2].

Yeni düğüm oluşup oluşmayacağını belirleyen büyüme eşiği (GT), her büyüme döneminde eşit adımlarla GT1'den GT2'ye doğru değişir. Büyüme dönemleri SOM'daki gibi 2-3 düzgünleştirme dönemini takip eder. Her dönemin sonunda her düğüm için hesaplanan hata değeri sıfırlanır. Yeni düğümler en az bir dönem haritada gözükmeyen düğümlerden yetişemez [2].

4.3 Düzgünleştirme (smoothing) Safhaları

Büyüme safhasını SOM'un düzgünleştirme safhasıyla neredeyse aynı olan iki düzgünleştirme safhası takip eder. Bu iki safha daha duru demetler oluşturmak için haritayı düzgünleştiren azalan öğrenme hızlarına (α) sahiptir [2].

4.4 HDGSOM algoritmasında rastgeleliğin tanıtılması

HDGSOM üzerindeki birçok deneyde, daha düzgün harita oluşturmak için daha uzun düzgünleştirme sürelerine ihtiyaç olduğu belirlenmiştir. Bu da, büyük boyutlardaki geniş veri kümeleri için maliyeti arttıran daha uzun işlem zamanı demektir [2]. Düzgünleştirme safhaları, ağırlık vektörlerinin daha büyük değerlerle güncellenmesini sağlayacak daha yüksek öğrenme hızıyla uygulanırlarsa, bu safhalar daha verimli olacaktır. Büyük öğrenme hızlarındaki daha iyi sonuçlar, algoritmanın küçük öğrenme hızlarında yerel minimuma yakınsadığını göstermektedir [2]. Yerel minimumla ilgili problemlerin üstesinden gelmek amacıyla kullanılan rastgelelikten esinlenerek, öz-düzenleme safhasına rastgeleliğin dahil edilmesine karar verilmiştir [2]. Bu işlem, kazanan düğümün ve komşu düğümlerin ağırlıklarının, hatanın rastgele bir değerdeki oranıyla güncellenmesiyle sağlanmaktadır. Sonuç güncelleme, rastgele olarak eğitim miktarını arttıracak veya azaltacaktır. Bunu sağlamanın en kolay yolu, öğrenme hızını (α) rastgele bir sayıyla değiştirmektir. Değiştirilen ağırlık güncelleme denklemi denklem 3'de gösterilmiştir [2]:

$$w_i^n = w_i^0 + [\ \alpha + 2\alpha\ (r - 0{,}5)\ ]\ \eta\ (\ x_i - w_i^0\ ) \quad (3)$$

w_i^n = Düğümün i. bileşeninin yeni ağırlığı

w_i^0 = Düğümün i. bileşeninin eski ağırlığı

x_i = Girişin i. bileşeninin değeri

α = öğrenme hızı

r = 0 ile 1 arasında rastgele bir sayı

η = Ağırlığa komşuluk fonksiyonu katkısı. (Kazanan için 1, komşuluk mesafesi arttıkça sıfıra yaklaşmaktadır.)

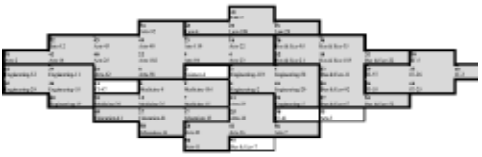
Denklem öğrenme hızını kendisinin belirli bir oranıyla arttırır ya da azaltır. Değiştirilmiş ağırlık güncelleme denklemi, HDGSOM'un bütün öz-düzenleme bölümlerine uygulanmıştır.

4.5 (HDGSOMr) algoritmasındaki Deneyler

Seçilen ilk veri kümesi, metin demetleme algoritmalarının değerlendirilmesinde kıyas veri kümesi olarak sıkça kullanılmış olan 20 haber grubu veri kümesidir [16]. İkinci veri kümesi Monash Üniversitesi'nin 2002 rehberidir. Rehber, 2002'de öğretilen her konunun özetini fakülte ve bölüm bazında vermektedir. Algoritmayı önerenler tarafından geliştirilen bir web madenciliği robotu, monash.edu.au/pubs/2002handbooks internet adresinden sayfaları çıkarmak ve sonrasında başlığı, konu kodunu, fakülte ve özeti çıkarıp işlemek üzere kullanılmıştır. Demetleme işleminde yalnızca özet içeriği kullanılmış, diğerleri sonuçların analizinde kullanılmıştır [2]. Bütün veri kümesi ön-işlemeye tabi tutulmuştur: Veri kümelerindeki metin alınmış ve metinsel-olmayan içerik giderilmiştir. Metin daha sonra kelimelere ayrılmış ve Porter'in stemming algoritması [17] kullanılarak stem edilmiştir. 'a', 'the', 'of' gibi yaygın olarak kullanılan kelimeleri gidermek için bir durma kelimesi listesi kullanılmıştır. 50'den fazla belgede ya da dosyada var olmayan kelimeler yine elenmiştir. Kalan kelimeler, TF-IDF yöntemi [18] kullanılarak kodlanmış ve algoritmanın giriş vektörlerini oluşturmak üzere normalize edilmiştir. Demetlemede bir gelişme olup olmadığını görebilmek amacıyla, HGDSOM ve HGDSOMr algoritmasının her aşamasındaki iterasyon sayısı azaltılmıştır. Bu, HGDSOM için daha az kaliteli demetler oluşturan bir haritayla sonuçlanacak ve rastgelelikten kaynaklanan gelişme varsa HGDSOMr için daha kaliteli demetler oluşturan bir haritayla sonuçlanacaktır [2]. Şekil 5 - 6'da, rehber veri kümesi üzerinde HDGSOM ve HDGSOMr tarafından üretilen haritaları açıklayan sonuçlar gösterilmiştir [2]. Şekil 5 ve Şekil 6'da, baskın fakülte isimleri ve haritalanma sayısı düğümler üzerinde gösterilmiştir [2]. Düğümler sonuçların kolay değerlendirilmesi açısından birlikte gruplandırılmıştır.



Şekil 5. Rehber veri kümesinden HDGSOM ile elde edilen harita [2]



Şekil 6. Rehber veri kümesinden HDGSOMr ile elde edilen harita [2]

Gösterilen sonuçların, her iki algoritma için de rastgeleliğin etkisini görmek adına iterasyon sayısının az seçilmesi nedeniyle optimum olmadığına dikkat edilmelidir [2].

20 haber grubu veri kümesinde HDGSOM algoritması sadece az sayıda demet üretirken, HGDSOMr algoritmasında, fark edilebilir bir iyileşme elde edilmiştir.

Rehber veri kümesiyle, HDGSOM algoritmasının haritası 66 düğümün 12'sini (%18.18)

gruplayamazken, HDGSOMr algoritmasının haritası 66 düğümün yalnızca 5 tanesini (%7.94) gruplayamamıştır [2].

5. Sonuçlar ve Değerlendirme

İncelemeler sonucunda, sGSOM'un HDGSOMr'ye göre daha önemli bir çalışma olduğu düşünülmektedir. Her ne kadar, sGSOM gerçek bir Grid yapısıyla denenmemişse de, demetleme işlemini birbirinden bağımsız parçalara ayırıp, normale göre çok daha kısa bir süre içinde gerçekleştirebilmesi bakımından daha umut vaad eden bir yöntem olarak gözükmektedir. HDGSOMr, rastgelelik sayesinde, aynı süre içerisinde HDGSOM algoritmasından çok daha iyi bir demetleme yapmaktadır. Ancak burada iterasyon sayısının az seçildiği unutulmamalıdır. İterasyon sayısı büyüdükçe, HDGSOM'un başarımının, HDGSOMr'ye göre artacağı düşünülmektedir.

Bundan sonraki çalışmalarda, yayılma faktörünün - dolayısıyla büyüme eşiğinin - optimum olarak belirlenmesi ve düğümlerin farklı büyüme şekillerinin araştırılacağı aşikârdır. Gelişen teknolojiye bağlı olarak, bilgisayarların işlem gücü artsa da, veri boyutları da gittikçe daha büyük hale gelmektedir. Bu yüzden, metin madenciliğinde kullanılan algoritmaları daha hızlı hale getirmeye yönelik yapılacak her gelişme, büyük önem taşıyacaktır.

6. Kaynaklar

[1] Zhai, Y. Z., Hsu, A. and Halgamuge, S. K., "Scalable Dynamic Self-Organising Maps for Mining Massive Textual Data", **ICONIP 2006, Part III, LNCS** 4234, 260 – 267, (2006).

[2] Amarasiri, R., Alahakoon, D., Smith, K. and Premaratne, M., "HDGSOMr: A High Dimensional Growing Self-Organizing Map using Randomness for Efficient Web and Text Mining", **Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)**, 215-221, (2005).

[3] Kaski, S., Honkela, T., Lagus, K. and Kohonen, T., "WEBSOM—Self-Organizing Maps of Document Collections", **Neurocomputing**, 21:101–117, (1998).

[4] Kohonen, T., "Self-organizing Maps", **Springer-Verlag,** Berlin, (1995).

[5] Alahakoon, D., Halgamuge, S. K. and Srinivasan, B., "Dynamic Self-Organising Maps with Controlled Growth for Knowledge Discovery", **IEEE Transactions on Neural Networks, Special Issue on Knowledge Discovery and Data Mining**, 11(3), (2000).

[6] Lagus, K., Kaski, S. and Kohonen T., "Mining Massive Document Collections by the WEBSOM Method", **Information Sciences**, 163(1-3),135-156, (2004).

[7] Honkela, T., "Self-Organizing Maps in Natural Language Processing," **Ph.D. thesis,** Helsinki University of Technology, Neural Networks Research Center, Espoo, Finland, (1997).

[8] Nürnberger, A., "Interactive Text Retrieval Supported by Growing Self-Organizing Maps", **Proceedings of the International Workshop on Information Retrieval**, 61-70, (2001).

[9] Larsen, B. and Aone, C., "Fast and Effective Text Mining using Linear Time Document Clustering," **Proceedings of the conference on Knowledge Discovery and Data Mining,** 16-22, (1999).

[10] Depoutovitch, A. and Wainstein, A., "Building Grid Enabled Data-Mining Applications", **http://www.ddj.com/184406345**, (2005)

[11] Salto, G., "Developments in Automatic Text Retrieval", **Science**, 253:974-979, (1991).

[12] Hsu, A. and Halgamuge, S.K., "Enhancement of Topology Preservation and Hie-

rarchical Dynamic Self-Rrganising Maps for Data Visualisation", **International Journal of Approximate Reasoning**, 32/2-3, 259-279, (2003).

[13] Hsu, A., Tang, S. and Halgamuge, S. K., "An Unsupervised Hierarchical Dynamic Self-Organising Approach to Class Discovery and Marker Gene Identification in Microarray Data", **Bioinformatics, Oxford University Press,** (2003).

[14] Alahakoon, D., "Controlling the Spread of Dynamic Self Organising Maps", **Neural Computing and Applications,** 13(2):168-174, Springer Verlag, (2004).

[15] Wickramasinghe, L.K and Alahakoon, L. D., "Dynamic Self Organizing Maps for Discovery and Sharing of Knowledge in Multi Agent Systems in Web Intelligence and Agent Systems", **An International Journal, (IOS Press)**, 3(1), (2005).

[16] Lang, K., "Newsweeder: Learning to filter netnews", **Twelfth International Conference on Machine Learning,** (1995).

[17] Porter, M., "An algorithm for suffix stripping", **Program**, 14:130-137, (1980).

[18] Salton, G. and Buckley, C., "Term-weighting approaches in automatic text retrieval", **Information Processing & Management,** 24(5):513-523, (1988).