Geniş Veri Kümeleri Üzerinde Paralel Veri Madenciliği Yaklaşımları: Wavecluster Yöntemi ile Öbekleme Uygulaması

Ahmet Artu YILDIRIM, Bilg. Müh., Müh. Mim. Fak., Çankaya Üni. 100.Yıl/ANKARA, artu@computer.org

Efe ÇİFTCİ, Bilg. Müh., Müh. Mim. Fak., Çankaya Üni. 100.Yıl/ANKARA, efeciftci@cankaya.edu.tr

Cem ÖZDOĞAN, Bilg. Müh., Müh. Mim. Fak., Çankaya Üni. 100.Yıl/ANKARA, ozdogan@cankaya.edu.tr

Depolama boyutlarının artması ve günümüzde bilgi ediniminin daha kritik hale gelmesi ile birlikte geniş veri kümeleri üzerinde paralel veri madenciliği çalışmaları hızla artmakta ve önem kazanmaktadır. Bu çalışmamızda, öncelikle veri madenciliği hakkında bilgi verilecek ve sonrasında da veri madenciliğinde paralel hesaplamanın kullanımı tartışılacaktır. Bu kapsamda literatürde geliştirilmiş olan temel paralel veri madenciliği yaklaşımları ve bu yaklaşımların birbirine kıyasla yarar ve yarar yitimleri tanıtılacaktır. Çalışmamızda, veri madenciliği konusunda son zamanlarda sıkça kullanılmaya başlanılan ve doğası gereği koşut zamanlı (paralel) çalıştırılmaya uygun olan wavecluster algoritması temel alınmıştır. Wavecluster algoritması, veri kümesi üzerinde wavelet dönüşümü uygulayarak benzer küme üyelerini bulmada kullanılan çok boyutlu bir öbekleme algoritmasıdır. Bu algoritmanın paralelleştirilmesinde kullanılan yöntem ve çalışmamızda elde edilen başarım grafikleri ile kazanımları anlatılacaktır. Paralel uygulamalarda en çok kullanılan başarım ölçütleri; hızlanma ve verimliliktir. Çankaya Üniversitesi'nde bulunan bilgisayar öbeğinde yapılan bu çalışmada elde ettiğimiz sonuçlar, geliştirilen algoritmanın geniş veri kümeleri üzerinde paralel veri madenciliği yapmak için uygun olduğunu göstermektedir.

Anahtar Kelimeler: *Veri madenciliği, wavecluster yöntemi, paralel algoritmalar.*

1. Giriş

Veritabanı teknolojilerinin gelişimi ile birlikte kayıt altına alınan veri miktarında önemli bir artış ortaya çıkmıştır. İşletmeler arası rekabetin artması ve hızlı bilgi ediniminin kritik hale gelmesi ile depolanan bu veri yığınlarından anlamlı bilgi çıkarmak oldukça büyük önem kazanmıştır. Veri madenciliği başlığı altında bulunan öbekleme analizi, veri kümesi içerisindeki verileri belirli bir benzerlik kıstasına göre önceden tanımlanmamış sınıflara ayırma işi olarak özetlenebilir. Öbekleme analizi; örüntü tanımada, coğrafi bilgi sistemlerinde (GIS), görüntü işlemede, makine öğrenmesinde, web dokümanlarının sınıflandırılmasında vs. kullanılan önemli bir araçtır. Öbekleme analizi algoritmalarından biri olan wavecluster algoritması ise veri kümesi üzerinde wavelet dönüşümü uygulayarak benzer küme üyelerini bulan çok boyutlu bir öbekleme algoritmasıdır [11]. Veri kümelerinin boyutlarının artık terabaytlar seviyelerine gelmiş olması ve kullanılan algoritmanın hesap karmaşıklığına bağlı olarak, günümüz bilgisayarları (özellikle bellek miktarı ve işlemci gücü açısından) hala yeterli olamamaktadır. Bu nedenle hem çalışma zamanını kısaltmak hem de kaynakları etkin bir şekilde kullanmak için geniş veri kümeleri üzerinde pek çok paralel öbekleme çalışmaları yapılmaktadır [4, 8, 9]. Bu makalemizde, wavecluster algoritmasını önce seri (sıralı) kod olarak yazdık ve sonrada koşut zamanlı hale getirdik. Yaptığımız çalışmalar paralel wavecluster algoritmasının geniş veri kümeleri üzerinde paralel veri madenciliği yapmak için uygun olduğunu göstermiştir.

Bu çalışma şu şekilde yapılanmıştır. Çalışmanın ikinci bölümünde veri madenciliği ve uygulama alanı geniş olan bazı veri madenciliği algoritmaları hakkında bilgi vereceğiz. Üçüncü bölümünde temel paralel veri madenciliği algoritmalarını ve yaklaşımlarını tartışacağız. Dördüncü bölümde paralel wavecluster yöntemini, beşinci bölümde ise çalışmamızda geniş veri kümelerini kullanarak çıkardığımız hızlanma ve verimlilik grafiklerini ve elde ettiğimiz sonuçlar verilecektir.

2. Veri Madenciliği

Bilişim teknolojilerinin gelişimi ile birlikte, günümüzde büyük veri kümeleri ile karşılaşmamız artık çok olağan bir hale geldi. Veri madenciliği, bu büyük miktarlardaki verinin içinde bulunan "gizli" örüntüleri keşfederek anlamlı bilgi çıkarma işi olarak tanımlanabilir. Veri madenciliğinin kullanılmasının temel nedeni, eldeki bu büyük veri yığınlarındaki bulunan örüntülerden elde edilen anlamlı bilgileri kullanarak geleceğe dönük tahminde bulunabilme ve veri kümesi içersindeki elemanları tanımlayabilme ihtiyacıdır [6]. Günümüzde veri madenciliği; astronomide sinir ağları tekniğini kullanarak yıldız-galaksi ayrımının yapılmasında [10], pazarlama alanında birliktelik kuralları öğrenimi yönteminin kullanılmasıyla müşteri eğilimlerinin tahmin edilmesinde [1], finans kuruluşlarında kredi kartı dolandırıcılarının tahmin edilmesinde [5], kuantum kimyası alanında kristal örgü yapılarının tahmin edilmesinde [7] ve daha pek çok alanlarda sıkça karşımıza çıkmaktadır.

Veri madenciliği teknikleri dört temel sınıfa ayrılmaktadır [6]. Bunlar sınıflandırma, öbekleme (kümeleme), regresyon ve birliktelik kuralları öğrenimi olarak tanımlanmaktadır. Bu sınıfları kısaca özetlemek gerekirse:

- Sınıflandırma: Veri kümesi elemanlarını önceden tanımlanmış sınıflara ayırarak, tahminleme yapar,
- Öbekleme: Verileri belirli bir benzerlik kıstasına göre önceden tanımlanmamış sınıflara ayırma işidir,
- Regresyon: En az hata sapmasına sahip veri elemanlarını ifade eden fonksiyonu bularak geleceğe dönük tahminleme yapar,
- Birliktelik Kuralları Öğrenimi: Veri kümesini oluşturan kayıtların arasındaki ilişkileri araştırır.

3. Paralel Veri Madenciliği

Mevcut veri madenciliği algoritmaları üzerinde günümüze kadar pek çok iyileştirmeler yapılmasına karşın, geniş veri kümeleri üzerinde veri madenciliği algoritmalarının çalışma süresi hala oldukça uzundur. Paralel veri madenciliğinin amacı, sıralı çalışan veri madenciliği algoritmasının yaptığı işi, işlemciler arasında dağıtarak koşut zamanlı hale getirmektir. Veri madenciliği algoritmalarının koşut zamanlı hale getirilmesinde izlenen üç farklı yaklaşımdan bahsedebiliriz [12]. Bunlar bağımsız arama, paralelleştirilmiş sıralı veri madenciliği ve tekrarlı sıralı veri madenciliği yaklaşımlarıdır. Bu yaklaşımlar arasında hangisinin daha iyi sonuç verdiği, kullanılan veri madenciliği algoritmasına göre farklılık göstermesiyle birlikte, genelde tekrarlı sıralı veri madenciliği yaklaşımı daha iyi sonuç vermektedir.

Bu yaklaşımlarda temel olarak yapılan üç temel iş vardır. Bunlar hesaplama, işlemciler arasındaki veri iletişimi ve veri kümesine erişimdir. Bu işler arasında en fazla zaman alan iş, veri kümesine erişimdir.

3.1.Bağımsız Arama Yaklaşımı

Bağımsız arama yaklaşımında, her işlemcinin bütün veri kümesine erişimi vardır ve tüm işlemciler birbirinden bağımsızdır. Her işlemci için rastgele arama uzayı belirlenmesinin ardından, işlemci aynı algoritmayı ρ defa çalıştırır. İşlemciler çıkardıkları sonuçları birbirleri ile değiş tokuş ederler ve bu sonuçlar arasından en iyi sonuç seçilir. Örnek olarak biyolojik bir uygulama olan kalıtım algoritmasında, her bir işlemci farklı rastgele kromozomdan başlayarak aynı algoritmayı çalıştırır. Algoritmanın sonunda her bir işlemcinin ürettiği hata payı en az olan sonuç en iyi sonuç olarak kabul edilir. Bu metodun en büyük yarar yitimi tüm işlemcilerin bütün veri kümesine erişebiliyor olmasıdır. Bu açıdan oldukça maliyetli bir yoldur. Her makinede veri tabanının tamamının tutulması gerekebilir. Bu da hafıza kullanımını arttırmaktadır. Diğer taraftan algoritmada, işlemciler sadece hesaplamanın en sonunda elde ettikleri sonuçları dağıttıklarından, işlemciler arasındaki veri iletişiminin de en az olduğu yaklaşımdır.

3.2.Paralelleştirilmiş Sıralı Veri Madenciliği Yaklaşımı

Bu yöntem "kavram bilgilerini dağıtma" işi üzerine kuruludur. Öncelikle birincil kavramlar işlemciler arasında dağıtılır. Daha sonra her işlemci veri kümesinin tamamı üzerinde veya bir kısmı üzerinde veri

madenciliği algoritmasını çalıştırır. Algoritmanın çalıştırılmasından sonra oluşan yeni kavramlar işlemciler arasında değiş tokuş edilir ve genel olarak doğru olmayan kavramlar elenir. Bu yöntem, işlemciler arasında değiş tokuş edilecek hiç bir kavram olmayana dek devam eder. Bu metoda örnek olarak birliktelik kuralları bulma algoritmasının paralelleştirilmesinde kullanılan veri dağıtımı algoritmasını (Data Distribution) verebiliriz [2]. Bu teknikte, her iterasyonda sık veri kümeleri işlemciler arasında paylaştırılır ve işlemci tüm veri kümesine ulaşarak gerekli destek sayma işlemlerini yapar. Bu algoritmada işlemciler her iterasyonda kavram bilgilerini değiş tokuş ettiklerinden dolayı veri iletişimi açısından pahalı bir algoritmadır.

3.3.Tekrarlı Sıralı Veri Madenciliği Yaklaşımı

Bu yöntemde, veri kümesi tüm işlemciler arasında paylaştırılır ve her işlemci veri madenciliği tekniğini kendi yerel veri kümesi üzerinde uygular. Sonuç yerel olarak doğru iken genel olarak doğru olmayabilir. Bu yüzden algoritmanın sonunda her işlemci kendi çıkardığı yerel değerleri diğer işlemciler ile değiş tokuş eder ve bu yerel değerler birleştirilerek genel olarak doğru olan değerler oluşturulur. Bu metoda örnek olarak yine birliktelik kurallarının paralelleştirilmesinde kullanılan sayma dağıtımı algoritmasını (Count Distribution) verebiliriz [2]. Bu algoritmada, her bir işlemci için veri kümesi paylaştırılır ve her işlemci kendi yerel veri kümesi üzerinde aday kümelerinin oluş sayısını sayarak destek değerlerini hesaplar. Çıkan sonuç yerel olarak doğru iken genel olarak değildir. O yüzden tüm işlemciler aday kümeleri için geçerli olan destek değerlerini diğer işlemciler ile paylaşır ve bu yerel destek değerleri toplanarak genel destek değerleri hesaplanır. Genel destek değeri daha önceden tanımlanan minimum destek değerinden küçük olan aday kümeler elenirler ve sık veri kümeleri oluşur. Sık veri kümeleri artık oluşmadığında algoritma sonlanır ve minimum güven aralığına göre kurallar çıkartılır. Bu algoritmada, her iterasyonda işlemciler elde ettikleri bilgileri birbirleri ile değiş tokuş etmektedirler ve bu açıdan bağımsız arama tekniğine göre veri iletişimi daha fazladır.

4. Paralel Wavecluster Yöntemi

Wavecluster yöntemi, geniş veri kümeleri üzerinde kesikli wavelet dönüşümü uygulayarak farklı örüntülere sahip olan karmaşık kümeleri bile keşfedebilme özelliğine sahip, ızgara tabanlı bir öbekleme algoritmasıdır [11]. Veri kümesi üzerinde kesikli wavelet dönüşümünün uygulanmasının temel amacı, bu dönüşümü uygulayarak veri kümesi içerisinde bulunan elemanların arasındaki uzaklığı azaltmak ve bu şekilde dönüştürülmüş daha yoğun bir veri kümesi elde etmektir. Kesikli wavelet dönüşümü veri kümesine birden fazla uygulanarak, farklı doğruluk seviyelerinde (hassastan daha kabaya) küme elemanları keşfedilebilir.

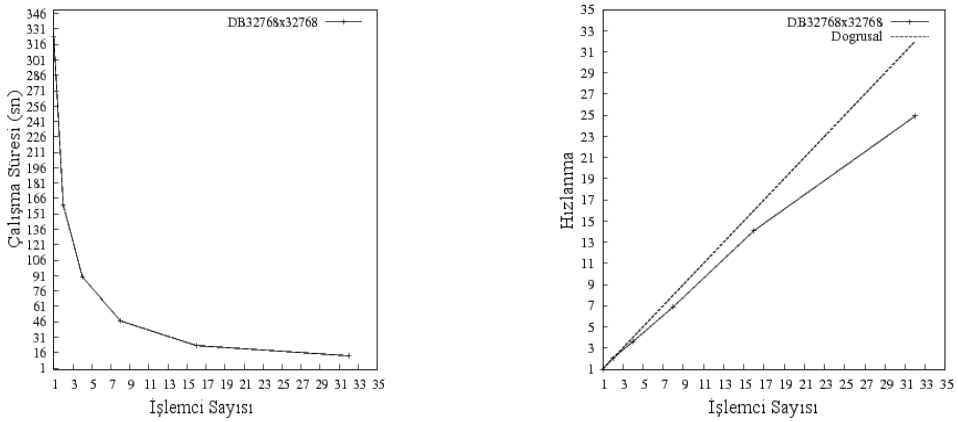
Algoritmada ilk olarak, η boyuta, $1 \leq \eta$, sahip öznitelik uzayının her bir boyutunun önceden belirlenmiş aralık değer kümelerine bağlı olarak sayısallaştırılması işlemi yapılır. Bu işlem aynı zamanda farklı aralık değer kümeleri için başarım ve kümeleme kalitesini de etkileyen önemli bir faktördür. İkinci adım olarak da, sayısallaştırılmış öznitelik uzayı üzerinde kesikli wavelet dönüşümü işlemi uygulanır. Kesikli wavelet dönüşümünün 2 boyutlu öznitelik uzayına uygulanması sonucunda 3 adet detay sinyali olarak bilinen yüksek frekanslı bileşenler ve 1 adet ortalama sinyali olarak bilinen düşük frekanslı bileşen ortaya çıkar. Üçüncü adımda, dönüşümde kullanılan veri kümesinden daha yoğun olan düşük frekanslı bileşen kullanılarak üzerinde bağlı parçaları işaretleme (connected components labeling) algoritması uygulanır ve her bir parçanın komşuluk özelliğine bağlı olarak küme elemanları tespit edilir. Bu adım farklı doğruluk derecelerinde küme elemanlarını tespit etmek için birden fazla uygulanabilir. Son adımda ise basit bir işlem ile dönüşümden geçmiş her bir eleman, özgün uzaydaki birden fazla eleman ile eşleştirilerek çözümleme işlemi yapılır ve kümeleme işlemi sonlandırılır.

Wavelet algoritması etkin bir algoritma olmasına karşın küme tespiti için kullanılan veri kümelerinin çok büyük boyutlarda olması ile birlikte bellek alanları yeterli olamamaktadır. Bu sorunun üstesinden gelmek ve kaynakları daha etkin kullanarak algoritmanın çalışma zamanını kısaltmak için tekrarlı sıralı veri madenciliği yaklaşımı izlenmiş ve paralel wavecluster algoritması geliştirilmiştir. Çalışmamızı yaptığımız öbek altında bulunan bilgisayarlardan bir tanesi ana birim görevinde, diğer bilgisayarlar ise ana birime bağlı olarak çalışan alt birim görevinde çalıştırılmaktadırlar. Ana birimin amacı alt birimlerden gelen sınır verilerini karşılaştırarak alt birimlere küme eşleme değerlerini bildirmektir. Alt birimlerin görevi ise geniş

veri kümesinden kendi kısmına düşen alt veri kümesini almak, bu alt veri kümesi üzerinde wavecluster algoritmasını uygulamak, sınır verilerini ana birime yollamak ve ana birimden aldığı sınır eşleme değerleri ile birlikte yerel olarak doğru fakat genel olarak yanlış olan küme numaralarını güncellemektir. Son olarak sonuç veri kümesi, orijinal veri kümesine geri dönüştürülerek kümeleme sonucu yazdırılır.

5. Sonuçlar

Geliştirdiğimiz yazılım, paralel wavelet algoritmasının uygunluğunu değerlendirmek için Çankaya Üniversitesinde bulunan bilgisayar öbeği üzerinde çalıştırıldı [3]. Algoritma C programlama dili kullanılarak geliştirildi. İşlemciler arasındaki iletişimin sağlanması için mesaj tabanlı arayüze sahip olan OpenMPI Kütüphanesi; veri yapılarından ve bazı yardımcı fonksiyonlarından faydalanmak için ise Glib kütüphanesi kullanıldı. Paralel uygulama, 2.34 GHz gücünde 32 işlemci üzerinde yürütüldü. Çalışmamızda veri kümesi üzerinde üç defa wavelet dönüşümü uygulandı ve başarım grafikleri oluşturuldu.



(a) Çalışma Süresi Grafiği ***(b)*** *Hızlanma Grafiği*

Şekil-1:*Başarım Grafikleri*

Hızlanma katsayısı 1 nolu formülde gösterildiği gibi; birden fazla işlemcili paralel sistemin, tek işlemcili sıralı sisteme kıyasla göreceli performans artışına denir. t_s tek işlemcili sıralı sistemin çalışma süresi, t_p ise birden fazla işlemcili sistemin algoritmayı paralel çalıştırma süresidir.

$$Hızlanma = \frac{t_s}{t_p} \qquad (1)$$

32768x32768 çözünürlüğünde yapay olarak üretilen iki boyutlu veri kümesi üzerinde yaptığımız çalışma sonucunda, Şekil 1(b)'de görüldüğü gibi algoritmamızın doğrusallık karakteristiği taşıdığını gördük. Bu verilerin ışığında geliştirdiğimiz paralel wavecluster algoritmasının geniş veri kümeleri üzerinde paralel veri madenciliği yapmak için oldukça uygun olduğu sonucuna vardık.

6. Kaynakça

[1] Agrawal, R., Imieliński, T., Swami, A., "Mining association rules between sets of items in large databases", SIGMOD `93: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216 (1993).

[2] Agrawal, R., Shafer, J. C., "Parallel mining of association rules", IEEE Transactions on Knowledge and Data Engineering, 86, 962-969, (1996).

[3] Boron, "http://siber.cankaya.edu.tr/boron-ganglia/", (2010).

[4] Boutsinas, B., Gnardellis, T., “On distributing the clustering process”, Pattern Recognition Letters, 238, 999-1008, (2002).
[5] Brause, R., Langsdorf, T., Hepp, M., “Neural data mining for credit card fraud detection”, ICTAI `99: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, 103, (1999).
[6] Fayyad, U., Piatetsky-shapiro, G., Smyth, P., “From data mining to knowledge discovery in databases”, AI Magazine, 17, 37-54, (1996).
[7] Fischer, C. C., Tibbetts, K. J., Morgan, D., Ceder, G., “Predicting crystal structure by merging data mining with quantum mechanics”, Nature Materials, 51, 641-646, (2006).
[8] Kantabutra, S., Couch, A., “Parallel k-means clustering algorithms on nows”, Nectec Technical Journal, 16, 243-248, (2000).
[9] Lo Bosco, G., “Pgac: A parallel genetic algorithm for data clustering”, CAMP `05: Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception, 283-287, (2005).
[10] Odewahn, S., Stockwell, E., Penning-ton, R., Humphreys, R., Zumach, W., “Automated star/galaxy discrimination with neural networks”, Astronomical Journal, 1031, 318-331, (1992).
[11] Sheikholeslami, G., Chatterjee, S., Zhang, A., “Wavecluster: A wavelet-based clustering approach for spatial data in very large databases”, The VLDB Journal, 83-4, 289-304, (2000).
[12] Skillicorn, D., “Strategies for parallel data mining”, IEEE Concurrency, 74, 26-35, (1999).