ÇEŞİTLİ TİP VERİTABANLARINDA HASSAS BİLGİ GİZLEME

Osman Abul
Bigisayar Mühendisliği Bölümü, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara, Türkiye
osmanabul@etu.edu.tr

ABSTRACT

Many different approaches for knowledge hiding, hiding rules/patterns that can be inferred from published data and considered sensitive, have emerged over the years, mainly in the context of frequent itemsets mining. The hiding is usually obtained by a sanitization process that transforms data such that sensitive information can not be inferred with the objective of minimal loss in data quality. Following many real-world data and application demands, in this paper we shift the problem of knowledge hiding to various kinds of databases including sequential, spatio-temporal, graph, 3D-graph and time series.

Keywords: sensitive knowledge hiding, data sanitization, data publishing, data mining

1. GİRİŞ

Bilgisayarlar ve otomatik veri toplama aygıtlarındaki hızlı gelişmelerin sonucu olarak, kurumlar veri-aç yapıdan veri-tok yapıya ulaşmışlardır. Buna örnek vermek gerekirse, bir hastanın klinik muayenesinde bile hastayla ilgili onlarca veri otomatik olarak toplanabilmektedir. Hâlihazırda kurumlar topladıkları veriyi çoğunlukla gündelik işlemlerini sürdürmek için kullanmakta; nadiren bu verileri analiz etmekte ve bulgularını süreç iyileştirme, iş zekası vb. amaçlı kullanmaktadırlar. Öte yandan tüm kurumların kendi verilerini tamamen analiz etmesi beklenmemelidir. Çünkü veri analizi günümüzde bir uzmanlık alanı olarak ortaya çıkmış ve envai çeşit istatistikî ve veri madenciliği (VM) sorguları şeklinde formüle edilebilmektedir. Hatta kurumda yeterli veri analiz uzmanı olması ya da bu hizmetin satın alınması durumunda bile veriden elde edilebilecek tüm bilgilerin çıkarılması söz konusu değildir. Bunun nedeni ise sınırsız sayıda amaç için veritabanının kullanılabilmesidir. Dolayısıyla, buradaki en geçerli çözüm veritabanının üçüncü parti kullanıcılara açılması ve kullanıcıların kendi ilgi alanı ve amacına uygun olarak veritabanını kullanabilmesinin sağlanmasıdır (İngilizce tabiri ile "*do-it-yourself computation*").

Veritabanlarının paylaşılmasının/yayınlanmasının toplumsal refaha ve bilimsel birikime katkı gibi çok sayıda yararı olduğu aşikârdır. Kolayca tahmin edileceği üzere bu paylaşımın sakıncaları da mevcuttur ve bu yüzden kurumlar veritabanı paylaşımına isteksiz olabilmektedirler. En çok bilinen sakıncalar güvenlikle ilgilidir ve şu şekilde saymak mümkündür;

- ticari gizlilik: verinin ticari bir değer taşıması ve dolayısıyla rakiplerin eline geçmesinin istenmemesi,
- hassas bilgi: veri içerisinde kurumca hassas kabul edilen bilgi olması,
- mahremlik: verinin gerçek şahıslara ait mahrem bilgi içermesi (mikro veri durumu),
- gizlilik: verinin örneğin askeri geçiş yolları vb. bilgiler içermesi.

Bu sakıncalar, teknik açıdan bakıldığında veri güvenliği kısıtları olarak düşünülebilir. Dolayısıyla, veritabanı yayınlamanın bahsi geçen kısıtlar altında yapılması kurumların veritabanı yayınlama isteksizliğinin önüne geçilmesinde bir çözüm olacak ve veritabanı paylaşmanın avantajlarından yararlanmayı mümkün kılacaktır.

Güvenlikli veritabanı yayınlamada bilgi gizleme (*knowledge hiding*) ve veri sersemletme (*data perturbation*) iki önemli uygulama alanıdır. Her iki durumda da orijinal veritabanı bir temizlik işlemine tabi tutulmakta ve temizlenmiş veritabanı yayınlanmaktadır. Bilgi gizlemede, hassas bilginin temizlenmiş veritabanı üzerinde VM yapılsa dahi çıkarımsanamaması hedeflenir. Veri gizlemede ise temizlenmiş veritabanının hassas veriyi silmesi/değiştirmesi fakat istatistikî olarak orijinal veritabanına eşit olması hedeflenir. Her iki durumda ise ortak özellik, temizlenmiş veritabanından geçerli VM modellerinin elde edilebilmesidir.

Şimdiye kadarki anlatımda veritabanı kavramı soyut bir biçimde ele alınmıştır. Gerçekte ise veritabanları saklanan verinin doğasına bağlı olarak niteliksel farklılık arz etmektedir; örneğin, ilişkisel, sıralılar, mekan-zaman izleri, çizge, üç boyutlu çizge ve zaman serileri tipinde veritabanlarından bahsedilebilir. Uygulama alanlarındaki çeşitliğe bağlı olarak veri tipi gereksinimleri de zamanla çeşitlenmektedir. Örneğin, mobil cihazların yaygınlaşması yüksek hacimli mekan-zaman izleri tipi verileri gündeme getirmiştir. Veritabanları ve VM veri güdümlü disiplinler olduğundan, her bir veri tipi farklı yaklaşımlar ve algoritmaları gerekli kılmaktadır.

Bu çalışmada, çeşitli tipteki veritabanlarının hassas bilgi gizlenerek yayınlanması problemi ele alınmıştır. Devam eden kısımlarda, birliktelik analizinin hedef VM uygulaması düşüncesi altında

çeşitli tipteki veritabanlarından hassas bilginin etkin bir şekilde uzaklaştırılması için problem tanımı ve özellikleri, çözüm stratejileri ve algoritmaları üzerinde durulmuştur.

2. BİRLİKTELİK

Market-sepeti verisinde her bir alışveriş bir çoklu (*tuple*) ve bu alışverişteki her bir parçamal (*item*) ise bir özniteliğe karşılık gelir. Birliktelik problemi parçamalların sık olarak beraber satın alındığı ile ilgilidir. Örneğin bir milyon alışveriş sepetinin yüzbininde ekmek ve peynir beraber alınması, bu mallar arasında tüketici davranışı açısından bir bağımlılık olduğunu gösterir. Bu örnekten de görüleceği üzere bu bilgi pazarlama ve rekabet açısından önemlidir; çünkü müşteri davranışları hakkında doğrudan bilgi vermektedir. Sıklık değerine destek (*support*) adı verilir. Belli bir destek eşik değerinin üzerinde desteğe sahip olan parçamal kümelerine sık parçamal kümesi (*frequent itemset*) denir. Sık parçamal kümesi madenleme problemi ise bu koşulu sağlayan tüm parçamal kümelerinin bulunmasıdır.

Yukarıda verilen tanımların matematiksel modeli [2]'de verilmiştir. Bu modelde D bir parçamal kümesi veritabanını ve her bir $T \in D$ ise bir parçamal kümesini temsil eder. Modelde $I = \{i_1, i_2, \ldots, i_m\}$ verilen bağlamda yer alan tüm parçamalları (ürünleri) belirtir ve $T \subseteq I$ sağlanır. Ayrıca her bir $T \in D$ için biricik bir $T.tid$ (*identifier*) numarası vardır. X bir k-parçamal kümesi (k adet parçamal içeren, $0<k<m$) olmak üzere, T hareketi X parçamal kümesini ancak ve ancak $X \subseteq T$ şartı sağlanıyorsa içerir. Bu durumda T hareketi X parçamal kümesini destekler denilir. Buradan herhangi bir T hareketinin kendi tüm alt kümelerini desteklediği, bunun haricinde hiçbirşeyi desteklemediği sonucuna varılabilir. Herhangi bir X parçamal kümesinin D'deki desteği ise $X \subseteq T$ şartını sağlayan hareketlerin sayısıdır. X parçamal kümesinin D veritabanındaki destek değeri matematiksel olarak aşağıdaki gibi tanımlanır.

$$sup_D(X) = |\{T.tid : X \subseteq T \text{ ve } T \in D\}|$$

D veritabanında, verilen bir eşik değeri, σ, ve üzerinde sıklığa sahip tüm parçamal kümeleri sık parçamal kümesi, F, olarak adlandırılır ve matematiksel olarak aşağıdaki gibi tanımlanır.

$$F(D, \sigma) = \{X : X \subseteq I \text{ ve } sup_D(X) \geq \sigma\}$$

Yukarıdaki formülden anlaşılacağı üzere, F kümesi I kümesinin tüm alt kümelerinin (2^I) önce üretilip sonra bunların destek değerlerinin D'de sayılması yöntemiyle bulunabilir. Fakat kolaylıkla görüleceği üzere eğer herhangi bir $X \subseteq I$ için $X \notin F(D, \sigma)$ sağlanıyorsa, $X \subseteq Y \subseteq I$ özelliğini sağlayan bir Y parçamal kümesinin sık olması mümkün değildir. Bu özellik *Apriori* özelliği olarak bilinmektedir. Bu özelliğin doğal bir sonucu ise $X \subseteq I$ için $X \in F(D, \sigma)$ sağlanıyorsa, tüm $Y \subseteq X$ ler için $Y \in F(D, \sigma)$ sağlanıyor olmasıdır. Bu özellik F kümesinin, gereksiz saymaların yapılmaması ve dolayısıyla hızlı bulunması için çok yararlıdır.

Literatürde birliktelik sorgusu için çok sayıda algoritma önerilmiştir. Bunlar arasında en bilinenleri *AIS* [1], *Apriori* [2], *DHP* [3], *Partition* [4] ve FP-Growth [5] algoritmalarıdır.

2.1. Çeşitli Tip Veritabanlarında Birliktelik

Birliktelik analizleri sadece market sepeti tipi veritabanlarıyla sınırlı olmayıp, çeşitli veritabanları için adreslenmiş durumdadır.

Sıralılar veritabanları bir sıralılar kümesidir. Her bir sıralı ise birbirini takip eden sembollerden (her biri önceden belirlenmiş olan bir alfabenin elemanı) oluşur [6]. Sıralı örüntü (*sequential pattern*) lerin klasik tanımı ise sıralıların tanımı ile aynıdır. Bir sıralı örüntü eğer bir sıralının altsıralısı (*subsequence*) ise o sıralı tarafından desteklenir denilir. Sık sıralı örüntüler ise market-sepetindeki belirtilen çerçevede anlatıldığı ile büyük oranda benzerdir. Kavram olarak benzer olmakla beraber, problemin doğasından kaynaklanan farklılık geliştirilen algoritmaları da farklı kılmıştır.

Sık sıralıların bulunmasına yönelik ilk algoritma [6] çalışmasında verilmiştir. Bu çalışma Apriori algoritmasının sıralılar için uyarlaması olarak düşünülebilir. Daha sonra literatürde etkin veri yapıları kullanılarak performansı arttırmaya yönelik yeni algoritmalar, örn [7], geliştirilmiştir. Ayrıca, sonuç kümesinde çoğunlukla kullanışsız sıralıların olmasını azaltmak ve dolaylı olarak performansı artırmaya yönelik olarak verilen kısıtlar altında sıralı örüntüleri arayan algoritmalar, örn [8], geliştirilmiştir.

Zaman-mekan izleri veritabanları da adreslenmiş ve bunlar için de algoritmalar gerçekleştirilmiştir. Zaman-mekan izleri, sıralıların özel bir durumu olmakla birlikte onlardan bir çok yönden farklıdır. En büyük fark, konum ve zamanın çoğu uygulamada kesinlik değerinin düşük olması ve büyük oranda belirsizlik (*uncertainty*) içermesidir [9]. Dolayısıyla sık örüntüler eşitlik değil benzerlik temelinde aranır. Bu veritabanlarında, belirsizlikten dolayı, problem tanımı da, diğerlerinden farklı olarak, standart değildir.
[10] çalışmasında yapısal veritabanları için özel birliktelik analizi algoritmalarına ihtiyaç olduğu vurgulanmış ve çizge veritabanları için sık altçizge bulma problemi tanıtılmış ve tüm sık altçizgeleri bulan bir algoritma verilmiştir. Buradaki çerçeve, market-sepeti analizindeki ile aynıdır; yani

algoritma Apriori tabanlıdır. Algoritma hem yönlü hem yönsüz çizgeler için çalışabilmektedir. Yaklaşım olarak benzemekle beraber, teorik açıdan problemin doğası market-sepetinden oldukça farklıdır. En önemli fark ise, altçizgelerin destek değerlerini sayarken karşılaşılan altçizge şekil benzerliği (*subgraph isomorphism*) probleminin NP-Hard olmasıdır. [11] çalışmasında ise üç boyutlu çizgelerden örüntü çıkarma problemi üzerinde çalışılmıştır.

3.GÜVENLİKLİ VERİ MADENCİLİĞİ

Veri madenciliğinin veritabanı güvenliği için bir tehdit oluşturduğu fikir olarak ilk kez O'Leary [13] tarafından belirtilmiş ve daha çok istatistiksel veritabanları kapsamında araştırılmıştır. Temelde güvenlik açıkları iki grupta incelenmektedir [21;14]; (i) ham veri yayınlama (ii) VM sonuçları yayınlama. Terminolojiden de anlaşılacağı üzere birinci durumda verinin kendisi, ikinci durumda ise veriden elde edilen bilgi ve örüntüler yayınlanır. İkinci durum bu çalışma kapsamı dışındadır.

Hem ham veri yayınlama hem de VM sonuçları yayınlamada farklı nitelikte güvenlik açığı oluşabilmektedir. Bunlardan en bilineni (mikro veritabanlarında) anonimliktir; yayınlanan veriden, ya da VM sonucundan gerçek kişilere ait hassas bilgilere ulaşılabilmesi. Bu durumun en kanonik çözümü k-anonimlik (*k-anonymity*) [15] dir; herkes diğer en az *k-1* kişiden ayırt edilemez. Böylelikle kişilerin mahremiyeti korunmuş olur. Veri yayınlama için problemin optimal çözümünün NP-Hard olduğu hareketsel veritabanları [16] ve zaman-mekan izleri veritabanları [22] için gösterilmiştir. Dolayısıyla geliştirilen algoritmalar [17;22] ya sub-optimal olmakta ya da bir sezgisel kullanmaktadır.

Veritabanında ham veriler içerisinde sakıncalı veriler varsa fakat veritabanından VM sonucu çıkacak bilgiler açısından bir sakınca yoksa, bu durumda sakıncalı veriler silinerek veritabanı yayınlanır. Bu durumda ferdi kayıtlarda yapılan değişiklikler, geçerli VM modelleri elde edilebilmesi açısından, yayınlanan veritabanından bütüncül seviyede bir değişikliğe yol açmamalıdır. Buna örnek olarak, maaş bilgilerinin ortalama ve standart sapmaları korunarak değiştirilip yayınlanması verilebilir. Bu işlem literatürde veri sersemletme olarak anılır ve bu konudaki ilk çalışma [12] tarafından yapılmıştır.

Diğer bir önemli güvenlik açığı yayımlayıcının istemediği (ya da bilinmesinin mahsurlu gördüğü) bilgilerin ortaya çıkarılmasıdır. Buna örnek olarak belirli bir bankanın müşteri profili ile ilgili kurallar verilebilir. Diğer bir örnek ise piyasaya yeni çıkmış bir ürünün tutulup tutulmadığının market kayıtlarından anlaşılmaya çalışılmasıdır. Bu gibi durumlarda sakınca addedilen bilgi veritabanını alan kişi tarafından çıkarsanamamalıdır. Dolayısıyla, veritabanı bu sakıncalar temizlendikten sonra yayınlanmalıdır. Bu konu literatürde bilgi gizleme olarak bilinmektedir.

4. BİLGİ GİZLEME

Yayınlanan veri içerisinde, yayınlayıcının bilinmesini istemediği bazı istenmeyen ilişkiler ve örüntüler olabilir. Bilgi gizleme ile hedeflenen bu istenmeyen ilişkiler ve örüntülerin yok edildikten sonra verinin yayınlanmasıdır. Veriyi alan kişinin bu veri üzerinde VM yapsa bile gizlenmiş örüntüleri elde edememesi için yapılan veri dönüştürme işlemine *veri temizleme* denilir. Temizlenmiş veri kümesinin, istenmeyen örüntülerin VM ile çıkarsanamamasının garantilenmesinin yanı sıra orijinal veri kümesine de maksimum benzerlik göstermesi hedeflenir. Problemin NP-Hard sınıfında olduğu market sepeti veritabanları [18], sıralılar veritabanları [19] ve zaman-mekan izleri veritabanları [20] için gösterilmiştir.

Bu konudaki çalışmaların çoğunluğu market-sepeti verilerinden birliktelik kuralları gizleme VM modeli için yapılmıştır [23;8]. Sık parçamal kümesi gizleme probleminde kullanıcı tarafından verilen hassas parçamal kümesi listesinin destek değerleri yine kullanıcının belirlediği bir açığa vurma eşik değerinin altında olacak şekilde düşürülür. Düşürme işlemi bazı hareketlerden parçamal kümesinin içinde bulunan bazı parçamalların silinmesi şeklinde yapılır. Dikkat edilirse veritabanını alan kişi öngörülen açığa vurma eşik değeri ile VM yaparsa gizlenmiş kuralları ortaya çıkartamaz. Veritabanı kullanışlılığını maksimize etmek için veritabanında gereğinden fazla parçamal bastırılmamalıdır, aksi halde tamamen kullanışsız bir veritabanı yayınlanmış olur. Önerilen algoritmalar çeşitli stratejilerle veritabanı utilitesini maksimize etmeye çalışırlar.

Atallah et al. [18] çalışmasında, hassas olarak verilen kuralların destek değerleri, diğer hassas olmayanların destek değeri olabildiğince az değişecek biçimde, bir açgözlü algoritma ile azaltılır. Bunun için öncelikle çok sayıda hassas parçamal kümesini destekleyen hareketler seçilir. Bu hareketlerden öyle bir parçamal seçilir ve silinir ki hassas tüm parçamallarca içerilir ve bu parçamalın etkisi diğer 2-sık parçamal kümelerini en az etkiler. Bu konudaki diğer bazı yaklaşımlar [23;24;8] de bulunabilir.

4.1. Sıralılar Gizleme

Bilgi gizlemenin market-sepeti tipi veritabanları dışına taşınması Abul et al. [19] ile başlamıştır. Bu çalışmada hassas kurallar sıralılar olarak tanımlanır ve sıralılar veritabanlarındaki bu hassas sıralıların destek değeri açığa vurma eşik değerinin altına

çekilerek azaltılır. Bunun için kaynak veritabanından öncelikle temizliği yapılacak sıralılar bir sezgisel yardımı ile seçilir ve seçilen sıralılar başka bir sezgisel yardımı ile temizlenir. Bu çalışmada ayrıca, hassas sıralılar üzerinde kimi kullanıcı tanımlı kısıtların tanımlanmasına da izin verilmiştir. Bu kısıtlar minimum aralık, maksimum aralık ve maksimum pencere dir. Kısıtların amacı, gereğinden fazla temizlikten kaçınmak ve veri utilitesini artırmaktır.

Problem 1 (*Sıralılar Gizleme*): $\mathbf{S_h}=\{S_1, S_2, \ldots, S_n\}$, burada $S_i \in \Sigma^*$ (i=1..n) ve Σ bir alfabe, hassas sıralılar kümesi, D sıralılar veritabanından, ψ açığa vurma eşik değeri parametresi ile gizlenmek istensin. Sıralılar gizleme, D veritabanından D' veritabanına dönüştürme işidir, şöyle ki;

1) $sup_{D'}(S_i) \leq \psi$, (i=1..n), ve
2) $\Sigma_{Si \in Sh} |sup_{D'}(S_i) - sup_D(S_i)|$ en küçüktür. □

Dikkat edilirse Problem 1, açığa vurma parametresini de problem tanımına eklemiştir. Dolayısıyla D' üzerinde veri madenciliği uygulayan bir kişi ψ destek değeri altında madencilik yaptığında hiçbir hassas sıralıyı keşfedemez. ψ=0 özel durumda hassas sıralılar tamamen veritabanından silinir.

Dikkat edilecek olursa, probleminin ikinci kısıtı optimizasyon problemidir. Gerçekte bu kısıt problemin zor olmasına sebep olur. Aksi halde sadece birinci kısıtı sağlayan çok basit çözümler rahatlıkla bulunabilir ($D' = \varnothing$ en basit çözümdür).
Problemin tanımında belirtilmese de çözüm kümesinin, D', diğer bazı özellikleri de taşıması istenebilir. Bu özelliklerden en önemlisi sahte örüntü katmamasıdır. Yani, D veritabanından çıkarsanamayacak hiçbir örüntü D' veritabanından da çıkarsanamamalıdır. Kolayca anlaşılacağı üzere, aksi halde gerçek dünyada olmayan bir örüntü tamamen yanlış kararlara sebebiyet verebilir. Diğer bir önemli özellik ise sahte çoklu eklenmemesidir. Öte yandan, bu özelliğin ihlali sahte örüntü katma kadar önemli değildir. Diğer bir önemli nokta ise, veritabanının temizlendiğinin saldırganlar tarafından anlaşılamamasıdır. Yani, yayınlanan veritabanının temizlenmiş bir veritabanı olduğu izlenimi oluşturulmamalıdır. Aksi halde, bu anlaşılırsa bu durum saldırgan tarafından karşı saldırı olarak kullanılabilir.
Problem 1, daha da basitleştirilerek tek bir sıralının (diğer bir deyişle $|D|$=1) temizlenmesi durumu için sorulabilir.

Problem 2 (*Tek bir sıralı gizleme*): $\mathbf{S_h}=\{S_1, S_2, \ldots, S_n\}$, burada $S_i \in \Sigma^*$ (i=1..n) ve Σ bir alfabe, hassas sıralılar kümesi, T bir sıralı olsun. Sıralı gizleme, T sıralısını T' sıralısına bazı elemanların silinerek dönüştürme işidir, şöyle ki;

1) $sup_{T'}(S_i) = 0$, (i=1..n), ve
2) T' de silinen eleman sayısı en azdır. □

Teorem 1: Problem 2, NP-Hard dır. (İspatı için [19]). □
Problemin çözümü için global seviyede ve lokal seviye de sezgiseller kullanan algoritma önerilmiştir [19]. Global seviyede, D den kardinalitesi $|D| - \psi$ olan bir alt küme seçilir ve lokal olarak ise bu kümedeki her bir sıralı birbirinden bağımsız olarak temizlenir. Her iki sezgisel de hassas sıralı kümesinin sıralılarda kaç farklı şekilde eşlendiğini hesaplar ve karar buna göre verilir. Bu iki sezgiselin etkin olduğu deneysel olarak gösterilmiştir. Bahsedilen çalışmada ayrıca, hassas sıralıların, veritabanı sıralılarına eşlenmesini engelleyebilecek kısıtlar da konmuştur. Bu kısıtlar, minimum aralık, maksimum aralık ve maksimum penceredir. Böylelikle temizlik işlemi sonrası veri utilitesi artırılmış olur. Yöntemle ilgili detaylı bilgiler bahsedilen çalışmada bulunabilir.

4.2. Zaman-Mekan Bilgileri Gizleme

Zaman-mekan izleri veritabanları için bilgi gizleme problemi Abul et al. [20] tarafından adreslenmiştir. Zaman-mekan izlerinin, sıralılardan farkı boyut sayısının artmasıdır, yani her nokta bir zaman birde mekanla ayırt edilir. Bu problem için de sezgisel çözümler üretilmiştir. Zaman-mekan izlerinin bir diğer zorluğu da cahil temizleme yaklaşımının çalışmamasıdır. Çünkü, zaman-mekan bilgileri bir harita ile karşılaştırılıp temizlenmiş noktaların muhtemel olabileceği yerler kolayca belirlenebilir. Bu durumda yapılan temizlik ancak sözde bir temizliktir. Bu özellik fark edilmiş ve istenmesi halinde konum haritası dikkate alınarak gerçek bir temizlik algoritması aynı çalışmada verilmiştir. Böylelikle gizlenen noktaların saldırgan tarafından tekrar oluşturulması mümkün olmamaktadır. Buradan da görüldüğü üzere, bilgi gizleme farklı çerçevelerde kendine has problemler taşımaktadır. Bu çalışmanın en önemli özelliklerinden birisi de, konum haritası nedeniyle, bilgi gizleme ve k-anonimlik arasında bir ilinti kurmuş olmasıdır.
Sıralılar gizleme problemi ile kıyaslandığında zaman-mekan bilgileri gizleme daha karmaşıktır. Burada gerek problemin zorluğu gerekse uygulama alanlarının genişliği (cep telefonları ile kaydedilen kişilerin günlük hareketleri, araçların GPS ile konum kayıtları vb.) dolayısıyla daha detaylıdır. Günümüzde mobil e-uygulamaların gerek nicelik gerekse nitelik olarak artmasından dolayı problemin farklı uygulamaları için (örn, konum-tabanlı hizmet) farklı formülasyonlar ve dolayısıyla farklı çözümler ve çözümden beklenen başarım kriterleri mevcuttur.

5. SONUÇLAR

Bilgi gizleme problemi daha çok market-sepeti tipi veritabanları için çalışılmıştır. Günümüzde yeni uygulamalar çeşitli tipten veriler gerektirmektedir. Buna paralel olarak çeşitli tip, sıralılar ve mekan-zaman izleri, veritabanları için bilgi gizleme çalışmaları literatürde yeni başlamıştır. Bu konulardaki mevcut çalışmalar literatürün olgunluğa erişmesi için farklı açılardan (örn, ölçeklenebilir algoritmalar, daha etkin ve hızlı algoritmalar) geliştirilmelidir. Ayrıca problem için uygun algoritma geliştirme yanında alternatif problem formülasyonları ve problemlerin teorik özelliklerinin gösterilmesi de önem kazanmaktadır.

Diğer veri tipleri olan çizge, üçboyutlu çizge ve zaman serileri veritabanları için henüz literatürde bir çalışma yoktur. Fakat, yukarıda da bahsedildiği üzere, sıralılar ve mekan-zaman izleri veritabanlarında takip edilen yöntem bu veritabanları için de uygulanabilir. Çizge tipi yapısal veritabanları çoğunlukla bilimsel veritabanları için kullanılmakta fakat başka potansiyel kullanım alanları da mevcuttur. Buna en güncel örnek *facebook* benzeri sosyal ağlar verilebilir. Dolayısıyla, bu veritabanlarından bilgi gizleme probleminin de ilginç uygulama alanları bulması beklenmektedir. Zaman serileri veritabanları, temelde sıralılar veritabanlarına benzemekle beraber onlardan farklı açılardan ayrılır. Dolayısıyla, bu veritabanları için de doğrudan metodlar adreslenmelidir.

Hassas bir bilginin hassas olmayan bir bilgi ile büyük oranda korele olduğu durumlar olabilir. Bu durumda hassas bilgi gizlenmiş bile olsa, ilgili korelasyonu bilen kişi hassas olmayan bilgiden hassas bilgiyi kolayca türetebilir. Dolayısıyla, her türlü sözde bilgi gizleme durumundan kaçınılmalıdır.

KAYNAKLAR

[1] Agrawal, R., Imielinski, T. Swami, A. *Mining association rules between sets of items in large databases*, SIGMOD'93, 1993.

[2] Agrawal, R., Srikant, R. *Fast Algorithms for Mining Association Rules*, VLDB'94, 1994.

[3] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W. *New Algorithms for Fast Discovery of Association Rules*, KDD'97, 1997.

[4] Savasere, A., Omiecinski, E., Navathe, S. *An Efficient Algorithm for Mining Association Rules in Large Databases*, VLDB'95, 1995.

[5] Han, J., Pei, J., Yin, Y., *Mining frequent patterns without candidate generation*, SIGMOD'00, 2000.

[6] Agrawal, R., Srikant, R. *Mining Sequential Patterns*, ICDE'95, 1995.

[7] Zaki, J.M., *SPADE: An Efficient Algorithm for Mining Frequent Sequences*, Machine Learning, Vol. 42(1/2), 2001.

[8] Oliveira, S.R.M., Zainae, O.R. *Protecting sensitive knowledge by data sanitization*, ICDM'03, 2003.

[9] Yang, J., Hu, M. *TrajPattern: Mining Sequential Patterns from Imprecise Trajectories of Mobile Objects*, Springer LNCS, Vol. 3896, 2006.

[10] Inokuchi, A., Washio, T., Motoda, H. *Complete Mining of Frequent Patterns from Graphs*, Machine Learning, Vol. 50, 2002.

[11] Wang, X., Wang, J.T.L., Shasha, D., Shapiro, B.A., Rigoutsos, I., Zhang, K. *Finding Patterns in Three-Dimensional Graphs: Algorithms and Applications to Scientific Data Mining*, IEEE TKDE, Vol 14(4), 2002.

[12] Agrawal, R., Srikant, R. *Privacy-preserving data mining*, SIGMOD'00, 2000.

[13] O'Leary, D. E. *Knowledge discovery as a threat to database security*, In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases, pages 507-516. AAAI/MIT Press, 1991.

[14] Bonchi, F., Saygin, Y., Verykios, V.S., Atzori, M., Gkoulalas-Divanis, S. V., Kaya, E. *Privacy in Spatio-temporal Data Mining*, In "Mobility, Data Mining, and Privacy", F. Giannotti and D. Pedreschi Eds., Springer, 2008.

[15] Samarati, P, Sweeney, L. *Generalizing data to provide anonymity when disclosing information*, PODS'98, 1998.

[16] Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., Zhu, A. *Anonymizing Tables*, ICDT'05, 2005.

[17] LeFevre, K., DeWitt, D.J., Ramakrishnan, R. *Mondrian: multidimensional k-anonymity*, ICDE'06, 2006.

[18] Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., Verykios, V. S. *Disclosure limitation of sensitive rules*, KDEX'99, 1999.

[19] Abul, O., Atzori, M., Bonchi, F., Giannotti, F. *Hiding Sequences*, PDM workshop, ICDE'07, 2007.

[20] Abul, O., Atzori, M., Bonchi, F., Giannotti, F. *Hiding Sensitive Trajectory Patterns*, PADM workshop, ICDM'07, 2007.

[21] Abul, O. *E-Sağlık ve Mahremiyete Veri Madenciliği Kaynaklı Tehdit*, 2. Ulusal E-Sağlık Kongresi-Teknik program, 2007, Antalya.

[22] Abul, O., Bonchi, F., Nanni, M. *Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases,* ICDE'08, 2008.

[23] Dasseni, E., Verykios, V.S., Elmagarmid, A.K., Bertino, E. *Hiding association rules by using confidence and support*, 4th International Workshop on Information Hiding, 2001.

[24] Saygin, Y., Verykios, V.S., Clifton, C. *Using unknowns to prevent discovery of association rules*, SIGMOD Record, Vol. 30(4), 2001.