**Imputation (Töhmet)**

Bu yazının amacı, bilgisayar bilimlerinde özellikle veri madenciliği (data mining) konularında eksik verilerle karşılaşılması halinde bir çözüm olarak bu eksik verilerin töhmet edilmesi (yerine uygun verilerin üretilmesi, imputation) yöntemini açıklamaktır.

Töhmet, sözlükte olmayan bir şeyin yüklenmesi anlamındadır. Örneğin olmayan bir suçun birisine yüklenmesine töhmet altında bırakmak denilebilir. Bu anlamda bir veri kümesi (data set) üzerinde çalışılırken bazı sebeplerden dolayı verilerin eksik olması halinde bu verilerin uygun başka sayılarla tamamlanması sağlanabilir.

Genelde verinin hatalı okunması, veri kaynağında yaşanan bozulma gibi sorunlar veya bazı verilere erişim zorluğu eksik verilere sebep olabilmektedir.

Bu verilerin eksik olması durumu genelde sorunlara sebep olur. Örneğin çoğu hazır istatistik paketleri (SAS, SPSS, Weka veya r-project gibi) bu tip durumlarda sorunlar yaşamaktadır. Gerçi bu paketlerin ücretli ve gelişmiş olanlarının çoğunda (SAS veya SPSS gibi) töhmet modülleri (imputation) bulunmakta ve bu işi otomatik olarak yapabilmektedirler ancak bu yazı kapsamında gerek bu modüllerin nasıl çalıştığını anlamaya çalışacağız gerekse bu işlemi elle yapmak istediğimizde nasıl müdahale etmemiz gerektiğini açıklamaya çalışacağız.

Töhmet yöntemleri duruma ve beklentilerimize göre çeşitlilik arz eder ve halen üzerinde çalışılmaktadır. Bunlardan çok bilinen bazılarını saymamız gerekirse:

* Sıcak deste (hot deck)
* Soğuk deste (cold deck)
* Liste boyunca silme (listwise deletion)
* Eşlerin silinmesi (pairwise deletion)
* Ortalama töhmet (mean imputation)
* İlkelleme töhmedi (regression imputation)
* Son gözlemin taşınması (last observation carried forward)
* Olasılıksal töhmet (stochastic imputation)
* Çoklu Töhmet (Multiple imputation)

Yukarıdaki bu kavramları kısaca açıklamaya çalışalım. Öncelikle sıcak deste ve soğuk deste (hot deck , cold deck) algoritmaları için ne yazık ki tam bir mutabakat olmadığını ve çeşitli versiyonları bulunduğunu belirtmek isterim ( bkz. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3130338/>) Bu yüzden bu iki algoritmayı atlayacağım ancak hot deck (sıcak deste) için kısaca rast gele bir verinin seçilerek eksik olan veri yerine töhmet edildiğini söylememiz yeterli olacaktır. Bu rast gele seçim ise oldukça tartışmalı bir konu. ve diğer metotları kısaca açıklayacağım.

**Liste boyunca silme (listwise deletion)**

Örneğin bir veri tabanından okunan tabloda bazı değerlerin eksik olduğunu düşünelim:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | —— |
| —- | — | Muhasebe |
| Veli Demir | 43 | Yönetim |

Yukarıdaki tabloda sondan ikinci ve üçüncü satırlarda kayıp bilgi bulunmaktadır. Bu durumda, örneğin çalışanların yaş ortalamalarının alınması istendiğinde hata ile karşılaşılacaktır.

Liste boyunca silme yaklaşımında (listwise deletion) bu kayıp veri içeren satırların tamamı tablodan temizlenir ve tablo aşağıdaki hale getirilerek işlem sonuçlandırılır.

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Veli Demir | 43 | Yönetim |

Artık bu tablo üzerinde istenilen işlemler yapılabilir.

**Eşlerin silinmesi (Pairwise deletion)**

Bu yöntemde bütün tablonun temizlenmesi yerine gerekli olan işlem sırasındaki eksik veriler temizlenir. Örneğin bir önceki tabloya geri dönecek olursak:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | —— |
| —- | — | Muhasebe |
| Veli Demir | 43 | Yönetim |

Ve bu tablo üzerinde çalışanların yaşının ortalaması istenmişti. Bu durumda, bu soruya özel olarak sadece sondan ikinci satırın silinmesi yeterlidir:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | —— |
| Veli Demir | 43 | Yönetim |

Artık istenen işlem yani yaşların ortalaması çalıştırılabilir.

Bu yöntemde farklı bir işlem yapılmak istendiğinde, örneğin çalışanların kısımlara göre dağılım grafiği istendiğinde ise işlem yapacağımız tablo aşağıdaki şekilde olacaktır:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| —- | — | Muhasebe |
| Veli Demir | 43 | Yönetim |

Görüldüğü üzere sadece o işlem için problem çıkaran satır silinmiş diğer satırlar farklı kolonlarında eksik veri bulunmasına rağmen saklanmıştır.

Bu yöntemin menfi yanı, her işlem için farklı veri kümesi üretiliyor olması ve hatta her işlem sonucunda farklı sayıda veri ele alınıyor olmasıdır. İşlemler sonucunda verilerin karşılaştırılma güçlüğü ortaya çıkabilir.

**Ortalama Töhmet (Mean imputation)**

Bu yöntemde, bir eksik verinin töhmeti sırasında ortalama değer hesaplanır. Örneğin yine aynı tablo üzerinden töhmet yaklaşımını izah edelim:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | —— |
| —- | — | Muhasebe |
| Veli Demir | 43 | Yönetim |

Şayet bu tablodaki ortalama yaş için eksik olan satırların töhmeti söz konusuysa bu durumda satırların silinmesi yerine veri üretilir. Örneğimizdeki yaşların ortalaması:

http://www.bilgisayarkavramlari.com/wp-content/uploads/sbres_1359095036_0__.gifolarak bulunur. Yaşın ondalıklı sayı olması mümkün olmadığı için 32 olarak yuvarlanarak kabul edilebilir ve veri kümemiz aşağıdaki şeklini alır:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | —— |
| —- | 32 | Muhasebe |
| Veli Demir | 43 | Yönetim |

Bu işlemin ardından artık istenen veri madenciliği yöntemleri uygulanabilir.

Bu yöntemin dez avantajı ise, çok büyük veriler de uygulanma zorluğudur (örneğin güncel problemlerin artık [haritalama-indirgeme (map reduce)](http://www.bilgisayarkavramlari.com/2013/01/18/map-reduce-haritalama-indirgeme/) ortamlarında işlendiği düşünülürse, böyle bir ortamda kullanılamaz). Ayrıca işlenmesi için bütün verinin hafızaya yüklenip hesaplama yapılması zorluğu da bulunmaktadır.

Son olarak veri eksikliğinin çok fazla olduğu durumlarda üretilen verilerin sağlığı problem çıkarmaktadır.

**İlkelleme Töhmeti (Regression Imputation)**:

Bu yöntemde, mevcut veriler üzerinden fonksiyonel bir ilkelleme yapılır (örneğin [doğrusal ilkelleme (linear regression)](http://www.bilgisayarkavramlari.com/2013/01/26/linear-regression-dogrusal-ilkelleme/)) ve ardından ilkellenen fonksiyondan üretim yapılarak eksik veri doldurulur.

**Son Gözlemin Taşınması (Last Observation Carried Forward):**

Bu yöntemde, veri kümesindeki eksik veriler, kendilerinden bir önceki veri kopyalanmak marifetiyle töhmet edilirler. Örneğimize geri dönecek olursak:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | —— |
| —- | — | Muhasebe |
| Veli Demir | 43 | Yönetim |

Veri kümesindeki verilerin son gözlemin taşınması yöntemiyle töhmet edilmiş hali aşağıdaki şekildedir:

|  |  |  |
| --- | --- | --- |
| İsim | Yaş | Kısım |
| Şadi Evren ŞEKER | 33 | Bilgisayar |
| Ali Demir | 23 | Muhasebe |
| Cem Yıldız | 26 | Bilgisayar |
| Ahmet Yılmaz | 33 | Bilgisayar |
| Ahmet Yılmaz | 33 | Muhasebe |
| Veli Demir | 43 | Yönetim |

Artık bu veri kümesi üzerinde veri madenciliği yöntemleri uygulanabilir.

Bu yöntemin dez avantajı ise bazı marjinal noktaların çoğaltılması olarak görülebilir. Örneğin Yaş sütununda marjinal olarak 70 yaşında bir çalışan varsa bu kişinin kopyalanması, veri madenciliği sonuçlarını menfi etkileyebilir.

Son iki yöntemimiz olan istatistiksel töhmet (stochastic imputation) ve çoklu töhmet (multiple imputation) için şunları söylememiz yeterli olacaktır.

Öncelikle mevcut veri kümesi üzerinden bir istatistiksel dağılım elde edilir (örneğin ilkelleme (regression) burada kullanılabilir) ardından bu dağılım sayesinde eksik verileri dolduran bir bağlantı kullanılır. Çoklu töhmette ise bu işlem birden fazla kere yapılarak her bir veri kümesi daha sonra kullanılmak üzere saklanır. Ayrıca her veri kümesinin hata miktarı hesaplanır. Ardından bu veri kümelerinin ortalama değerleri alınarak nihai veri kümesi elde edilir. Buradaki tek ayrınta standart hatanın (standard error) hesaplanması sırasında iki veri kümesinin birleştirilmesi için iki standart hata miktarı toplanıp karekökleri alınır.

Kısacası çoklu töhmet, daha iyi sonuç elde etmek için istatistiksel töhmetin birden fazla kere çalıştırılması olarak düşünülebilir.