

Durum Tabanlı Çıkarsama (DTÇ)

Tahir Emre KALAYCI

March 23, 2007

- 1 Gündem
- 2 Giriş
- 3 Durum Tabanlı Çıkarsama
 - Nasıl yaparız
 - Durumların Temsili
 - Eşleme için benzerlik hesaplaması
 - Durum Saklama
 - Durum Çekme
 - Durumların Tekrar Kullanılması
 - Durum Gözden Geçirme
- 4 DTÇ Uygulamaları
 - DTÇ Geliştirme Aşamaları
 - DTÇ Araçları
- 5 Sonuç

Nedir?

- Şu anki bir problemi çözmek için önceden karşılaştığımız problemlerden yararlanıyoruz.
- İnsanın problemi ele almasına benzemektedir.
- Verilen problemle daha önce karşılaşılan problemlerin benzerlik ve farklılıklarından yararlanılır
- DTÇ deneyimle öğrenir.
- YSA öğrenmesinden farkı genelleme yapmamasıdır.
- KTS'lerden farkı olayların kural olmamasıdır.

Tanımlar

- Durum ("Case"): Bir problem durumudur, deneyimler veya öğrenilenler durum tabanında saklanır.
- Bir durum çözülmemiş veya ileride çözülecek bir durum olabilir.
- Hukuk'a benzemektedir. Önceki davalar yeni davalar için emsal olabilmektedir.

Tarihçe

- Roger Schank 1980'lerde başlamıştır.
- DARPA tarafından fon desteği verilmiştir.
- Daha sonraları Avrupa'da başlamıştır. Özellikle Almanya'da çalışmalar yoğunudur.
- 1990: Ticari uygulamalar geliştirmeye başlanmıştır.
- 1998: Dünya üzerinde 35'ten fazla enstitüde araştırma çalışmaları
- Bir çok ticari ürün vardır.
- Günlük kullanımda bir çok uygulaması vardır.

Nasıl

- DTÇ'nin bilgi durum tabanı problemler ve çözümlere yönelik deneyimleri ve durumları içerir.
- Durum tabanındaki her durum özellikleri ve bulunan çözüm terimleriyle tanımlanmaktadır.

Adımlar

- Eldeki probleme eşlenen bir durum getiriyoruz.
- Eşlenen durumun çözümünü istenen çözümü bulmak için uyarlıyoruz.
- Önerilen çözümü deniyoruz ve gözden geçiriyoruz.
- Onaylanan çözümü ileride kullanabilmek için durum tabanına saklıyoruz.

Örnek Bir Problem

- Problem: Arabadaki hatayı bulup bir onarım önerisi sunalım
- Durum durum tabanı aşağıdaki durumları içeriyor:
 - Belirtilerin tanımları
 - Sebeplerin tanımları
 - Onarım stratejilerinin tanımları
- Her durum bir tanıyı tanımlar.
- Durumlar olgunun özelliklerini ve özgün değerlerini kaydeder.
- Her durum diğer durumlardan farklı bir olguyu tanımlar.

Durumların Temsili

- DTÇ durum tabanının içerik ve yapısına bağlıdır.
- Durum arama ve eşlenen durumları çekme işlemlerinin etkin olması gereklidir.
- Gerçek durum temsili alan ve görev gereksinimlerine bağlıdır.
- Ayrıca elde var olan durum verisinin yapısından da etkilenir.

Temsil Yaklaşımları

- Düz özellik-değer listesi
 - Basittir.
 - Bazı alanlardaki problemleri çözmek için yeterlidir.
 - Kolay depolama ve çekme sağlar.
- Nesne yönelimli temsiller
 - Durum yapısı hiyerarşik olarak yansıtılır.
 - Durumlar nesne kümelerinden oluşur.
 - Nesneler özelliklerle tanımlanır.
 - Her nesne bir nesne sınıfına aittir. Nesne sınıfları kalıtımı desteklemektedir.

Temsil Yaklaşımları

- Çizge temsili
 - Düğümlerden oluşmaktadır.
 - Düğümleri birleştiren bağlantılar vardır.
 - Hiyerarşik temsile göre daha esnek ama karmaşıktır.

Eşleme için benzerlik hesaplaması

- Farklı özellikler farklı önem seviyesine sahiptir.
- Çok basit karşılaştırmalar işimizi görmez.
- Özellik değerlerine atanan ağırlık değerleri önemlerini yansıtmaktadır.
 - Kullanıcı deneyimlerine dayanan öncelikler atanabilir.
 - Belli problem örnekleri için varsayılan önem atamaları yapılabilir.
- Özelliklerin sembolik ve nümerik olmasına göre değişmektedir.

Eşleme için benzerlik hesaplaması

- Nümerik değerli özellikler
 - En yakın komşu algoritması ("Nearest-Neighbour Algorithm")
 - Özellikler arttıkça EYKA'nın güvenilirliği azalmaktadır.

Eşleme için benzerlik hesaplaması

- Sembolik değerli özellikler
 - Sezgiye dayalı küme-alt küme ilişkileriyle ölçülebilir.
 - Durumlar özelliklerine bağlı olarak hiyerarşik olarak gösterilebilir.
 - İki sembolik durum arasındaki uzaklık ortak atalarına olan uzaklıkla ölçülebilir.
 - Uzaklık hesaplama sistemi olarak istatistiksel model, kural tabanlı sistem veya YSA biçiminde olabilir.

Durum saklama

- Büyük durum veritabanları için etkin saklama ve çekme gereklidir.
- Saklama yöntemi durum gösterimi ve durum tabanının büyüklüğüne bağlıdır.
 - Doğrusal listeler
 - Dizin yapılar (Ağaçlar, ağlar) (Geniş veritabanları için)
- İçsel ve dışsal olabilir.
- Küçük ve paylaşımsız durum tabanları hafızada saklanabilir.
- Geniş veya paylaşımlı durum tabanları veritabanında saklanabilir.

Durum çekme

- Özelliklerin belirlenmesi
 - Kullanıcı girdisinden problem tanımlayıcıları çıkarılır.
 - Bilinmeyen tanımlayıcılar atılır veya kullanıcıdan açıklanması istenir.
 - Tanımlayıcılar kavramsal genel bilgiye dayanarak ifade edilebilir.
- İlk eşleme
 - Tüm girdi özelliklerine eşlenen durumlar seçim için uygundur.
 - Belli bir oranda eşlenen durumlar da çekilebilir.
 - Benzerlik değerlendirmesi bilgi yoğunluklu olabilir.
- Seçim
 - İlk eşlemede bulunan adaylardan en iyi eşlenen seçilir.
 - Sıkı inceleme ve sıralama içermektedir.
 - Bilgi yoğunluklu seçim yöntemleri açıklamalar da üretir.

Durumların tekrar kullanılması

- Tekrar kullanımın amacı en iyi eşlenen durumu verilen probleme uyarlamadır.
- Olasılıklar
 - Çözümde değişiklik yok: Eşlenen durumun sonucunu aynen kopyalama
 - Kullanıcı etkileşimli elle/otomatik çözüm uyarlama
 - Otomatik çözüm uyarlama
- Otomatik çözüm uyarlama
 - Geçmiş durum çözümünü kullanarak-Dönüşümsel benzerlik ("Transformational Analogy")
 - Geçmiş durumu yapan yöntemi kullanarak-Türetimsel benzerlik ("Derivational Analogy")

Durumların tekrar kullanılması

- Dönüşümsel benzerlikte kural veya işleçler geçmiş durum çözümünü iki durum arasındaki farklılıklara göre ayarlar.
- Türetimsel benzerlikte geçmiş durum çözme için kullanılan yöntem hakkında bilgi içerir. Bu bilgi yeni duruma uygulanır.

Durum gözden geçirme

- İki görevden oluşmaktadır
 - 1 Tekrar kullanımla üretilen çözümün değerlendirilmesi, değerlendirmeden geçemezse
 - 2 Alana özgü bilgiyi kullanarak çözümü onarma
- Değerlendirme
 - Çözümü gerçek ortama uygulayarak doğruluğunu ve kalitesini onaylama ve kullanıcının kabulünü alma
 - Bilgisayar benzetimi (Uygulamadan önce deneme)
- Onarma: Çözümdeki hataları belirleme ve çözümü değiştirerek hataların olmasını önleme
- Durum saklama: Sınanan, onaylanan ve (gerekliyse) gözden geçirilen çözümler doğru olarak kabul edilerek durum tabanına saklanır.

DTÇ Uygulamaları

- Çoğu karar destek sistemleri için durum çekmeye odaklanmıştır.
- Durum çekme durum uyarlama aşamasından kaçınmaktadır.
- Destek kararları en çok benzeyen öncekilere dayanmaktadır.
- Örnekler[1]
 - Compaq Computers' SMART System
 - NEC's SQUAD
 - Nippon Steel and Lockheed
- Uygulama alanları
 - Teknik hata tanımlama
 - Sınıflandırma ve tahminleme
 - Kontrol ve izleme
 - Otomatik Planlama ve zamanlama
 - Banka borç analizi

DTÇ geliştirme aşamaları

- Durum tabanının tasarlanması
 - Kaynak malzemeler (yazılı belgeler, uzman hesapları, veritabanı kayıtları) kullanılarak durumların genel bir gösterimi geliştirilir
 - Kullanıcı, yönetici ve sistem geliştiricileri arasında koordinasyon gerekmektedir.
 - Görevler
 - Sorun özelliklerini tanımlamak için kullanılan terimlerin sözlük derlemesi
 - Durumları kataloglamak için uygun özelliklerin seçilmesi
 - Durumları saklamak için kullanılacak veritabanı şemalarının tanımlanması
 - Durum tabanlı yazım standartlarının tanımlanması

DTÇ geliştirme aşamaları

- İlk durum tabanı Geliştirme
 - Bir durum tabanı temel olarak geliştirilir.
 - Bu durum tabanı durum uzayının yeterli bir kısmını kapsayana kadar kullanıcılar tarafından gözden geçirilir ve iyileştirilir.
- Bakım ve devam eden geliştirme
 - Kullanım sırasında durum tabanının gözden geçirilmesine ve iyileştirilmesine devam edilir.
 - Durum doğruluk ve faydası sürekli izlenir
 - Durum tabanı örgütsel veritabanı gibi yönetilir

DTÇ araçları

- Uzman sistem kabukları gibi ticari ürünler vardır.
- Uygulamaların hızlı geliştirilmesini sağlarlar.
- Tipik bir DTÇ geliştirme ortamı aşağıdakileri sunmalıdır:
 - Durum temsili için varsayılan veritabanı şemaları
 - Durum tabanlı karar destek için çözüm aracı
 - Durumları, özelliklerini ve çözümleri düzenlemek için formlar
 - Araçlar
 - Durumları el veya otomatik olarak dizinleme
 - İlişkili veritabanı tablolarından otomatik kayıt içe aktarma
 - Çözümleme (analiz) için kavramsal kümeleme

DTÇ araçları

- Örnekler

- ReMind (Cognitive Systems Inc.)
- CBR Express (Inference Corporation)
- Esteem (Esteem Software Inc.)
- CasePower (Inductive Solutions Inc.)
- ReCall (Isoft)[2]

Avantajlar

- Modellenmesi zor problemleri çözer. (Çoğu alanda problemler modellenmesi zor problemlerdir.)
- Bilgi elde etmede daha az çaba harcanmaktadır.
- Bakıma daha az çaba harcanmaktadır.
- Yüksek ölçeklenebilirlik ve esneklik
- Hatalar kolayca düzeltilebilmektedir.
- İyileştirme sayesinde başarımlar zamanla artmaktadır.
- Ortamdaki değişiklikler durumun silinmesi/eklenmesi ile rahatça yansıtılabilmektedir.

Zayıflıklar

- Hafızaya dayalı geçici kanıtlar kullanmaktadır.
- İstatistiksel geçerli veri olmadan yapılan genellemelerin doğru olma garantisi yoktur.
- Bazı durumlarda insan etkileşimine ihtiyaç duyulmaktadır.
- Durum tabanı büyüdükçe yanıtlama zamanı artabilmektedir.

Kaynaklar

- <http://www.it.murdoch.edu.au/units/ICT619/>
- <http://www.ai-cbr.org/>
- <http://www.aiai.ed.ac.uk/links/cbr.html>
- Java İle Yapay Zeka Mekanizmasına Sahip Bir Ağ Yönetim Sistemi Geliştirilmesi, Erkan Binici, Ege Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 2006



<http://www.ai-cbr.org/applied.html>



<http://www.ai-cbr.org/tools.html>