

Bulanık Mantık Park Yönlendirme Sistemi

Trafik yoğunluğu, günün saati, hava koşulları, boşluk oranı ve kullanıcı türü gibi çoklu girdi faktörlerine dayalı olarak en uygun park alanlarını önermek ve bekleme sürelerini tahmin etmek için Python tabanlı bir bulanık mantık kontrol sistemi.

Projeye Genel Bakış

Bu park yönlendirme sistemi, kesin olmayan bilgileri ele almak ve dilsel değişkenlere dayalı insan benzeri kararlar vermek için bulanık mantık kullanır. Hangi park alanının kullanılacağı (A alanından E alanına kadar) konusunda tavsiyelerde bulunur ve mevcut koşullara göre bekleme süresini (çok kısıdan çok uzuna kadar) tahmin eder.

Özellikler

- 5 girdi değişkenini işler: trafik yoğunluğu, günün saati, hava durumu, boşluk oranı ve kullanıcı tipi
- 2 çıktı önerisi üretir: park alanı ve bekleme süresi
- Giriş için kaydırıcılar, radyo düğmeleri ve açılır menüler içeren etkileşimli GUI
- Bulanık üyelik fonksiyonlarının gerçek zamanlı görselleştirilmesi
- Geçerli zaman senkronizasyonu
- Ayrıntılı sonuç ekranı

Sistem Mimarisi

Girdi Değişkenleri

1. **Traffik Yoğunluk (%0-100)**
 - Dilsel değerler: Düşük, Orta, Yüksek
2. **Günün Saati (0-24 saat)**
 - Dilsel değerler: Sabah Erken, Sabah, Öğlen, Öğleden Sonra, Akşam, Gece
3. **Hava Durumu (0-10)**
 - Dilsel değerler: Açık, Hafif Yağmur, Şiddetli Yağmur, Kar
4. **Boşluk Oranı (%0-100)**
 - Dilsel değerler: Çok Düşük, Düşük, Orta, Yüksek, Çok Yüksek
5. **Kullanıcı Tipi (1-5)**
 - Dilsel değerler: Normal, Üye, VIP, Engelli, Personel

Çıktı Değişkenleri

1. **Önerilen Park Alanı (1-5)**
 - Dilsel değerler: Alan A (en yakın), Alan B, Alan C, Alan D, Alan E (en uzak)
2. **Tahmini Bekleme Süresi (0-30 dakika)**
 - Dilsel değerler: Çok Kısa, Kısa, Orta, Uzun, Çok Uzun

Bulanık Kurallar

Sistem, girdilere dayalı olarak çıktıları belirlemek için 32 bulanık kural kullanır. Bazı örnekler şunlardır:

- Boşluk oranı çok yüksekse, A Bölgesini tavsiye edin
- Trafik yoğunluğu yüksek ve boşluk oranı düşükse, bekleme süresi çok uzundur
- Hava karlıysa ve kullanıcı devre dışı bırakılmışsa, Alan A'yı tavsiye edin

Kurulum ve Ayarlama

Ön Koşullar

- Python 3.6 veya üstü

Bağımlılıklar

Proje aşağıdaki Python paketlerini gerektirir: - numpy - scikit-fuzzy - matplotlib - tkinter (genellikle Python ile birlikte gelir)

Kurulum Adımları

1. Depoyu klonlayın:

```
git clone https://github.com/needforcoding/fuzzy-parking.git
```

```
cd fuzzy- parking
```
2. Gerekli paketleri yükleyin:

```
pip install -r requirements.txt
```

Nasıl Koşulur

Uygulamayı şu şekilde çalıştırın:

```
python main.py
```

Bu, yapabileceğiniz GUI uygulamasını açacaktır:

1. Sürgüleri, radyo düğmelerini ve açılır menüleri kullanarak giriş parametrelerini ayarlayın
2. Park önerilerini almak için "Öneriyi Hesapla" düğmesine tıklayın
3. Önerilen park alanını ve tahmini bekleme süresini görüntüleyin
4. Görselleştirme sekmeleri aracılığıyla bulanık üyelik fonksiyonlarını keşfedin

Kullanıcı Arayüzü Kılavuzu

Giriş Kontrolleri

- **Traffik Yoğunluğu:** Mevcut trafik sıkışıklığını ayarlamak için kaydırıcıyı kullanın (%0- %100)

- **Günün Saati:** Günün saatini ayarlayın (0-24) veya geçerli saati kullanmak için "Geçerli" seçeneğine tıklayın
- **Hava Durumu:** Kaydırıcıyı kullanın ya da açılır menüden bir hava durumu seçin
- **Boşluk Oranı:** Kullanılabilir park yerlerinin yüzdesini ayarlayın (%0-100)
- **Kullanıcı Türü:** Kullanıcı türünü Normal, Üye, VIP, Engelli veya Personel arasından seçin

Tavsiye Çıktısı

- Sistem önerilen park alanını gösterecektir (A'dan E'ye)
- Sistem tahmini bekleme süresini dakika cinsinden gösterecektir
- Her iki öneri için sayısal değerler de verilmiştir

Görselleştirme

- Tüm giriş ve çıkış değişkenleri için üyelik fonksiyonlarını görüntüleme
- İlgili bulanık kümelerini görmek için açılır menülerden farklı değişkenler seçin

Sistem Mantığı

Sistem, öneriler oluşturmak için aşağıdaki süreci kullanır:

1. Giriş değerleri üyelik fonksiyonları kullanılarak bulanıklaştırılır
2. Bulanık kurallar, aktivasyon derecesini belirlemek için değerlendirilir
3. Etkinleştirilen kurallar toplanır
4. Keskin çıktı değerleri üretmek için bulanıklaştırma için centroid yöntemi kullanılır

Proje Yapısı

- `fuzzy_parking_system.py`: Bulanık mantık sisteminin çekirdek uygulaması
- `fuzzy_parking_gui.py`: Tkinter kullanarak GUI uygulaması
- `main.py`: Uygulama için giriş noktası
- `requirements.txt`: Gerekli Python paketleri

Gelecekteki Geliştirmeler

- Park sensörlerinden gerçek zamanlı veri entegrasyonu
- Mobil uygulama arayüzü
- Tahmine dayalı öneriler için geçmiş veri analizi
- Önerileri iyileştirmek için kullanıcı geri bildirimi entegrasyonu

Teşekkür

Bu proje, bulanık mantık kontrol sistemlerinin eğitim amaçlı bir gösterimi olarak geliştirilmiştir.