YAPAY SİNİR AĞLARI VE UZAKLIKLARIN KESTİRİMİ

İ. Kuban Altınel[1] ve Necati Aras[2]

SUMMARY

The actual distance between any two points on the earth surface is the length of the shortest road connecting them. Because it is often not feasible to measure the actual distances for all pairs of points, it is a common practice to use distance estimators. Then, the question is to choose a good estimator so that accurate distance approximations are obtained. Unlike traditional operations research methods, which compute aggregate parameters of functional estimators according to certain goodness-of-fit criteria, we utilize neural network approaches and vector quantization principles for estimating actual distances.

ÖZET

İki yerleşim merkezi arasındaki gerçek uzaklık, merkezleri bağlayan en kısa yolun uzunluğu olarak tanımlanır. Bir bölgedeki herhangi iki nokta arasındaki gerçek uzaklığı ölçmek ve saklamak her zaman olanaklı olmayabilir. Bu durumda gerçek uzaklığın kestirilmesi söz konusudur. Uzaklık kestirimi için geleneksel yaklaşım, aralarındaki uzaklığın hesaplanacağı noktalara ait kartezyen koordinatların parametrik bir fonksiyonu olan kestiriciler kullanmaktır. Daha değişik bir yaklaşım ise, uzaklıkları yapay sinir ağları yardımı ile kestirmek ve noktaları herhangi bir kestirim girişiminden önce bir yönlev (vektör) niceleme yöntemi kullanarak kümelemektir.

1. GİRİŞ

Yeryüzünde bulunan herhangi iki nokta arasındaki *gerçek uzaklık*, o noktaları birleştiren en kısa yolun uzunluğu olarak tanımlanır. Bir bölge içinde, noktalar arasındaki gerçek uzaklığı her nokta çifti için ölçme olanağı olmayabilir. Böyle bir bilgiye gereksinim olduğunda benimsenebilecek en iyi yaklaşım kestiriciler kullanmaktır. Bu nedenle gerçek uzaklıkların doğrulukla belirlenmesini sağlayacak iyi bir uzaklık kestiricisinin seçimi önemlidir.

Gerçek uzaklıkların doğru kestirimi birçok uygulamada önemli bir yer tutar. Örneğin, İstanbul için gerekli itfaiye istasyonlarının sayısını belirlemek için yaptıkları benzetim çalışmasında, Erkut ve Polat [1] gerçek uzaklığı, Öklid uzaklığı ile *yol*

[1] Doç. Dr., Boğaziçi Üniversitesi, Endüstri Mühendisliği Bölümü, Bebek, İstanbul
[2] Yüksek Endüstri Mühendisi, İdea A.Ş., Tuzla, İstanbul

katsayısı adını verdikleri bir sabiti çarparak kestirirler. Hemen hemen tüm yer seçimi problemleri; *dağıtım problemleri*, *gezgin satıcı problemi* ve *taşıt güzergahı belirlenmesi problemi*, formülasyonlarında gerçek uzaklıkların bilindiğini varsayar.

Uzaklık kestirimi problemini daha genel olarak ifade etmek olanaklıdır. Çok boyutlu bir *fiziksel* uzayda yer alan N adet noktadan (yerleşim merkezinden) oluşan G kümesini göz önüne alalım. Noktalar arasında bilinmeyen keyfi bir Ψ uzaklık fonksiyonunun bulunduğunu varsayalım. *Keyfi* sözcüğünün kullanılmasının nedeni, Ψ fonksiyonunun belirlediği uzaklıkların norm özelliğini sağlamama olasılığıdır. Ayrıca, Ψ fonksiyonu üçgen eşitsizliğini de doğrulamayabilir. Bununla beraber, uzaklık ölçütü kavramını az da olsa belirginleştirebilmek amacıyla, Ψ, Öklid normu ile şöyle ilişkilendirilebilir: İlk olarak $\Psi(P_i, P_j)$ sıfır değerini almalı ve $\Psi(P_i, P_j)$ simetrik olmalıdır. Ayrıca, G kümesinin herhangi dört noktası P_i, P_j, P_m ve P_n gözönünde bulundurulduğunda, eğer (P_i, P_j) ve (P_m, P_n) birbirlerine fiziksel uzayda yakınlarsa, (P_i, P_m) ve (P_j, P_n) çiftleri arasındaki keyfi uzaklıklar da aynı oranda yakın olmalıdır.

2. UZAKLIK KESTİRİMİ PROBLEMİ

Uzaklık kestirimi problemini sayısal olarak tanımlamak amacıyla kartezyen koordinatları $\left(x_a^1, x_a^2\right)^T$, $\left(x_b^1, x_b^2\right)^T$ olan P_a ve P_b noktaları düzlemde seçilmiş olsun. Amaç bu iki nokta arasındaki gerçek uzaklığı kestirmede kullanılacak $\Phi\left(P_a, P_b|\theta\right)$ kestiricisini belirlemektir. $\Pi_i =< P_i^1, P_i^2 >$, i inci nokta çifti ve r_i de P_i^1 ve P_i^2 noktaları arasındaki gerçek uzaklık olsun ve olası tüm nokta çiftleri ile noktalar arası gerçek uzaklıklar

$$S = \left\{ \left(P_i^1, P_i^2, r_i\right) : 1 \le i \le n,\ n = \binom{N}{2} \right\} \tag{1}$$

kümesi ile gösterilsin. Parametre yönlevi, θ, S kümesi kullanılarak,

$$\hat{\theta} = \arg\min_{\theta} \left\{ E\left[\delta\left(\Phi\left(P_i^1, P_i^2|\theta\right), r_i\right); S\right]\right\} = \arg\min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \delta\left(\Phi\left(P_i^1, P_i^2|\theta\right), r_i\right)\right\} \tag{2}$$

uyuşum ölçütü yardımıyla kestirilen yönlevdir. Burada $\delta(\cdot)$ kestirilen değerler ile gerçek değerler arasındaki kestirim hatasını ölçen *sapma ölçütü*dür. Bir seçenek, parametrelerin türevi alınabilen sürekli bir fonksiyonu olması nedeniyle gradyanın azaldığı yönde ilerlemeyi öngören, sinir ağları eğitimi gibi çeşitli alanlarda kullanımı olan enküçükleme yaklaşımlarının uygulanmasına olanak veren aşağıdaki ölçüttür:

$$\delta\left(\Phi\left(P_i^1, P_i^2|\theta\right), r_i\right) = \left(\frac{\Phi\left(P_i^1, P_i^2|\theta\right) - r_i}{\sqrt{r_i}}\right)^2 \quad (3)$$

Uzaklık kestirimi için geleneksel yaklaşım, aralarındaki uzaklığın hesaplanacağı noktalara ait, elde edilmesi kolay bilginin (örneğin noktalara ait kartezyen koordinatlar) parametrik bir fonksiyonu olan kestiriciler kullanmaktır. Bu yaklaşım [2] yaygın olarak uygulanır, çünkü bir kez parametreler hesaplandıktan sonra koordinatların basit birer fonksiyonu olan kapalı çözümler elde edilebilir. Ama, kestirimlerin doğruluğu parametrik fonsiyonun varsayılan biçimine sıkı sıkıya bağlıdır. Bu nedenle uygun olmayan varsayım kötü sonuçlara neden olur. Yakın tarihli bir çalışmada [3] algaçlar kullanan kestiricileri $\Phi\left(P_a, P_b|\theta\right)$ kestiricisini belirlemek için kullandılar. Bu yöntemler, parametresiz olmaları nedeni ile, herhangi bir önsel model varsaymazlar.

3. UZAKLIK FONKSİYONLARI

Herhangi iki nokta çifti arasındaki gerçek uzaklığı kestirmek amacıyla kullanılan yaygın yöntem, noktalara ait düzlemsel koordinatların parametreli bir fonksiyonu olan uzaklık fonksiyonları ile yaklaşık değerler elde etmektir. Uzaklık fonksiyonlarını, fonksiyonu oldukları koordinatlara göre sınıflandırmak olanaklıdır. Norm ya da norm tabanlı fonksiyonlardan yaygın kullanımları nedeni ile Tablo 1'de verilen dört tanesi en önemlileridir [4].

Tablo 1. En çok kullanılan uzaklık fonksiyonları ve parametreleri

Uzaklık Fonksiyonu	Parametreler (θ)
$\Phi_1\left(P_1, P_2\right) = k\left(\left\|x_{11} - x_{21}\right\| + \left\|x_{12} - x_{22}\right\|\right)$	k
$\Phi_2\left(P_1, P_2\right) = k\left(\left(x_{11} - x_{21}\right)^2 + \left(x_{12} - x_{22}\right)^2\right)^{1/2}$	k
$\Phi_3\left(P_1, P_2\right) = k\left(\left(x_{11} - x_{21}\right)^p + \left(x_{12} - x_{22}\right)^p\right)^{1/p}$	k, p
$\Phi_4\left(P_1, P_2\right) = k\left(\left(x_{11} - x_{21}\right)^p + \left(x_{12} - x_{22}\right)^p\right)^{1/s}$	k, p, s

Burada $\mathbf{x}_1 = \left(x_{11}, x_{12}\right)^T$, $\mathbf{x}_2 = \left(x_{21}, x_{22}\right)^T$ aralarındaki karayolu uzaklığını kestirmek istediğimiz P_1 ve P_2 noktalarının düzlemdeki kartezyen koordinatlarını gösterir. Parametre yönlevi θ 'yı oluşturan parametreler k, p ve s ise iyi yaklaşımlar sağlamak amacıyla kullanılırlar ve değerlerinin seçimi, kestirim doğruluğu açısından çok önemlidir.

4. ÇOK KATMANLI ALGAÇLAR

Çok katmanlı algaçlar *ileri beslemeli sinir ağları*dır. Bir sinir ağındaki her ünite girdilerinin ağırlıklı ortalamasını alır. İleri beslemeli bir sinir ağında, girdi giriş

katmanından başlayarak çıkış katmanına doğru genellikle tek katman halinde bulunan doğrusal olmayan saklı ünitelerden geçerek ilerler.

H adet saklı ünitesi olan tek saklı katmanlı bir ileri beslemeli sinir ağının çalışması matematiksel olarak şöyle belirtilebilir:

$$F(\mathbf{u}|\mathbf{T},\mathbf{W}) = \sum_{h=1}^{H} T_h g_h(\mathbf{W}_h, \mathbf{u}) + T_0 \tag{4}$$

Burada **u** çok boyutlu girdi yönlevi, $\mathbf{T}_h$, *h* saklı ünitesinden çıkış ünitesine olan bağlantı üzerindeki katsayı, T_0 ise yanlılığı belirleyen ağırlıktır. Böylece sinir ağının çıktısı saklı katman ünitelerine ait baz fonksiyonlarının ($g_h(\mathbf{W}_h, \mathbf{u})$) ağırlıklı ortalamasına dönüşür. Saklı katmanlar ilkin *d*-boyutlu girdi yönlevleri **u**'ların $\mathbf{W}_h$ katsayılarıyla ağırlıklandırılmış ortalamalarını alırlar. Daha sonra da bulunan değerleri doğrusal olmayan sigmoid uyarma fonksiyonu (activation function) ile düzlerler. Sigmoid uyarma fonksiyonu aşağıda verilmektedir:

$$H_h = g_h(\mathbf{W}_h, \mathbf{u}) = \frac{1}{1 + \exp\left(- \sum_{i=1}^{d} u_i W_{hi} - W_{0h} \right)} \tag{5}$$

Çok katmanlı algaçların eğitiminde kullanılan yaygın yöntem *geri yayılma* (back propagation) öğrenme algoritmasıdır. Bu yöntem gradyanın tersi yönde ilerleyerek bir hata fonksiyonuna ait enküçük değeri bulmayı amaçlayan bir iniş (descent) algoritmasıdır.

Bu makalede özetlenen gerçek uzaklıkların belirlenmesi ile ilgili uygulamada, hatayı ölçmek amacıyla (2) ve (3) numaralı denklemler kullanıldı. Sinir ağının girdisi aralarındaki gerçek uzaklığın kestirilmek istendiği nokta çiftlerine ait koordinat değerlerinin $\mathbf{u}^i = \phi(\mathbf{x}^i)$ şeklinde bir fonksiyonudur. Benzer şekilde, bilinen bir dönüşümle hesaplanan sinir ağının çıktısı da bize uzaklığın kestirilen değeri olan $y^i = d(\mathbf{x}^i|\theta) = \chi(F(\mathbf{u}^i|\mathbf{T},\mathbf{W}))$ değerini verir. Burada $\mathbf{x}^i$ nokta çifti *i*'ye ait koordinat değerlerini, y^i ise aynı çifti oluşturan noktalar arasındaki uzaklık kestirimini göstermektedir. Kestiriciye ait θ parametre yönlevi sinir ağının ağırlık katsayıları **W** ve **T** değerlerine karşılık gelir.

5. REGRESYON SİNİR AĞLARI

Bir parametresiz olasılık dağılımının herhangi bir değer için kestirimi örnek noktalarının etkilerinin ağırlıklı ortalamasıdır. *n* adet *d* boyutlu $\mathbf{X}_i$ değeri verildiğinde herhangi bir *n* değeri için olasılık dağılımı kestiricisi,

$$\hat{p}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h} \right) \tag{6}$$

olur [5]. Burada $K(\cdot)$ çekirdek yoğunluk fonksiyonu ve h çekirdeğin pencere genişliğidir. Ortak yoğunluk fonksiyonu çekirdek kestiricisi ile değiştirildiğinde ise *Nadaraya-Watson* kestiricisi elde edilir:

$$\hat{y}(\mathbf{x}) = \frac{\sum_i Y_i K\left((\mathbf{x}-\mathbf{X}_i)/h\right)}{\sum_i K\left((\mathbf{x}-\mathbf{X}_i)/h\right)} \tag{7}$$

Analitik özelliklerinin çekici olması nedeniyle (Silverman, 1986), çekirdek fonksiyonu genelde Gauss fonksiyonu olarak seçilir ki, buna *Parzen penceresi* adı verilir (Duda ve Hart, 1973). Bu daha sonra $\hat{y}(\mathbf{x})$ kestirimi için aşağıdaki formülün elde edilmesine neden olur:

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{n} Y_i \exp\left(-\left\|\mathbf{x}-\mathbf{X}_i\right\|^2/2h^2\right)}{\sum_{i=1}^{n} \exp\left(-\left\|\mathbf{x}-\mathbf{X}_i\right\|^2/2h^2\right)} \tag{8}$$

Görüldüğü gibi $\hat{y}(\mathbf{x})$ kestirimi, gözlemlenen Y_i değerlerinin, herbirisi değerlerin $\mathbf{x}$ yönlevine ait Öklid uzaklığı ile üstel orantılı olan çarpanlara sahip ağırlıklı ortalamasıdır. Burada pencere enini belirleyen h değerinin seçimi önemlidir. Bu değer çok geniş ise çok uzak komşular bile $\mathbf{x}$'deki kestirimi etkiler ve düzlenmiş bir kestirim elde edilir. Sınır durumda, yani h artı sonsuza giderken kestirim Y_i değerleri üzerinde alınan ortalamadan oluşur ve girdiden bağımsızdır. h çok küçük olduğunda ise, yalnızca birkaç örnek kestirime etki yapar ve gürültülü bir değere ulaşılır.

Specht [6] tarafından *genel regresyon sinir ağı* olarak isimlendirilen bu yaklaşım (4) denklemi çerçevesinde de ifade edilebilir. Burada H tablonun boyutuna, başka bir deyişle öğrenme örüntülerinin sayısına eşittir. Ayrıca $\mathbf{u}_h$'nın arzulanan çıktıyı gösterdiği varsayımı altında $\mathbf{W}_h = \mathbf{u}_h$ ve $T_h = y^h$ olur. Böylece,

$$g_h(\mathbf{W}_h, \mathbf{u}) = \frac{\exp\left(-\left\|\mathbf{W}_h - \mathbf{u}\right\|^2/2h^2\right)}{\sum_{i=1}^{H} \exp\left(-\left\|\mathbf{W}_i - \mathbf{u}\right\|^2/2h^2\right)} \tag{9}$$

şeklinde belirlenir.

6. BİRLEŞİK KESTİRİCİLER

Bu bölümde özetlenen yaklaşım tek bir seçim yapmaktansa seçenekleri birleştirerek başarımın daha da artırılabileceği düşüncesi üzerine oturur. Böylece sinir ağı üstün taraflarının yanı sıra zayıf taraflarıda bulunabilecek olan tek bir seçeneği öğrenmek yerine değişik seçeneklerin üstün taraflarını birleştirmeyi öğrenebilir.

Birden fazla kestiriciyi birleştirmenin iki temel yolu vardır: *oylama* ve *yığma*. Dikkat edilmesi gereken nokta birleştirilecek kestiricilerin farklı olmalarıdır. Bu çalışmada özetlenen sonuçlar üç kestiricinin birleştirilmesi ile elde edildi: bir parametrik

kestirici olan uzaklık fonksiyonu, çok katmanlı algaç ve regresyon sinir ağı. Birleştirilen bu kestiricilerin uzaklık kestirimi problemi açısından birbirlerinden çok farklı özelliklere sahip olduklarına inanılmaktadır [3].

6.1. Oylama

Oylamada c adet oy verenden her birisi, $d_j(\mathbf{x}|\theta_j)$ kestiricisi, girdi $\mathbf{x}$ yönlevi için bir çıktı verir ve son çıktı r ise bunların ağırlıklı ortalamalarından oluşur. Birleştirilen kestiricilerden birisi diğerlerine göre daha öncelikli olabilir. Bu durum o kestiriciye ait oyun değeri diğerlerine ait oyların değerlerinden daha büyük seçilerek sağlanabilir.

6.2. Yığma

Birden fazla kestiricinin birleştirilmesi ayrı bir kestirim problemi olarak görülebilir. Bu görüşü ilk ortaya atan Wolpert [7] düşüncesini *yığılmış genelleme* (stacked generalization) olarak adlandırır. Breiman [8] bu düşünceyi istatistiksel olarak yorumlar ve *yığılmış regresyon* (stacked regression) olarak nitelendirir. Burada iki kestirici katmanı vardır. İlk olarak sıfırıncı katman L_0, birleştirilmesi amaçlanan kestiriciler seçilerek oluşturulur. Bu kestiricilerin bireysel olarak belirli bir başarımı sağlamaları koşulu altında olabildiğince değişik olmalarına dikkat edilir. Bir üst katman olan L_1 ise girdisi L_0 katmanını oluşturan kestiricilerin çıktıları olan kestiricilerden oluşur.

7. YÖNLEV NİCELEME (YN) VE ÖZDÜZENLEMELİ HARİTA (ÖH)

YN ve ÖH'nın temelinde yatan düşünceler olasılık yoğunluk dağılımlarının kestiriminde uygulanan temel kavramlara dayanır. Geleneksel olarak, istatistiksel örüntü tanımada, dağılımlar ya parametreli ya da parametresiz olarak belirlenir. Parametreli dağılım halinde genelde dağılımın biçimi varsayılır ve eldeki veri noktaları kullanılarak dağılıma ait parametreler öğrenilir. Parametresiz yöntemlerde ise, verinin tamamıyla işleneceği varsayılır [9].

YN ise kestiriciler kullanarak bütün veriyi temsil etmek yerine, veriyi gerçek öznitelik uzayında temsil etmeyi amaçlar. Bununla beraber, sınıflandırmanın öğrenme ve sınama aşamalarında bütün veriyi kullanan parametresiz yöntemlerin aksine YN bilgiyi *şifre yönlevi* adı verilen küçük bir yönlev kümesi ile temsil ederek yoğunlaştırır. fiifre yönlevleri göz önüne alınan dağılımı hep birlikte temsil edebilecek hale gelinceye kadar öznitelik alanı içerisinde hareket ettirilirler. Bu evre *bölge içi kutuplama evresi* olarak adlandırılır. Çok sınıflı bir problemde, daha sonra, şifre yönlevleri temsil ettikleri bölgeleri doğru temsil edebilecek, hatta diğer bölgelerden ayırt edilmelerini sağlayabilecek hale gelinceye kadar yeniden hareket ettirilirler. *Bölgelerarası kutuplama evresi* denilen bu yeni evre daha sonraki sınıflandırma işlemlerinde kullanılabilecek olan ayırımcı fonksiyonu da öğrenir.

7.1. Bölge İçi Kutuplama

G kümesinin ögesi olan herhangi iki P_l, P_m noktası arasındaki $\Psi(P_l, P_m)$ uzaklığının kestirilmek istendiği durum gözönüne alınsın ve G'nin bir alt kümesi olan L

öğrenme kümesi içerdiği noktalar arasındaki gerçek uzaklık değerleri $\{\Psi(P_j, P_m) : P_j, P_m \in L\}$ ile birlikte verilmiş olsun. Çok bölgeli yaklaşım kullanan uzaklık kestirimindeki temel denence G'nin, Ψ için bölgeler içi ve bölgeler arası yaklaşıklamalar sağlayacak şekilde, daha küçük alt bölgelere ayrılabileceğidir. Bu nedenle, öğrenme evresinde, L öğrenme kümesi herbiri N_k nokta bulunduran R adet $C_k = \{P_{k,i} : 1 \le i \le N_k,\ 1 \le k \le R\}$ alt kümesine bölünür. Burada $P_{k,i}$ k bölgesinin i noktasını belirlemektedir. Ana amaç her C_k kümesini küme boyu N_k'dan çok daha küçük olan M adet $\{Q_{k,j} : 1 \le j \le M\}$ nokta ile temsil etmektir. Bu çalışmada [10] her bölge için üç şifre yönlevi kullanılmıştır. Her bölge, bir sonraki evre olan bölgeler arası kutuplamadan önce, bu bölge içi kutuplama evresinden geçer.

7.2. Bölgelerarası Kutuplama

Her bölge yukarıda açıklanan yer değiştirme yöntemi uygulanarak yerleştirilen M adet şifre yönlevi ile temsil edildikten sonra, şifre yönlevleri kendilerine ait noktaları doğru sınıflandırmaları açısından L kümesini kullanarak sınanırlar. Öğrenme kümesi L'deki her veri noktası Q_k'ların kümesine karşı denenirler. Bu deneme en yakın şifre yönlevinin kendilerine ait bölgeye düşüp düşmediğini görerek gerçeklenir. Bu evrede, temsilci şifre yönlevler diğer rakip kümelere ait veri noktalarından uzaklaşacak şekilde hareket ettirilirler.

Bölgeler içi ve bölgeler arası kutuplama evrelerinden sonra temsilci şifre yönlevlerinin belirlediği parçalı doğrusal sınırlar, L kümesini oluşturan noktaları başlangıçta atandıkları ilk bölgelerden biraz daha farklı bölgelere dahil ederler. Öğrenme kümesindeki noktalar bölgelere dağıtıldıktan ve her bölgeye ait şifre yönlevleri öğrenildikten sonra (yukarıda bahsedilen hareket ettirme yöntemleri kullanılarak), artık y gerçek uzaklık fonksiyonunu yaklaşıklayan alt fonksiyonların öğrenilmesine geçilebilir.

7.3. Vektör Niceleme Kullanarak Parametrelerin Öğrenilmesi

Ψ fonksiyonu, bölgelerin, belirlenen şifre yönlevleri ile temsil edilmeleriyle kestirilebilir. Bu evredeki temel varsayım, Ψ'nin bir bölgeler arası alt fonksiyonlar örüsü ile yaklaşıklanabileceğidir. Bu evrede, alt fonksiyonlar, şifre yönlevleri ve L öğrenme kümesini oluşturan P_i, P_j nokta çiftleri ile aralarındaki $\Psi(P_i, P_j)$ gerçek uzaklıklar kullanılarak öğrenilir. Burada, noktalar arasındaki gerçek uzaklıklar en yakın şifre yönlevleri ile tanımlanan ve fonksiyonel biçimi Tablo 1'de verilenlerden birisine yakın olan bir fonksiyon tarafından belirlenmektedir. Böylece kestirim problemi bilinen eniyileme problemlerinden birisi haline dönüşür [2,3].

8. DENEYSEL SONUÇLAR

8.1. Veri Tabanı

Türkiye'de bulunan yerleşim merkezlerini kullanarak iki ayrı örnek kümesi elde edildi. Bu kümelerden öğrenme kümesi YN yöntemi için 80, diğer yöntemler için 28 yerleşim merkezinin; sınam kümesi ise tüm yöntemler için 23 yerleşim merkezinin çiftler halinde eşlenmesinden oluşur. Koordinatlar iki boyutludur. Öğrenme kümesi

parametrelerin kestiriminde, sınam kümesi ise kestirilen parametrelerin başarımlarının değerlendirilmesinde kullanıldı. Raporlanan değerler hem öğrenme kümesi hem de sınam kümesi için nokta çifti başına düşen ve (3) eşitliği ile verilen normalleştirilmiş hata ölçütü kullanılarak bulunan kestirim hatasını kilometre cinsinden gösterirler.

8.2. Çok Katmanlı Algaçlar

Burada özetlenen sonuçlarda bir saklı katmanı olan çok katmanlı algaçlar kullanılmakta ve öğrenme geri yayılma kuralı ile gerçeklenmektedir. Ayrıca girdi gösteriminin nokta çiftlerine ait dört koordinat değerinin yanısıra nokta çiftleri arasındaki Öklid uzaklığını da içerir. Buna göre girdi, $\mathbf{u} = \phi(\mathbf{x}) = (x_{11}, x_{12}, x_{21}, x_{22}, \|\mathbf{x}_1 - \mathbf{x}_2\|)^T$ olur. Çıktı gösterimi ise gerçek uzaklığın Öklid uzaklığına bölünmesinden oluşur [3].

8.3. Regresyon Sinir Ağları

Burada kestirici girdi $\mathbf{u} = \phi(\mathbf{x}) = \left(|x_{11} - x_{21}|, |x_{12} - x_{22}|, \|\mathbf{x}_1 - \mathbf{x}_2\|\right)^T$ olarak gösterilir. Çıktı ise kestirilmek istenen gerçek uzaklık $y = \chi(r) = r$ olarak belirlenir. Sonuçlar regresyon sinir ağlarının çok katmanlı algaçlara göre biraz daha kötü kestiriciler olduklarını gösterir.

8.4. Birleşik Kestiriciler

8.4.1 Oylama: Birleştirilen kestiricilerin değişik oldukları oranda birleştirmeye ait başarımın arttığı gerçeğini gözönünde bulundurarak aşağıdaki şu üç kestirici birleştirilebilir: parametreleri *k*, *p* ve *s* olan uzaklık fonksiyonu $\Phi_4(P_1, P_2)$, 12 saklı üniteli ve tek katlı sakmanlı çok katmanlı algaç ve $h = 54$ olan regresyon sinir ağı. Oylar eşit alındığında Tablo 2'de verilen sonuçlar elde edilir. Bunlara göre oylama sonuçları tek başına oy verenlere ait sonuçlardan daha iyidir ve birleştirilen kestiriciler bazı durumlarda başarılı değildirler.

Tablo 2. Üç Kestiricinin ve "Oylama"nın Ortalama Hataları

Kestirici	Öğrenme Kümesi Hatası	Sınam Kümesi Hatası
Parametreli Model	3.61	8.10
Çok katmanlı algaç	2.63	7.91
Regresyon sinir ağı	2.38	8.49
Oylama	2.26	7.63

8.4.2. Yığma: L_0 katmanı için yine üç kestirici kullanılabilir. Bunlar, tek parametresi olan $\Phi_2(P_1, P_2)$ uzaklık fonksiyonu, 12 saklı üniteli ve tek katlı sakmanlı algaç ve $h = 54$ olan regresyon sinir ağıdır. Tablo 3'te özetlenen sonuçlardan da görülebileceği gibi yığma oylamadan daha başarılıdır.

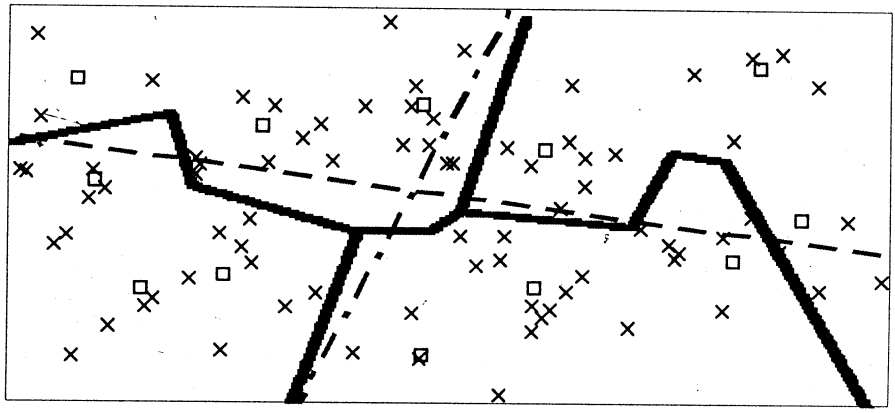
Tablo 3. Ortalama Hatalar

Birleştirici	Öğrenme Kümesi Hatası	Sınam Kümesi Hatası
Oylama	2.26	7.63
Yığma	3.81	7.41

8.5. Yönlev Niceleme

Bu makalede tanıtılan algoritmaları uygulayabilmek için önce Türkiye dört alt bölgeye bölündü. Şekil 1'de, öğrenme kümesinde kullanılan Türkiye'ye ait yerleşim merkezleri görülmektedir. Burada, dikdörtgenin sınırları sırasıyla 36° ve 42° kuzey enlemler ve 26° ve 45° doğu boylamlardan oluşmaktadır. Yerleşme merkezleri 'X' ile gösterilmişlerdir. Görüntü açısından karışıklığa neden olmaması amacıyla Türkiye'nin gerçek haritası şekil üzerine bastırılmadı. On iki adet kare ' ' ise şifre yönlevlerinin son konumlarını göstermektedir.

Daha önceki bölümlerde sözü edildiği gibi, en son bölme (şifre yönlevleri yakınsadıktan sonra) şifre yönlevlerini birleştiren doğruların orta noktalarından çizilen dikmeler yardımıyla belirlenen ayırım fonksiyonu ile elde edildi. Uyarlamalı olarak öğrenilen bu bölme koyu çizgilerle Şekil 1'de görülmektedir.



Şekil 1. Türkiye'nin Uyarlamalı Olarak Bölünmesi

Bölgeler içi ve bölgelerarası öğrenme evrelerinden sonra uzaklık fonksiyonunu yaklaşıklayan alt fonksiyonlar örüsüne ait parametreler öğrenme kümesindeki yerleşim merkezleri ve aralarındaki uzaklıklar kullanılarak kestirilmiş olur. Sonuçlar Tablo 4'te sergilenmektedir. Deneylerle ilgili ayrıntılar ve şekiller yakın tarihli bir çalışmada bulunabilir [10].

Tablo 4. Ortalama Hatalar

Kestirici	Öğrenme Kümesi Hatası	Sınam Kümesi Hatası
$\Phi_2(P_1, P_2)$	2.36	7.90
$\Phi_3(P_1, P_2)$	2.11	7.48
$\Phi_4(P_1, P_2)$	1.93	7.12

9. SONUÇLAR

Açıklanan yöntemlerin hepsi Türkiye'nin yerleşim merkezlerine ait veri kullanılarak sınanmış; önerilen algoritmaların çok hızlı yakınsadıkları ve iyi sonuçlar verdikleri gözlenmiştir. Sonuçlara genel olarak göz atıldığında, özdüzenlemeli harita kullanılarak yapılan yönlev niceleme ile yerleşim merkezlerinin kümelendiği ilk aşamadan sonra, parametrik kestiricilere ait parametrelerin kestiriminden oluşan iki aşamalı yöntemin kestirim hatası açısından en başarılı sonuçları verdiği görülmektedir. Çok katmanlı algaçlar ve birleşik kestiriciler kullanan karma yöntemler, yerel etkiyi eniyi modelleyen yönlev niceleme tabanlı iki aşamalı yönteme göre daha başarısız olmaktadır. Ancak, kestirim hatası açısından en kötü sonucu parametreli uzaklık fonksiyonları vermektedir. Bu çalışmada algaç tabanlı yöntemlerde kullanılan öğrenme verisi yönlev nicelemede ve uzaklık fonksiyonlarının parametrelerini hesaplamada kullanılan öğrenme verisinin bir alt kümesidir.

Özetle Yönlev Niceleme'nin ve Kohonen'e ait Özdüzenlemeli Harita'nın bilinen eniyileme yöntemlerinden ve algaç tabanlı yöntemlerden üstün olduğu söylenebilir.

KAYNAKLAR

[1] Erkut, H., and Polat, S. (1992), "A Simulation Model for an Urban Fire Fighting System", *Omega* 20, 535-542.

[2] Love, R.F., and Morris, J.G. (1972), "Modeling Inter-City Road Distances by Mathematical Functions", *Operational Research Quarterly* 23, 61-71.

[3] Alpaydın, E., Altınel, İ.K. ve Aras, N. (1996), "Parametric Distance Metrics vs. Nonparametric Neural Networks for Estimating Road Travel Distances", *European Journal of Operational Research* 92, 230-243.

[4] Francis, R.L., McGinnis L. F. Jr., and White, J.A. (1992), *Facility Layout and Location : An Analytical Approach, 2nd edition*, Prentice Hall, Englewood Cliffs.

[5] Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London.

[6] Specht, D.F. (1991), "A General Regression Neural Network", *IEEE Transactions on Neural Networks* 2, 568-576.

[7] Wolpert, D.H. (1992), "Stacked Generalization", *Neural Networks* 5, 241-259.

[8] Breiman, L. (1992), "Stacked Regression", TR-367, Department of Statistics, University of California, Berkeley.

[9] Duda, R.O., and Hart, P.E. (1993), *Pattern Classification and Scene Analysis*, John Wiley.

[10] Altınel, İ.K., Oommen, B.J., and Aras, N. (1997), "Vector Quantization for Arbitrary Distance Function Estimation", *INFORMS Journal on Computing* 9, No. 4, 439-451.