

LVQplus :LVQ Algoritmasına Bir Eklenti ve Etkileri

LVQplus : Extended LVQ Algorithm and Its Effects

M.Fatih Amasyalı

YTÜ Bilgisayar Mühendisliği Bölümü
mfatih@ce.yildiz.edu.tr

Özetçe

Learning Vector Quantization(LVQ) algoritması bilgi kuantalamada kullanılan başlıca yöntemlerden birisidir. Bu çalışmada LVQ'nun eğitim süresi kısaltılmaya çalışılmış ve birbirini kapsayan datalarda daha iyi bir performans göstermesi için algoritmaya bir ek yapılmıştır. Çalışma sonunda, yapılan eklemenin orjinal LVQ algoritmasının etkinliğini artırdığı görülmüştür.

Abstract

Learning Vector Quantization(LVQ) algorithm is one of the popular quantization algorithm. In this paper, a new version of LVQ has been developed. The new version requires less time for trainig phase and has more performance on data which include each other.

1.Giriş

Learning Vector Quantization(LVQ) algoritması Kohonen tarafından bilgi kuantalamada kullanılması için önerilmiştir[1]. LVQ algoritmasında başlangıçta kuantalanmak istenen bilgilerle aynı boyuttaki, kullanıcı tarafından belirlenen sayıda betimleyici(representative) vektör rasgele olarak seçilir. Eğitim setinin her bir örneği için, bu örneğe rasgele seçilen betimleyici vektörlerden en yakın olanı belirlenir ve eğer örnekle betimleyici vektör aynı etikete sahip ise(aynı sınıftan ise) betimleyici vektör o sınıfı daha iyi temsil etmesi için örnek vektöre yaklaştırılır. Eğer farklı etiketlere sahipseler daha sonraki adımlarda örneğin sınıfıyla aynı sınıftan olan betimleyici vektörler seçilsin diye bu adımda seçilen betimleyici vektör örnek vektörden uzaklaştırılır. Diğer bir ifadeyle her adımda betimleyici vektörlerden biri kazanır ve eğer doğru sınıflandırma yapılmışsa ödüllendirilir, yanlış sınıflandırma yapılmışsa cezalandırılır.

Orjinal LVQ algoritması ilk değer atamaya çok bağlıdır. LVQplus ise ilk değer atamasından daha az etkilenecek hızlı ve yüksek doğruluk yüzdesi olan bir eğitim sağlamaktadır. Orjinal LVQ algoritmasında eğer iki betimleyici vektör birbirine çok yakınsa bu ikisini birbirinden ayırmak fazla eğitim adımı gerektirir. Ancak bu çalışmada önerilen LVQplus bu adım sayısını düşürmektedir.

2. LVQplus: Yeni Bir Learning Vector Quantization Algoritması

Orjinal LVQ algoritmasının zayıf yönleri olan ilk değer atamaya çok bağımlı olmaları ve birbirine yakın betimleyici

vektörleri birbirinden uzun zamanda ayırabilmeleri problemlerine değinilmiştir.. İlk değerlere bağımlı olmayı eğitim datasının ortalamasını ilk betimleyici vektörler olarak belirlemek birçok durumda bu problemi çözmektedir. Ancak bu durumda vektörler (aynı değerlerle başlangıç yaptıkları için) birbirlerinden ayrılmaları uzun zaman almaktadır. Ayrıca birbirini çevreleyen farklı sınıftan datalar olduğu durumda bu vektörler orjinal LVQ algoritmasıyla belirli bir bölgede sıkışıp kalmaktadır ve doğru sınıflandırma yapamamaktadır.

İşte bu problemleri çözebilmek için bu çalışmada LVQplus algoritması ortaya atılmıştır.

LVQplus'ta orjinal LVQ algoritmasındaki cezalandırma ve ödüllendirmeye ek olarak;

Yanlış sınıflandırma durumunda doğru sınıflandırma yapmaya en yakın olan betimleyici vektör, eğitim örneğine yaklaştırılır.

2.1. LVQ ve LVQplus Arasındaki Farkın Grafiksel Gösterimi

Bu bölümde Orjival Algoritma ile Önerilen algoritma arasındaki fark grafiksel gösterilimi anlatılmıştır.

2.1.1. LVQ'da ödüllendirme

Şekil 1'de iki adet sınıf(siyah ve gri) ve her sınıfa ait ikiye adet betimleyici vektör bulunmaktadır. Vektör(X) ise eğitim datasından bir vektördür ve siyah sınıftandır. Datalarımız grafik gösterimi kolay olsun diye 2 boyutlu uzayda yer almaktadırlar.

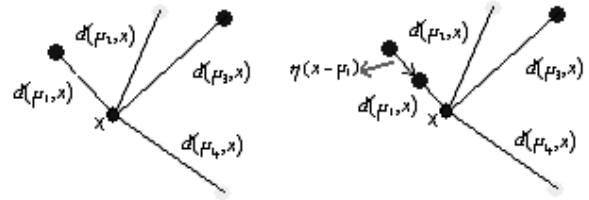
İki vektör arasındaki mesafenin belirlenmesinde öklid uzaklığı kullanılmıştır. Şekil 1'de x eğitim datasına betimleyici vektörlerin uzaklıkları arasında eşitlik 1'deki gibi bir bağıntı vardır.

$$d(\mu_1, x) < d(\mu_2, x) < d(\mu_4, x) < d(\mu_3, x) \quad (1)$$

Eğitim datasına en yakın betimleyici vektör μ_1 dir. X ve μ_1 in sınıfları aynıdır(siyah). Bu durumda μ_1 vektörü X vektörüne eşitlik 2'deki şekilde yaklaştırılacaktır.

$$\mu_1 \leftarrow \mu_1 + \eta(x - \mu_1) \quad (2)$$

Bu güncelle işlemi şekil 2'de gösterilmiştir.



Şekil 1

Şekil 2

2.1.2. LVQ'da cezalandırma

Eğitim işlemimizi gerçekleştireceğimiz ortam şekil 3'teki gibi olsun. Şekil 1'den tek farkı x eğitim datasının sınıfının gri olmasıdır.

x eğitim datasına betimleyici vektörlerin uzaklıkları arasında bağıntı yine eşitlik 2'deki gibidir.

Eğitim datasına en yakın betimleyici vektör μ_1 dir. X ve

μ_1 in sınıfları farklıdır(siyah ve gri). Bu durumda μ_1 vektörü X vektöründen eşitlik 3'teki şekilde uzaklaştırılacaktır.

$$\mu_1 \leftarrow \mu_1 - \eta(x - \mu_1) \quad (3)$$

Bu güncelle işlemi şekil 4'de gösterilmiştir.

Bu uzaklaştırma işlemi sayesinde X in bir sonraki sunumunda

μ_2 betimleyici vektörünün kazanması sağlanmıştır.

2.1.3. LVQplus'da ödüllendirme

LVQ ile aynı şekilde yapılır.

2.1.4. LVQplus'da cezalandırma

LVQ dan farklı olarak 1 değil 2 güncelleme işlemi yapılır.

Eğitim işlemimizi gerçekleştireceğimiz ortam yine şekil 3 gibi olsun.

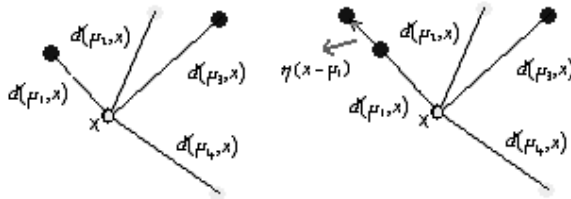
X eğitim datasına betimleyici vektörlerin uzaklıkları arasında bağıntı yine eşitlik 2 deki gibidir.

Eğitim datasına en yakın betimleyici vektör μ_1 dir. X ve

μ_1 in sınıfları farklıdır(siyah ve gri). Bu durumda μ_1 vektörü X vektöründen eşitlik 4'teki şekilde uzaklaştırılacaktır. Ayrıca X eğitim datasıyla aynı sınıftan olan ve X e en yakın olan vektör μ_2 eşitlik 4'teki şekilde X vektörüne yaklaştırılacaktır.

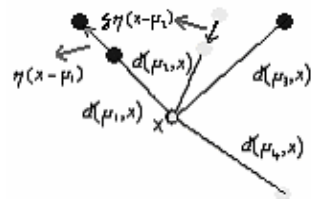
$$\mu_2 \leftarrow \mu_2 + \delta\eta(x - \mu_2) \quad (4)$$

Bu güncelle işlemleri şekil 5'de gösterilmiştir.



Şekil 3

Şekil 4



Şekil 5

2.2. LVQplus Algoritmasının Adımları

LVQ algoritmasına çok benzer olmasına rağmen algoritmanın bir bütün içinde gösterilmesi amacıyla tüm algoritma aşağıda verilmiştir.

[η] öğrenme oranı

[δ] 2.öğrenme oranı

[n] maximum eğitim sayısı

[c] betimleyici vector sayısı

[μ_1, \dots, μ_c] betimleyici vektörler (centroids)

[x] eğitim datasından bir örnek

[S(x)] x vektörünün ait olduğu yada betimlediği sınıf olmak üzere

1. $\eta, \delta, n, \mu_1, \dots, \mu_c$ için ilk değer

atamalarını gerçekleştir

2.Eğitim adımları

2.1 X eğitim datasını al

2.2 X e en yakın betimleyici vektörü bul

$$(\mu_k) : k \leftarrow \arg \min_j \|x - \mu_j\| \quad j=1..c$$

2.3 μ_k nın güncellenmesi:

Eğer x doğru sınıfsa ($s(x)=s(\mu_k)$ sınıfları aynı ise)

$\mu_k \leftarrow \mu_k + \eta(x - \mu_k)$ ödüllendir x'e yaklaştır

değilse

$\mu_k \leftarrow \mu_k - \eta(x - \mu_k)$ cezalandır x'den uzaklaştır

ve

LVQplus eklentisi başlangıç

x in sınıfından olan ve x e en yakın betimleyici vektörü

bul (μ_f) ve güncelle

$$f \leftarrow \arg \min_j \|x - \mu_j\|$$

$$j=1..c, s(x) = s(\mu_f)$$

$$\mu_f \leftarrow \mu_f + \delta(x - \mu_f)$$

ödüllendir x'e yaklaştır

LVQplus eklentisi bitiş

3. μ larda belirgin bir değişim olmuşsa veya Maximum iterasyon sayısına erişilmemişse 2. adıma dön. Değilse çık.

3.Uygulama Sonuçları

LVQplus algoritmasının değerlendirmesi için hem yapay hem gerçek dünya datalarından faydalanıldı.

3.1. Gerçek Verilerin Uygulama Sonuçları

Data setlerinin özellikleri Tablo 1'de verilmiştir.

Tablo 1: Gerçek verilerin özellikleri

Data Set	Ecoli	Cam
Sınıf Sayısı	6	8
Boyut Sayısı	9	7
Eğitim data sayısı	170	266
Test data sayısı	44	70

Eşitlik 5'teki Sigma'nın değerlerine göre, ilk değer atamasının yapıış şekline göre ve betimleyici vektör sayısına göre performansları Tablo 2'de verilmiştir. Değerlerin ilki centroidler(betimleyici vektör)in ilk değerlerinin ortalama alınarak atandığındaki, ikinci değerler ise rastgele atandığındaki test performanslarını göstermektedir.

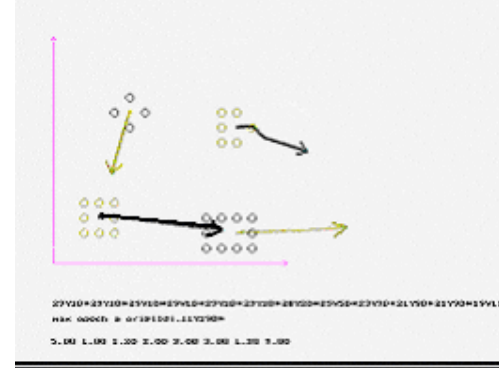
Tablo 2: Gerçek verilerin uygulama sonuçları

		LVQplus			LVQ
	Centroid	Sigma=1	Sigma=3	Sigma=5	
CAM 44 test	18	32/27	29/24	27/28	28/28
	24	29/30	30/28	29/27	27/23
	30	29/25	30/23	29/24	27/31
ECOLI 70 test	18	59/62	54/58	57/57	59/63
	24	58/62	54/62	54/60	59/62
	30	61/62	60/60	60/60	61/63

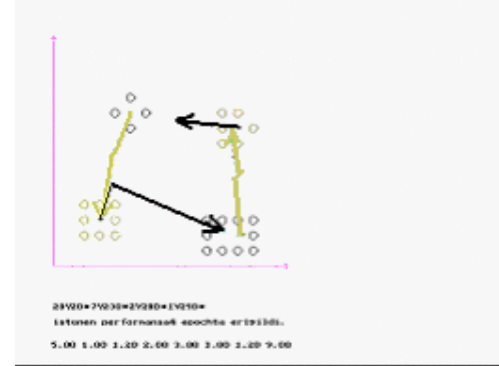
3.2. Yapay Verilerin Uygulama Sonuçları

LVQ ve LVQplus algoritmalarının aynı data setler üzerinde betimleyici vektörlerin değerlerini(konumlarını) nasıl değiştirdiği bir örnek üzerinde incelersek : Şekil 6'da 2 sınıf ve her sınıf için ikişer betimleyici vektör bulunmaktadır. Vektörler betimleMEDikleri sınıfların ortalarına yerleştirilmiştir. Bu betimleyici vektörlerin ilk değerleri rastgele seçildiğinde oluşabilecek bir durumdur. Şekil 6(a) ve (b)'de vektörlerin başlangıç değerleri aynıdır. Şekil 6(a)'da LVQ uygulandığında, (b)'de LVQplus uygulandığında vektörlerin izledikleri rotalar verilmiştir. LVQ'da vektörler ait oldukları sınıflara bir türlü yaklaşmazken, LVQplus'da az bir eğitimle yaklaşabilmişlerdir.

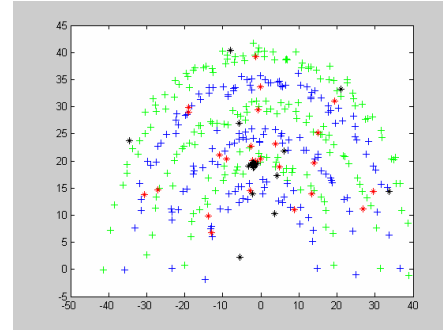
Birbiri içine geçmiş datalarda iki algoritmanın betimleyici vektörleri son olarak nerelerde bıraktıklarını İki yapay veri setinde incelersek : Şekil 7'de açık ve koyu renkli artılar eğitim datasını, şeklin her yanına dağılmış olan yıldızlar LVQplus'ın betimleyici vektörlerinin yerlerini, şeklin genelde ortasında kalmış noktalar ise LVQ'nun betimleyici vektörlerinin yerlerini göstermektedir. Şekilde görüldüğü gibi LVQplus eğitim datasını çok daha iyi temsil edebilmektedir. LVQ'da betimleyici vektörler ortada sıkışmışlarken, LVQplus'la datanın her yerine dağılmayı başarmışlardır.Sol taraftaki eğitim setinde LVQ, 336 datanın 218'ini doğru sınıflandırabilirken ; LVQplus 278'ini doğru sınıflandırabilmiştir.



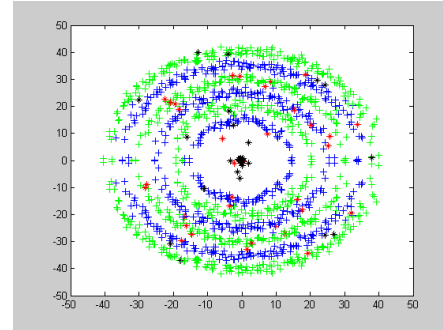
Şekil 6 (a)



Şekil 6(b)

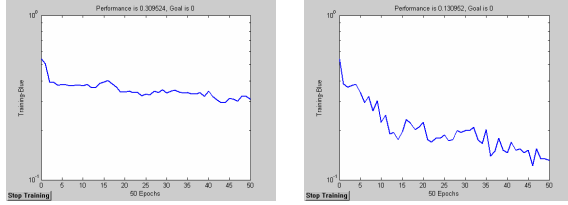


Şekil 7(a)



Şekil 7(b)

Eğitim sırasındaki her iterasyondaki yanlış sınıflandırma sayılarını incelersek :



Şekil 8(a)

Şekil 8(b)

Şekil 8’deki eğitim eğrilerinden soldaki LVQ’ya, sağdaki LVQplus’a aittir. Görüldüğü gibi LVQplus’ta birbiri içine girmiş datalarda eğitim daha iyi ve hızlı gerçekleşmektedir.

LVQ plus gerçek verilerde LVQ ile yakın performansa, birbiri içine girmiş, birbirini kapsayan datalarda LVQ’dan çok daha yüksek bir performansa erişmiştir.

4.Sonuç

Orjinal LVQ algoritmasının ilk deger atamalarına karşı oldukça duyarlı olduğu şekil 6’da görülmektedir. Dataların ortalamasından ilk değerler üretildiğinde ise birbirini kapsayan datalarda betimleyici vektörlerin dıştaki sınıflara pek kolay erişemediği Şeki 7’de görülmektedir. Ayrıca her eğitim datasında 1 (ya bir ödül, ya bir ceza) betimleyici vektörü hareket ettirerek ağı eğitmek zaman alıcı bir işlemdir.

Bu sorunların üstesinden gelinebilmesi için geliştirilen LVQplus ; ilk deger atamalarına LVQ kadar bagimli olmadigi görülmüştür. Çünkü betimleyici vektörler sigma katsayısı sayesinde daha hızlı ve adaptif hareket edebilmektedirler. Ortalama olarak ilk deger atamasında yine LVQplus’ın dıştaki sınıf yada sınıfları temsil etmesinin LVQ’ya göre yine beta katsayısı sayesinde daha kolay olduğu görülmüştür.

Son olarak her eğitimde ortalama 1.5 (ya bir ödül, ya bir ceza ve bir ödül) betimleyici vektörü hareket ettirerek eğitim için gereken zaman azaltılmıştır.

Sonuç olarak birbirini kapsayan datalarda performansı yükseltmek için, diğer tür datalarda ise eğitim zamanını azaltmak için LVQplus kullanılabilir.

5.Kaynakça

- [1] T.Kohonen," Self-Organization and associative Memory",3d ed, 1989, Berlin :Springer-Verlag.