Zamanla Değişen Dağılımların Evrensel Tahmini
Estimating Distributions Varying In Time In A Universal Manner

Kaan Gökçesu[1], Eren Manış[2], Ali Emirhan Kurt[2], Ersin Yar[1]
[1]Elektrik ve Elektronik Mühendisliği Bölümü, Bilkent Üniversitesi, Ankara, Türkiye
{gokcesu,eyar}@ee.bilkent.edu.tr
[2]Bilgisayar Teknolojisi ve Bilişim Sistemleri Bölümü, Bilkent Üniversitesi, Ankara, Türkiye
{eren.manis,ali.kurt}@ug.bilkent.edu.tr

***Özetçe* —Zamanla değişen parametrelere sahip olan dağılımların kestirimini incelemekteyiz. Gerçek olasılık dağılımına karşı en iyi negatif olabilirliği başaran bir algoritma sunuyoruz. Gerçek dağılımın parametrelerinin toplam değişikliği hakkında hiçbir bilgi sahibi olmaksızın bu en iyi pişmanlık performasına ulaşmaktayız. Sonuçlarımızın, temelde var olan diziler hakkında hiçbir varsayım olmaksızın ayrık bir dizi bağlamında sağlanacağı garanti edilmektedir. Pişmanlık sınırlarının yanı sıra, yapay deneyler ve gerçek hayat deneyleriyle literatürdeki modern olasılık yoğunluğu kestirim algoritmalarına göre önemli bir performans sergilemekteyiz.**

Anahtar Kelimeler—Ardışık yoğunluk kestirimi, üstel aile, durağan olmayan kaynak, ayrık dizi biçimi.

***Abstract*—We investigate the estimation of distributions with time-varying parameters. We introduce an algorithm that achieves the optimal negative likelihood performance against the true probability distribution. We achieve this optimum regret performance without any knowledge about the total change of the parameters of true distribution. Our results are guaranteed to hold in an individual sequence manner such that we have no assumptions on the underlying sequences. Apart from the regret bounds, through synthetic and real life experiments, we demonstrate substantial performance gains with respect to the state-of-the-art probability density estimation algorithms in the literature.**

Keywords—Sequential density estimation, exponential family, nonstationary source, individual sequence manner.

I. GİRİŞ

Bu makalede, her t anında sıralı olarak gözlemlenen $\{x_1, x_2, \ldots\}$ kullanılarak öğrenilen ve çeşitli makine öğrenme uygulamalarında [1]–[6] karşılaşılan sıralı olasılık kestirimi araştırılmaktadır. Mühendislik sistemlerindeki çoğu uygulamada, verinin istatistiksel özellikleri (özellikle büyük veri uygulamalarında) zamanla değişebileceğinden dolayı $\{x_t\}_{t\geq 1}$'nin düzensiz hafızasız kaynaktan üretildiği varsayılmaktadır [7]. Bu probleme, karşı tarafın gerçek olasılık dağılımı fonksiyonu olduğu rekabetçi bir bakış açısından yaklaşılmaktadır. Her t anında, bilinmeyen $f_t(x_t)$'e göre oluşan örnek bir öznitelik vektörü x_t gözlemlenmektedir. Geçmiş gözlemler olan $\{x_\tau\}_{\tau\geq 1}^{t-1}$'e dayanılarak bir tahmin $\hat{f}_t(x_t)$ oluşturulmaktadır. Hata fonksiyonu olarak olasılık dağılımları için en yaygın kullanılan logaritmik hata fonksiyonu, $-\log(\hat{f}_t(x_t))$, kullanılmaktadır [8]. Ayrık dizi (Individual Sequence) bağlamında güvenilir sonuçlar elde etmek için [9], logaritmik hatada "pişmanlık" kavramı kullanılarak performans tanımı yapılmaktadır. Bunun sonucunda t anındaki pişmanlık $r_t = -\log(\hat{f}_t(x_t)) + \log(f_t(x_t))$ iken, T anına kadar olan birikmiş pişmanlık ise $R_T = \sum_{t=1}^{T}\left(-\log(\hat{f}_t(x_t)) + \log(f_t(x_t))\right)$ olmaktadır. Üstel familyadan en iyi durağan olmayan dağılımın performansının elde edilmesi amaçlanmaktadır. Bu bağlamda, doğru dağılımı $f_t(x_t)$ tam olarak veya en yakın olacak şekilde temsil eden bir yoğunluk fonksiyonu olduğu varsayılmaktadır ve bu fonksiyon muhtemelen değişen bir parametre α_t'e sahip üstel ailenin bir parçasıdır [10]. Üstel aileden gelen dağılımlar özellikle incelenmektedir çünkü bunlar geniş bir parametrik model [6] aralığını kapsamakta ve olasılık dağılımlarının birçoğunun parametrik olmayan [11] sınıflarını doğru olarak kestirmektedir. α_t'deki T turda toplam sapma C_α değişkeni ile şöyle gösterilebilir

$$C_\alpha \triangleq \sum_{t=2}^{T} \|\alpha_t - \alpha_{t-1}\|. \tag{1}$$

$\|\cdot\|$ yukarıda L^2 metriğini belirtmektedir. Doğal parametrenin değişmediği durağan kaynaklar için $C_\alpha = 0$'dır. [6] ve [12] deki gibi, belirli bir hesaplama karmaşıklığına sahip sabit bir kaynak için pişmanlık sınırı $O(\log(T))$ olarak gösterilebilir. Ancak, sabit kaynaklar için logaritmik pişmanlık sınırı düşük hesaplama karmaşıklığı altında uygulanamaz [6]. [13], T zamanı ve C_α parametre vektöründeki toplam sapma önceden bilindiğinde sabit karmaşıklığa sahip $O(\sqrt{C_\alpha T})$ pişmanlık sınırına ulaşan bir algoritmayı sunmaktadır. C_α hakkında bir ön bilgi verilmemesi durumunda $O(C_\alpha\sqrt{T})$ pişmanlık sınırını sağlayan sabit karmaşıklığa sahip bir algoritma [6]'da önerilmiştir. Bu yüzden durağan olmayan bir kaynağın C_α (sürüklenme) hakkında herhangi bir ön bilgi bilinmemesi durumunda $O(\sqrt{C_\alpha T})$'nın elde edilmesi modern yöntemler ile mümkün değildir.

Literatürde ilk kez, durağan olmayan kaynaklarda herhangi bir ön bilgi olmaksızın optimum pişmanlık $O(\sqrt{C_\alpha T})$'ya ulaşan bir algoritma sunmaktayız. Sonuçların olası tüm gözlem dizileri için rasgele olmayacak şekilde sağlanması garanti edilmektedir. Algoritmamız T ve C_α'daki toplam sapmanın ikisininde bilinmediği bir şekilde ardışıktır. Bu performans yalnızca zaman uzunluğu T olan logaritmik doğrusal hesaplama karmaşıklığı ile elde edilmektedir.

Bölüm II'de öncelikle temel yoğunluk kestiricileri tanıtılmaktadır. Daha sonra, Bölüm III, temel yoğunluk kestirimlerinin tahminlerini birleştiren evrensel yoğunluk kestiricisini vermektedir. Bölüm IV'deki deneyler, modern yöntemlere göre önemli performans artışını göstermektedir ve bildiri Bölüm V'teki yorumlar ile sonlanmaktadır.

II. TEMEL YOĞUNLUK KESTİRİCİSİ

978-1-5090-6494-6/17/$31.00

Algorithm 1 Temel Yoğunluk Kestiricisi

Sabit değerlerin sıfırlanması $\eta \in \mathbb{R}^+$
Başlangıç parametresinin seçilmesi $\hat{\alpha}_1$
Ortalamanın hesaplanması $\mu_{\hat{\alpha}_1}$
for $t = 1$ **to** T **do**
  Kestirimin hesaplanması $\hat{\alpha}_t$
  Gözlem x_t
  Hesaplama $z_t = \mathcal{T}(x_t)$
  Parametrenin güncellenmesi: $\tilde{\alpha}_{t+1} = \hat{\alpha}_t - \eta(z_t - \mu_{\hat{\alpha}_t})$
  Dışbükey küme üzerine İzdüşüm: $\hat{\alpha}_{t+1} = P_S(\tilde{\alpha}_{t+1})$
  Ortalamanın hesaplanması $\mu_{\hat{\alpha}_{t+1}}$
end for

Bu kısımda, ilk olarak temelde var olan dizi hakkında ön bilgi ile en iyi pişmanlık değerine ulaşabilen temel yoğunluk kestiriciler oluşturulmuştur. Bu temel kestiriciler daha sonra herhangi bir ön bilgi olmadan en iyi pişmanlık değerini elde eden son algoritmayı oluşturmak için Bölüm III'te kullanılır. Burada, her t anında $x_t \in \mathbb{R}^{d_x}$ hafızasız bir üstel aile dağılımı olan $f_t(x_t) = \exp\left(-\langle \alpha_t, z_t\rangle - A(\alpha_t)\right)$ fonksiyonuna göre oluşturulur. $\alpha_t \in \mathbb{R}^d$, $D = \max_{\alpha\in S}\|\alpha\|$ olacak şekilde sınırlı dışbükey bir kümeye, S, ait olan üstel aile dağılımının doğal parametresidir. $A(\cdot)$, α_t parametresinin bir fonksiyonudur (normalizasyon faktörü), $\langle\cdot,\cdot\rangle$ iç çarpımı belirtir ve z_t, x_t'nin d-boyutlu yeterli istatistiğidir [10], yani, $z_t = \mathcal{T}(x_t)$'dir.

$f_t(x)$ dağılımını doğrudan tahmin etmek yerine, gözlemler olan $\{x_\tau\}_{\tau=1}^{t-1}$ kullanılarak her zaman t'deki doğal parametre α_t tahmin edilir ve Hannan kriterine göre tutarlı [14] pişmanlık sınırına ulaşıldığı gösterilir. Gerçek dağılımın kestirimi $\hat{f}_t(x_t) = \exp(-\langle\hat{\alpha}_t, z_t\rangle - A(\hat{\alpha}_t))$ ile verilmektedir.

Çevrimiçi meyilli azalım [13], başlangıç kestirimi $\hat{\alpha}_1$'den başlayıp gözlemlenen x_t'ye dayanarak $\hat{\alpha}_t$'yi sırayla elde etmek için kullanılmaktadır. $\hat{\alpha}_t$'yi güncellemek için öncelikle x_t'yi gözlemleyip logaritmik olan hata $l(\hat{\alpha}_t, x_t)$ kestirimimiz $\hat{\alpha}_t$ göre şu şekilde bulunur

$$l(\hat{\alpha}_t, x_t) = -\log(\hat{f}_t(x_t)) = \langle\hat{\alpha}_t, z_t\rangle + A(\hat{\alpha}_t). \quad (2)$$

Ardından hatanın $\hat{\alpha}_t$'a göre değişimi şu şekilde hesaplanır

$$\nabla_\alpha l(\hat{\alpha}_t, x_t) = z_t + \nabla_\alpha A(\hat{\alpha}_t) = z_t - \mu_{\hat{\alpha}_t}. \quad (3)$$

Burada $\mu_{\hat{\alpha}_t}$ x_t'nin $\hat{f}_t(x_t)$'e göre dağılması durumunda z_t'nin ortalamasıdır. $\hat{\alpha}_t$ parametresinin güncellenmesi aşağıdaki gibidir

$$\hat{\alpha}_{t+1} = P_S(\hat{\alpha}_t - \eta(z_t - \mu_{\hat{\alpha}_t})). \quad (4)$$

Burada $P_S(\cdot)$ sınırlı dışbükey uygun küme S'nin üzerine izdüşümü belirtir ve şu şekilde tanımlanmaktadır

$$P_S(x) = \arg\min_{y\in S}\|x - y\|. \quad (5)$$

Alg. 1'de detaylı açıklama yapılmıştır.

Daha sonra, Alg. 1'ın performans sınırları sunulmaktadır. Teorem 1 göstermektedir ki, Alg. 1 sabit öğrenme oranı ile, C_α'nın bilinmesi durumunda en iyi $O(\sqrt{C_\alpha T})$ pişmanlık değerine ulaşabilir.

Teorem 1. *Alg. 1, $f_t(x_t)$ dağılımını kestirmek için η parametresi ile kullanıldığında pişmanlık ölçütü aşağıdaki ile sınırlıdır*

$$R_T \leq \frac{1}{\eta}DC + \eta TG. \quad (6)$$

Burada $D = \max_{\alpha\in S}\|\alpha\|$, $C = 2.5D + C_\alpha$ öyle ki C_α (1)'deki gibi tanımlanır. $G = (\phi_2 + 2\phi_1 M + M^2)/2$, $M = \max_{\alpha\in S}\mu_\alpha$ ve $\phi_1 = \sum_{t=1}^T\|z_t\|/T$, $\phi_2 = \sum_{t=1}^T\|z_t\|^2/T$ olacak şekildedir.

Teorem 1'in ispatı: t zamanındaki pişmanlık şu şekilde tanımlanmıştır $r_t = l(\hat{\alpha}_t, x_t) - l(\alpha_t, x_t)$. Burada, $l(\alpha, x)$ (2)'deki gibidir. Hata fonksiyonu dışbükey olduğundan aşağıdaki eşitsizlik sağlanır

$$r_t \leq \langle\nabla_\alpha l(\hat{\alpha}_t, x_t), (\hat{\alpha}_t - \alpha_t)\rangle. \quad (7)$$

(7)'nin sağ tarafı (4)'deki güncelleme kuralı kullanılarak sınırlandırılır. (5)'deki izdüşüm tanımının kullanılması ve $\eta > 0$ olduğundan dolayı aşağıdaki eşitsizlik yazılabilir

$$\begin{aligned}&\langle\nabla_\alpha l(\hat{\alpha}_t, x_t), (\hat{\alpha}_t - \alpha_t)\rangle\\ &\leq\frac{1}{2\eta}(\|\hat{\alpha}_t\|^2 - \|\hat{\alpha}_{t+1}\|^2 - 2\langle\hat{\alpha}_t - \hat{\alpha}_{t+1}, \alpha_t\rangle) + \frac{\eta}{2}\|\nabla_\alpha l(\hat{\alpha}_t, x_t)\|^2.\end{aligned}$$

(7)'yi sol tarafta ve (3)'yi sağ tarafta kullanmak şuna yol açar

$$r_t \leq \frac{1}{2\eta}(\|\hat{\alpha}_t\|^2 - \|\hat{\alpha}_{t+1}\|^2) - \frac{1}{\eta}\langle\hat{\alpha}_t - \hat{\alpha}_{t+1}, \alpha_t\rangle + \frac{\eta}{2}\|z_t - \mu_{\hat{\alpha}_t}\|^2.$$

Bundan dolayı T zamanına kadar birikmiş pişmanlık şu şekilde ifade edilir

$$\begin{aligned}R_T \leq&\frac{1}{2\eta}(\|\hat{\alpha}_1\|^2 - \|\hat{\alpha}_{T+1}\|^2) + \frac{\eta}{2}\sum_{t=1}^T\|z_t - \mu_{\hat{\alpha}_t}\|^2\\ &- \frac{1}{\eta}(\langle\hat{\alpha}_1, \alpha_1\rangle + \sum_{t=2}^T\langle\hat{\alpha}_t, \alpha_t - \alpha_{t-1}\rangle - \langle\hat{\alpha}_{T+1}, \alpha_T\rangle),\\ \leq&\frac{1}{\eta}(2.5D^2 + DC_\alpha) + \frac{\eta T}{2}\left(\phi_2 + 2\phi_1 M + M^2\right).\end{aligned}$$

Burada M, ϕ_1 ve ϕ_2 şu şekilde verilir $M = \max_{\alpha\in S}\mu_\alpha$, $\phi_1 = \sum_{t=1}^T\|z_t\|/T$, $\phi_2 = \sum_{t=1}^T\|z_t\|^2/T$.

$G = (\phi_2 + 2\phi_1 M + M^2)/2$'nin, logaritmik hata meyili ve $C = C_\alpha + 2.5D$'nin efektif değişim parametresi ile ilgili olduğu belirtilmektedir. Bu yüzden, (6) elde edilir.

■

Teorem 1'in sonucunda, bir sonraki bölümde evrensel kestiricinin sınır değerini kanıtlamak için kullanılacak sabit öğrenme oranını kullanan bir tahmin kestiricisi elde edilir.

III. Evrensel Çevrimiçi Yoğunluk Kestirimi

Bölüm II'de, temel kestiriciler, ön bilgi kullanılarak en iyi pişmanlık değeri ile elde edilmiştir. Bu kısımda, temel kestiricilerin tahminlerini dikkatli bir şekilde oluşturulmuş öğrenme oranları ile kullanarak, ön bilgi olmadan en iyi pişmanlık değerini sağlayan evrensel bir algoritma oluşturulmaktadır.

Alg. 1, η ile birlikte kullanıldığında şu pişmanlık değerine ulaşılmış olur

$$R_T \leq \sqrt{DCGT}\left(\frac{\eta_*}{\eta} + \frac{\eta}{\eta_*}\right). \quad (8)$$

Burada $\eta_* \triangleq \sqrt{(DC)/(GT)}$'dir. Alg. 1 ile en iyi pişmanlık değerini elde etmek için η_* hakkında bir bilgiye sahip olunmalıdır. Bununla birlikte, önceden bilgi verilmeden Alg. 1 kullanarak en iyi pişmanlık değerini elde etmek mümkün değildir. Dolayısıyla, Alg. 1'i sabit bir öğrenme oranı ile kullanmak yerine, farklı öğrenme oranları η_* ile Alg. 1'ı birden çok kez çalıştırıp bunları birleştirmek en iyi pişmanlık değerini elde etmek için sahip olunması gereken η_*'ya yeterli bir dereceye kadar yaklaşılmasını sağlayacaktır.

Bu amaçla, öncelikle $r \in \{1, 2, \ldots, N\}$ için $\boldsymbol{\eta}[r] = \eta_r$ olacak şekilde N boyutunda bir parametre vektörü yaratılmaktadır. Her biri Alg. 1 η_r parametresi ile çalışacak şekilde N kestirici oluşturulmaktadır. Örnek olarak $\boldsymbol{\eta}$ parametre vektörünün $r^{nc\beta}$ elemanı gösterilebilir. N kestiricinin her biri, x_t girdisini alır ve her t anında bir $\hat{f}_t^r(x_t)$ ortaya koyar (kestirim

Algorithm 2 Evrensel Yoğunluk Kestirimi

Sabit değerlerin sıfırlanması η_r, for $r \in \{1, 2, \ldots, N\}$
Her biri Alg. 1'i η_r parametreleri ile çalıştıran N düğüm oluşturulması
ağırlıklarını başlangıç için $w_1^r = 1/N$
for $t = 1$ **to** T **do**
  Kestirim yapma $\hat{f}_t^u(x) = \sum_{r=1}^N w_t^r \hat{f}_t^r(x)$
  Gözlem x_t
  Hesaplama $z_t = \mathcal{T}(x_t)$
  for $r = 1$ **to** N **do**
    Alg. 1'e göre parametrelerin güncellenmesi $\hat{\alpha}_t^r$
    $w_{t+1}^r = w_t^r \hat{f}_t^r(x_t)/\hat{f}_t^u(x_t)$
  end for
end for

süreci). Daha sonra, tüm kestiricilerin çıktıları ağırlıklı bir kombinasyon alınarak aşağıdaki gibi birleştirilir

$$\hat{f}_t^u(x_t) = \sum_{r=1}^{N} w_t^r \hat{f}_t^r(x_t). \tag{9}$$

Burada w_t^r, t zamanındaki $r^{nc\text{ß}}$ kestiricisinin tahmininin ağırlığıdır. Başlangıçta bütün kestirici çıktılarına eşit ağırlıklar atanır, bu yüzden ilk başta kombinasyon ağırlıkları $w_1^r = 1/N$ olarak gösterilebilir. Daha sonra her t anında ağırlıklar aşağıdaki kurala göre güncellenir

$$w_{t+1}^r = w_t^r \hat{f}_t^r(x_t)/\hat{f}_t^u(x_t). \tag{10}$$

Burada $\hat{f}_t^u(x_t)$ düzgeleyici olarak kullanılır. Alg. 2'de evrensel algoritmanın tam bir tanımı yapılmıştır. Daha sonra, evrensel yoğunluk kestiricisinin, yani Alg. 2'nin performans sınırları verilmektedir. Teorem 2 ve Sonuç 1'in çıktıları C hakkında daha önce herhangi bir bilgi olmadan en iyi pişmanlık değerine bağlı $O(\sqrt{CT})$ elde edildiğini göstermektedir.

Teorem 2. *Alg. 2 şu pişmanlık sınırına sahiptir*

$$R_T \leq \log(N) + \sqrt{DCGT}\left[\min_{i\in\{1,2,\ldots,N\}}\left(\frac{\eta_*}{\eta_i} + \frac{\eta_i}{\eta_*}\right)\right].$$

Burada $D = \max_{\alpha\in S}\|\alpha\|$, $C = 2.5D + C_\alpha$ *öyle ki* C_α (1)*'deki gibi tanımlanmıştır.* $G = (\phi_2 + 2\phi_1 M + M^2)/2$ *öyle ki* $M = \max_{\alpha\in S}\mu_\alpha$, $\phi_1 = \sum_{t=1}^T \|z_t\|/T$, $\phi_2 = \sum_{t=1}^T \|z_t\|^2/T$, $\eta_* = \sqrt{DCGT}$ *ve* $i \in \{1, 2, \ldots, N\}$ *için* η_i *uzmanlar tarafından kullanılan parametrelerdir.*

Teorem 2'in ispatı: t zamanındaki pişmanlık değeri şu şekilde verilmiştir

$$r_t = -\log(\hat{f}_t^u(x_t)) + \log(f_t(x_t)). \tag{11}$$

(11)'i $t = 1$'den T'ye kadar toplayarak ve (9)'u kullanarak şuna ulaşılabilir

$$R_T = -\log(\prod_{t=1}^{T}(\sum_{r=1}^{N} w_t^r \hat{f}_t^r(x_t))) + \sum_{t=1}^{T}\log(f_t(x_t)). \tag{12}$$

(10)'dan ağırlıkların şu şekilde verildiği çıkarılabilir

$$w_t^r = \frac{\prod_{\tau=1}^{t-1}\hat{f}_\tau^r(x_\tau)}{\sum_{r=1}^{N}\prod_{\tau=1}^{t-1}\hat{f}_\tau^r(x_\tau)}. \tag{13}$$

(13)'ü (12)'nin içinde kullanmak şunu verir,

$$\begin{aligned} R_T &= -\log(\sum_{r=1}^{N}\prod_{\tau=1}^{T}\hat{f}_t^r(x_t)) + \log(N) + \sum_{t=1}^{T}\log(f_t(x_t)) \\ &\leq \log(N) - \max_r(\sum_{t=1}^{T}\log(\hat{f}_t^r(x_t))) + \sum_{t=1}^{T}\log(f_t(x_t)) \qquad (14) \\ &\leq \log(N) + \sqrt{DCGT}\left[\min_{i\in\{1,2,\ldots,N\}}\left(\frac{\eta_*}{\eta_i} + \frac{\eta_i}{\eta_*}\right)\right] \qquad (15) \end{aligned}$$

ve ispat tamamlanmış olur. ■

Teorem 2'in sonucu, sınırlamanın algoritmada kullanılan öğrenme oranları kümesine bağlı olduğunu göstermektedir. Sonuç 1'de bu çıktıyı kullanılarak logaritmik doğrusal karmaşıklıkla en iyi pişmanlık sınırlarının elde edilebildiği gösterilmektedir.

Sonuç 1. *Kestiricilerin çalıştırılması için* η' *ve* η'' *parametre aralığı seçilmiş olsun.* $K = \eta''/\eta'$ *ve* $N = \lceil \log_2 K \rceil + 1$ *olarak gösterilir. Ardından Alg. 2'nin* $\eta_i = 2^{i-1}\eta'$ *vektör parametresi ile* $i \in \{1, 2, \ldots, N\}$ *için çalıştırılması* η_**'ın değişik değerleri için şu pişmanlık sınırını vermektedir.*

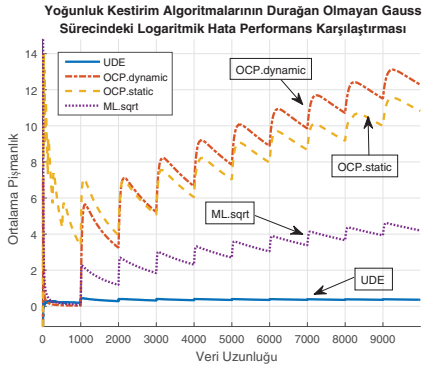
1) $\eta' \leq \eta_* \leq \eta''$ *durumunda:*
$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + \frac{3\sqrt{2}}{2}\sqrt{DCGT}.$$
Çünkü eğer η_* *belirli a için* $\eta_* = 2^a\sqrt{2}$ *halindeyse* $(\eta_*/\eta_i + \eta_i/\eta_*)$ *en büyüktür.*
2) $\eta_* \geq \eta''$ *durumunda*
$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + (1 + \frac{\eta_*}{\eta''})\sqrt{DCGT}.$$
$\eta_* \leq \sqrt{(4 + 1/T)D^2M^{-2}}$ *olduğundan,* η''*'yı* $\eta'' \geq \sqrt{(4 + T^{-1})D^2M^{-2}}$ *olarak ayarlayarak bu madde geçersiz kılınabilir.*
3) $\eta_* \leq \eta'$ *durumunda*
$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + (1 + \frac{\eta'}{\eta_*})\sqrt{DCGT}.$$
$\eta_* \geq \sqrt{2.5D^2(TG)^{-1}}$ *olduğundan,* η'*'yı* $\eta' \leq \sqrt{2.5D^2T^{-1}}$ *olarak ayarlamak şunu verir*
$$R_T \leq \log(\lceil \log_2 \eta''/\eta' \rceil + 1) + (1 + \sqrt{G})\sqrt{DCGT}.$$

Dolayısıyla Alg. 2'yi uygun bir parametre vektörü ile çalıştırarak, η' ve η'' arasındaki ayrım temelde $2.5D \leq C \leq (2T + 0.5)D$ ile sınırlanan C'ye bağımlı olduğundan $O(\log T)$ hesaplama karmaşıklığı ile $O(\sqrt{CT})$ pişmanlık değeri elde edilmektedir. Bilinmeyen T için, ikiye katlama numarasını kullanılarak $O(\sqrt{CT})$ pişmanlık değeri elde edilmektedir.

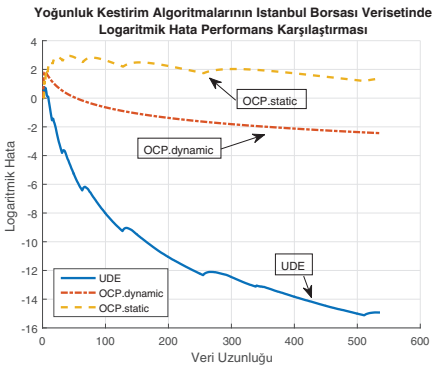
IV. Deneyler

Bu bölümde, gerçek veriler ve yapay olarak oluşturulmuş veriler üzerinde algoritmamızın performansı gösterilmekte ve modern yöntemlerle [6], [13] performans karşılaştırılması verilmektedir. Algoritma sabit adım büyüklüğü ile çevrimiçi dışbükey programlama kullandığından, [13]'te kullanılan tekniği OCP.static ile ifade edebiliriz. [15]'teki algoritmayı ise OCP.dynamic olarak ifade edebiliriz. Çünkü algoritma her turda dinamik olarak değişen bir adım büyüklüğü kullanmaktadır. Ayrıca, algoritmamız yaygın olarak kullanılan En Çok Olabilirlik (ML) kestiriminin çevrimiçi sürümü ile de karşılaştırılmaktadır [15]. Algoritma T uzunluğunda işleme için $1/T \leq \eta \leq T$ aralığındaki parametrelerle çalıştırılır. Tüm algoritmalar olduğu gibi kullanılmıştır.

İlk olarak birim standart sapma, $\sigma = 1$, ile tek değişkenli bir Gauss sürecinden 10000 büyüklüğünde bir veri kümesi oluşturulmaktadır ve her 1000 örnekte ortalama 10 ile -10 arasında değişen değerler almaktadır. Veri kümesi durağan olmadığından ML kestiricisi çok düşük performans göstermektedir. Bu nedenle, adil bir performans karşılaştırması için ML.sqrt adında yeni bir ML algoritması oluşturulmuştur. Bu, her t turda son $\sqrt{t}$ örneği kullanmaktadır. Şekil 1'de, bu dört algoritmanın pişmanlık performansları gösterilmektedir. OCP.dynamic ilk 1000 örnekte son derece hızlı bir şekilde yakınsama gösterse de, OCP.static ilk kısımda bu kadar hızlı yakınsama gösteremez çünkü algoritma, 2'nin her bir



Şekil 1: Durağan olmayan Gauss süreci üzerindeki yoğunluk kestirim algoritmalarının ortalama pişmanlık performansı.



Şekil 2: Yoğunluk kestirim algoritmalarının İMKB veri kümesindeki hata performansı.

kuvvetinde kendini baştan ayarlar (ikiye katlama numarası) ve öğrenme oranı yeterince büyük değildir. Bununla birlikte, yaklaşık 2000. örnek civarından sonra, OCP.static OCP.dynamic'i geçmekte ve kalan turlarda sürekli olarak daha iyi performans sergilemektedir. Yine de, her ikisi de her turda daha iyi performans gösteren ML.sqrt'den performans olarak daha aşağıda kalmaktadır. Bununla birlikte, bu algoritmaların hepsi hala veri istatistiklerindeki değişikliklere karşı çok hassastır ve ortalama pişmanlık değerleri her değişiklikte yani her 1000 örnekte yukarı doğru çıkma eğilimi göstermektedir. Dahası, üç algoritmanın ortalama pişmanlık değerleri veri uzunluğuna göre yarı doğrusal bir artışa sahiptir. Bununla birlikte, Evrensel Yoğunluk Kestiricisinin (UDE) böyle bir problemi bulunmamaktadır. UDE özenle oluşturulmuş yoğunluk kestiricilerinin bir karışımını kullanır ve öğrenme oranları üzerindeki evrensellik bakımından güçlüdür. Dolayısıyla, veri istatistiktiğindeki değişikliklerin UDE'nin pişmanlık değerleri üzerindeki etkisi yok denecek kadar azdır. UDE, OCP.dynamic, OCP.static ve ML.sqrt'ten önemli oranda daha iyi performans göstermektedir.

Gerçek veri karşılaştırması için İstanbul Menkul Kıymetler Borsası (İMKB) [16] veri kümesi kullanılmaktadır. İMKB veri seti için durağan olmayan çok değişkenli bir Gauss süreci varsayılarak bu dağılımın tahmin edilmesi için algoritmalar çalıştırılmıştır. Gerçek dağılım bilinmediğinden, algoritma performanslarını karşılaştırırken pişmanlıkları yerine logaritmik hataları kullanılmıştır. Şekil 2'de, UDE, OCP.static ve OCP.dynamic'in logaritmik hata performansları gösterilmiştir. ML ve ML.sqrt göz ardı edilmiştir çünkü bu iki algoritma da iyi çalışmamaktadır. Veri kümesi küçük olduğundan (yalnızca 536 örnek) yeniden ayarlama özelliği nedeniyle OCP.static kötü performans göstermektedir. OCP.static'in hızı verinin durağan olmamasına yetişemediğinden OCP.static yeterince hızlı yakınsayamamakta ve bu nedenle başarılı bir yoğunluk kestirimi üretememektedir. Öte yandan, OCP.dynamic veri dizisi boyunca iyi bir yakınsama göstermekte, ancak 300. örnekten sonra yakınsama hızı azalmaktadır. Bununla birlikte, dikkatle yapılandırılmış öğrenme oranlarının bir karışımını kullanan UDE, diğer yöntemlerden sürekli olarak daha iyi performans sergilemektedir. Dolayısıyla, UDE hızlı bir yakınsamaya sahiptir.

V. Sonuç

Belirsiz bir üstel aile kaynağından üretilen yoğunluk fonksiyonunu, C bilgisi olmadan en iyi pişmanlık katsayısı $\sqrt{CT}$ ile tanımlayan, gerçekten ardışık bir algoritma sunulmuştur. Sonuçların, muhtemel tüm gözlem dizileri için kesin bir anlamda sağlanması garanti edilmektedir. Farklı kestiricilerin titizlikle tasarlanıp bunların evrensel olarak birleştirilmesi, zamanda yalnızca logaritmik karmaşıklıkla en iyi sonucun elde edilebilmesini sağlamaktadır. Bu yüzden, algoritma büyük veri içeren uygulamalarda etkin bir şekilde kullanılabilir.

Kaynakça

[1] Y. Nakamura and O. Hasegawa, "Nonparametric density estimation based on self-organizing incremental neural network for large noisy data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–10, 2016.

[2] A. Penalver and F. Escolano, "Entropy-based incremental variational bayes learning of gaussian mixtures," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 534–540, March 2012.

[3] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 576–588, March 2015.

[4] E. Müller, I. Assent, R. Krieger, S. Günnemann, and T. Seidl, "Densest: Density estimation for data mining in high dimensional spaces," in *Proc. SIAM International Conference on Data Mining (SDM 2009), Sparks, Nevada, USA*. SIAM, 2009, pp. 173–184.

[5] Y. Cao, H. He, and H. Man, "Somke: Kernel density estimation over data streams by sequences of self-organizing maps," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1254–1268, Aug 2012.

[6] M. Raginsky, R. M. Willett, C. Horn, J. Silva, and R. F. Marcia, "Sequential anomaly detection in the presence of noise and limited feedback," *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5544–5562, Aug 2012.

[7] K. B. Dyer, R. Capo, and R. Polikar, "Compose: A semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 12–26, Jan 2014.

[8] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[9] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, May 1997.

[10] B. O. Koopman, "On distributions admitting a sufficient statistic," *Transactions of the American Mathematical Society*, vol. 39, no. 3, pp. 399–409, 1936.

[11] A. R. Barron and C.-H. Sheu, "Approximation of density functions by sequences of exponential families," *Ann. Statist.*, vol. 19, no. 3, pp. 1347–1369, 09 1991.

[12] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[13] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent." in *ICML*, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003, pp. 928–936.

[14] S. Hart and A. Mas-Colell, "A general class of adaptive strategies," *Journal of Economic Theory*, vol. 98, no. 1, pp. 26 – 54, 2001.

[15] I. J. Myung, "Tutorial on maximum likelihood estimation," *J. Math. Psychol.*, vol. 47, no. 1, pp. 90–100, Feb. 2003.

[16] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid rbf neural networks model as a forecaster," *Statistics and Computing*, vol. 24, no. 3, pp. 365–375, 2014.