Refleks Temelli Modeller El Kitabı VIP

Afshine Amidi ve Shervine Amidi

September 14, 2019

Yavuz Kömeçoğlu ve Ayyüce Kızrak tarafından çevrilmiştir

Doğrusal öngörücüler

Bu bölümde, girdi-çıktı çiftleri olan örneklerden geçerek, deneyim ile gelişebilecek refleks-temelli modelleri göreceğiz.

❒ **Öznitelik vektörü** – x girişinin öznitelik vektörü $\phi(x)$ olarak not edilir ve şöyledir:

$$\boxed{\phi(x)=\left[\begin{array}{c}\phi_1(x)\\\vdots\\\phi_d(x)\end{array}\right]\in\mathbb{R}^d}$$

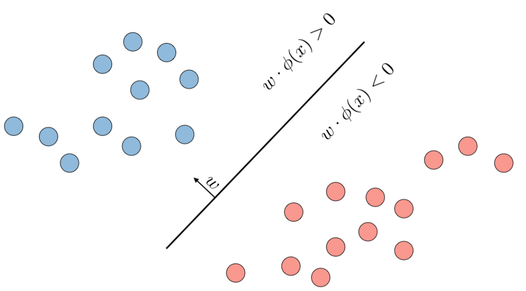
❒ **Puan** – Bir örneğin $s(x,w)$ si ni $(\phi(x),y)\in\mathbb{R}^d\times\mathbb{R}$, $w\in\mathbb{R}^d$ doğrusal ağırlık modeline bağlı olarak:

$$\boxed{s(x,w)=w\cdot\phi(x)}$$

Sınıflandırma

❒ **Doğrusal sınıflandırıcı** – Bir ağırlık vektörü $w\in\mathbb{R}^d$ ve bir öznitelik vektörü $\phi(x)\in\mathbb{R}^d$ verildiğinde, ikili doğrusal sınıflandırıcı f_w şöyle verilir:

$$\boxed{f_w(x)=\textrm{sign}(s(x,w))=\left\{\begin{array}{cl}+1 & \textrm{ise } w\cdot\phi(x)>0\\-1 & \textrm{ise } w\cdot\phi(x)<0\\? & \textrm{ise } w\cdot\phi(x)=0\end{array}\right.}$$



❒ **Marj** – $(\phi(x),y)\in\mathbb{R}^d\times\{-1,+1\}$ örneğinin $m(x,y,w)\in\mathbb{R}$ marjları $w\in\mathbb{R}^d$ doğrusal ağırlık modeliyle ilişkili olarak, tahminin güvenirliği ölçülür: daha büyük değerler daha iyidir. Şöyle ifade edilir:

$$\boxed{m(x,y,w)=s(x,w)\times y}$$

Bağlanım

❒ **Doğrusal bağlanım** – $w\in\mathbb{R}^d$ bir ağırlık vektörü ve bir öznitelik vektörü $\phi(x)\in\mathbb{R}^d$ verildiğinde, f_w olarak belirtilen ağırlıkların doğrusal bir bağlanım (linear regression) çıktısı şöyle verilir:

$$\boxed{f_w(x)=s(x,w)}$$

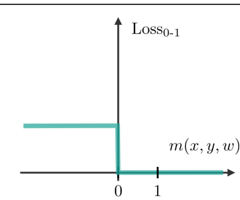
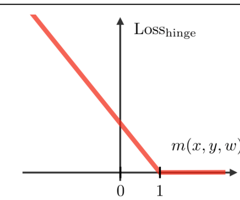
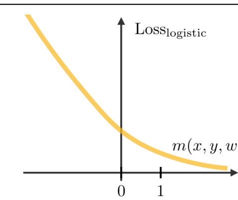
❒ **Artık** – Artık (residual) $\textrm{res}(x,y,w)\in\mathbb{R}$, $f_w(x)$ tahmininin y hedefini aştığı miktar olarak tanımlanır:

$$\boxed{\textrm{res}(x,y,w)=f_w(x)-y}$$

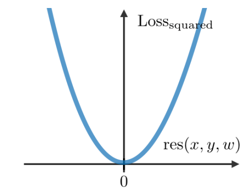
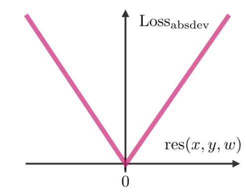
Kayıp minimizasyonu

❒ **Kayıp fonksiyonu** – Kayıp fonksiyonu $\textrm{Loss}(x,y,w)$, x girişinden y çıktısının öngörme görevindeki model ağırlıkları ile ne kadar mutsuz olduğumuzu belirler. Bu değer eğitim sürecinde en aza indirmek istediğimiz bir miktar.

❒ **Sınıflandırma durumu** – Doğru etiket $y\in\{-1,+1\}$ değerinin x örneğinin doğrusal ağırlık w modeliyle sınıflandırılması $f_w(x)\triangleq\textrm{sign}(s(x,w))$ belirleyicisi ile yapılabilir. Bu durumda, sınıflandırma kalitesini ölçen bir fayda ölçütü $m(x,y,w)$ marjı ile verilir ve aşağıdaki kayıp fonksiyonlarıyla birlikte kullanılabilir:

Ad	Sıfır-bir kayıp	Menteşe kaybı	Lojistik kaybı
$\textrm{Loss}(x,y,w)$	$1_{\{m(x,y,w)\leqslant0\}}$	$\max(1-m(x,y,w),0)$	$\log(1+e^{-m(x,y,w)})$
Örnekleme			

❒ **Regresyon durumu** – Doğru etiket $y\in\mathbb{R}$ değerinin x örneğinin bir doğrusal ağırlık modeli w ile öngörülmesi $f_w(x)\triangleq s(x,w)$ öngörüsü ile yapılabilir. Bu durumda, regresyonun kalitesini ölçen bir fayda ölçütü $\textrm{res}(x,y,w)$ marjı ile verilir ve aşağıdaki kayıp fonksiyonlarıyla birlikte kullanılabilir:

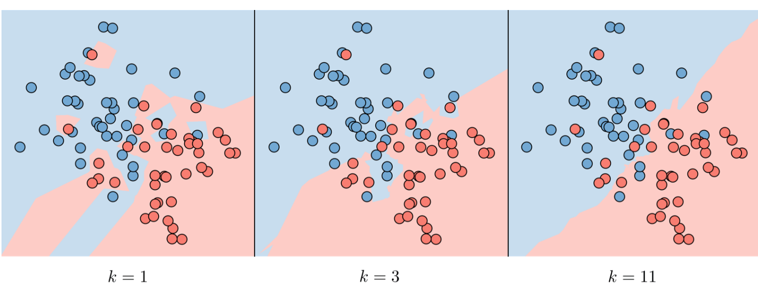
Ad	Kareler kaybı	Mutlak sapma kaybı
Loss(x,y,w)	$(\text{res}(x,y,w))^2$	$\lvert\text{res}(x,y,w)\rvert$
Görselleştirme		

❒ **Kayıp minimize etme çerçevesi** – Bir modeli eğitmek için, eğitim kaybını en aza indirmek istiyoruz:

$$\boxed{\text{TrainLoss}(w)=\frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{(x,y)\in\mathcal{D}_{\text{train}}}\text{Loss}(x,y,w)}$$

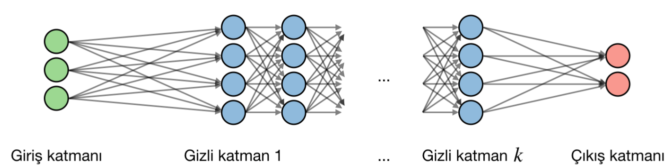
Doğrusal olmayan öngörücüler

❒ ***k*-en yakın komşu** – Yaygın olarak k-NN (k-nearest neighbors) olarak bilinen k-en yakın komşu algoritması, bir veri noktasının tepkisinin eğitim kümesinden k komşularının yapısı tarafından belirlendiği parametrik olmayan bir yaklaşımdır. Hem sınıflandırma hem de regresyon ayarlarında kullanılabilir.



Not: k parametresi ne kadar yüksekse, önyargı (bias) o kadar yüksek ve k parametresi ne kadar düşükse, varyans o kadar yüksek olur.

❒ **Yapay sinir ağları** – Neural networks are a class of models that are built with layers. Commonly used types of neural networks include convolutional and recurrent neural networks. The vocabulary around neural networks architectures is described in the figure below:



i ağın i. katmanı ve j, katmanın j. gizli birimi olacak şekilde aşağıdaki gibi ifade edilir:

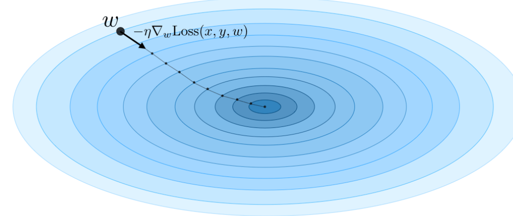
$$\boxed{z_j^{[i]}={w_j^{[i]}}^Tx+b_j^{[i]}}$$

w, b, x, z değerlerinin sırasıyla nöronun ağırlık, önyargı (bias), girdi ve aktive edilmemiş çıkışını olarak ifade eder.

Stokastik gradyan inişi

❒ **Gradyan inişi** – $\eta\in\mathbb{R}$ öğrenme oranını (aynı zamanda adım boyutu olarak da bilinir) dikkate alınarak, gradyan (bayır) inişine ilişkin güncelleme kuralı, öğrenme oranı ve Loss(x,y,w) kayıp fonksiyonu ile aşağıdaki şekilde ifade edilir:

$$\boxed{w\longleftarrow w-\eta\nabla_w\text{Loss}(x,y,w)}$$



❒ **Stokastik güncellemeler** – Stokastik gradyan inişi (SGİ/SGD, stochastic gradient descent), bir seferde bir eğitim örneğinin $(\phi(x),y)\in\mathcal{D}_{\text{train}}$ parametrelerini günceller. Bu yöntem bazen gürültülü, ancak hızlı güncellemeler yol açar.

❒ **Yığın güncellemeler** – Yığın gradyan inişi (YGİ/BGD, batch gradient descent), bir seferde bir grup örnek (örneğin, tüm eğitim kümesi) parametrelerini günceller. Bu yöntem daha yüksek bir hesaplama maliyetiyle kararlı güncelleme talimatlarını hesaplar.

İnce ayar modelleri

❒ **Hipotez sınıfı** – Bir hipotez sınıfı $\mathcal{F}$, sabit bir $\phi(x)$ ve değişken w ile olası öngörücü kümesidir:

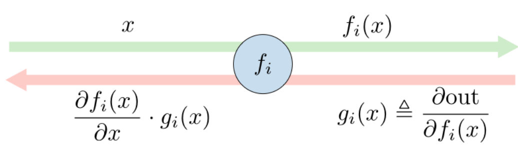
$$\boxed{\mathcal{F}=\left\{f_w:w\in\mathbb{R}^d\right\}}$$

❒ **Lojistik fonksiyon** – Ayrıca sigmoid fonksiyon olarak da adlandırılan lojistik fonksiyon (sigmoid function) σ, şöyle tanımlanır:

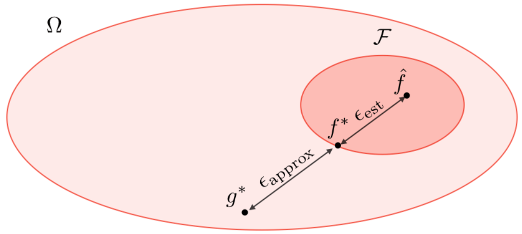
$$\boxed{\forall z\in]-\infty,+\infty[,\quad\sigma(z)=\frac{1}{1+e^{-z}}}$$

Not: $\sigma'(z)=\sigma(z)(1-\sigma(z))$ şeklinde ifade edilir.

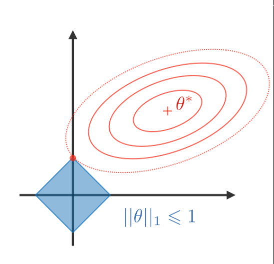
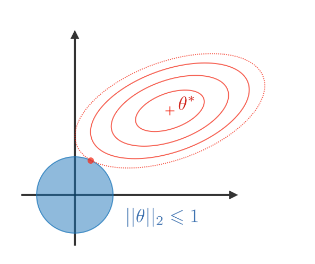
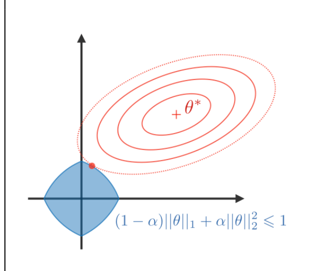
❒ **Geri yayılım** – İleriye geçiş, i'de yer alan alt ifadenin değeri olan fi ile yapılırken, geriye doğru geçiş $g_i=\frac{\partial\text{out}}{\partial f_i}$ aracılığıyla yapılır ve f_i'nin çıkışı nasıl etkilediğini gösterir.



❐ **Yaklaşım ve kestirim hatası** – Yaklaşım hatası ϵ_{approx}, $\mathcal{F}$ tüm hipotez sınıfının hedef öngörücü g^* ne kadar uzak olduğunu gösterirken, kestirim hatası ϵ_{est} öngörücüsü $\hat{f}$, $\mathcal{F}$ hipotez sınıfının en iyi yordayıcısı f^*'ya göre ne kadar iyi olduğunu gösterir.



❐ **Düzenlileştirme** – Düzenlileştirme (regularization) prosedürü, modelin verilerin aşırı öğrenmesinden kaçınmayı amaçlar ve böylece yüksek değişkenlik sorunlarıyla ilgilenir. Aşağıdaki tablo, yaygın olarak kullanılan düzenlileştirme tekniklerinin farklı türlerini özetlemektedir:

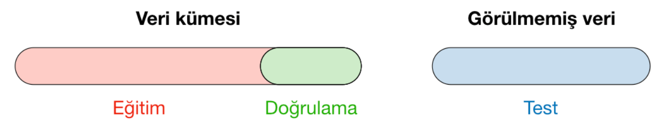
LASSO	Ridge	Elastic Net
- Katsayıları 0'a düşürür - Değişken seçimi için iyi	Katsayıları daha küçük yapar	Değişken seçimi ile küçük katsayılar arasında ödünleşim
		
$... + \lambda\|\|\theta\|\|_1$ $\lambda \in \mathbb{R}$	$... + \lambda\|\|\theta\|\|_2^2$ $\lambda \in \mathbb{R}$	$... + \lambda\Big[(1-\alpha)\|\|\theta\|\|_1 + \alpha\|\|\theta\|\|_2^2\Big]$ $\lambda \in \mathbb{R}, \quad \alpha \in [0,1]$

❐ **Hiperparametreler** – Hiperparametreler öğrenme algoritmasının özellikleridir ve öznitelikler dahildir, λ normalizasyon parametresi, yineleme sayısı T, adım büyüklüğü η, vb.

❐ **Kümeler** – Bir model seçerken, veriyi aşağıdaki gibi 3 farklı parçaya ayırırız:

Eğitim kümesi	Doğrulama kümesi	Test kümesi
- Model eğitilir - Veri kümesinin genellikle %80'i	- Model değerlendirilir - Veri kümesinin genellikle %20'si - Ayrıca tutma veya geliştirme kümesi olarak da adlandırılır	- Model tahminlerini verir - Görünmeyen veriler

Model seçildikten sonra, tüm veri kümesi üzerinde eğitilir ve görünmeyen test kümesinde test edilir. Bunlar aşağıdaki şekilde gösterilmektedir:



Gözetimsiz Öğrenme

Gözetimsiz öğrenme yöntemlerinin sınıfı, zengin gizli yapılara sahip olabilecek verilerin yapısını keşfetmeyi amaçlamaktadır.

k-ortalama

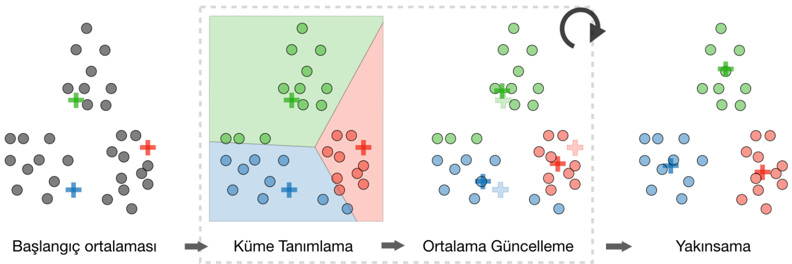
❐ **Kümeleme** – $\mathcal{D}_{\text{train}}$ giriş noktalarından oluşan bir eğitim kümesi göz önüne alındığında, kümeleme (clustering) algoritmasının amacı, her bir $\phi(x_i)$ noktasını $z_i \in \{1,...,k\}$ kümesine atamaktır.

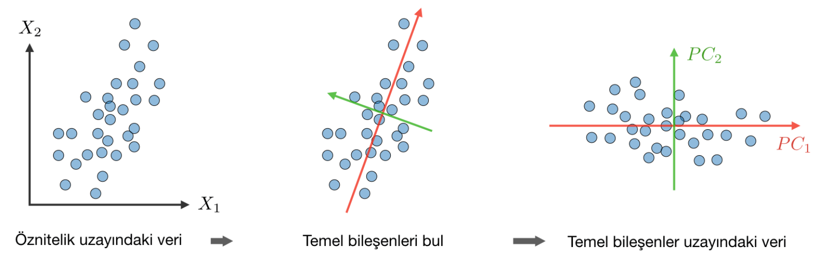
❐ **Amaç fonksiyonu** – Ana kümeleme algoritmalarından biri olan k-ortalama için kayıp fonksiyonu şöyle ifade edilir:

$$\boxed{\text{Loss}_{\text{k-means}}(x,\mu) = \sum_{i=1}^{n} ||\phi(x_i) - \mu_{z_i}||^2}$$

❐ **Algoritma** – Küme merkezlerini $\mu_1,\mu_2,...,\mu_k \in \mathbb{R}^n$ kümesini rasgele başlattıktan sonra, k-ortalama algoritması yakınsayana kadar aşağıdaki adımı tekrarlar:

$$\boxed{z_i = \underset{j}{\text{arg min}}||\phi(x_i) - \mu_j||^2} \quad \text{and} \quad \boxed{\mu_j = \frac{\displaystyle\sum_{i=1}^{m} 1_{\{z_i=j\}}\phi(x_i)}{\displaystyle\sum_{i=1}^{m} 1_{\{z_i=j\}}}}$$





Temel Bileşenler Analizi

❒ **Özdeğer, özvektör** – Bir $A\in\mathbb{R}^{n\times n}$ matrisi verildiğinde, $z\in\mathbb{R}^n\backslash\{0\}$ olacak şekilde bir vektör varsa λ, A'nın bir öz değeri olduğu söylenir, aşağıdaki gibi ifade edilir:

$$\boxed{Az=\lambda z}$$

❒ **Spektral teoremi** – $A\in\mathbb{R}^{n\times n}$ olsun. A simetrik ise, o zaman A gerçek ortogonal matris $U\in\mathbb{R}^{n\times n}$ olacak şekilde köşegenleştirilebilir. $\Lambda=\textrm{diag}(\lambda_1,...,\lambda_n)$ formülü dikkate alınarak aşağıdaki gibi ifade edilir:

$$\boxed{\exists\Lambda\textrm{ diagonal},\quad A=U\Lambda U^T}$$

Not: en büyük özdeğerle ilişkilendirilen özvektör, A matrisinin temel özvektörüdür.

❒ **Algoritma** – Temel Bileşenler Analizi (PCA, principal component analysis) prosedürü, verilerin varyansını en üst düzeye çıkararak k boyutlarına indirgeyen bir boyut küçültme tekniğidir:

- Adım 1: Verileri ortalama 0 ve 1 standart sapma olacak şekilde normalize edin.

$$\boxed{x_j^{(i)}\leftarrow\frac{x_j^{(i)}-\mu_j}{\sigma_j}}\quad\textrm{where}\quad\boxed{\mu_j = \frac{1}{m}\sum_{i=1}^mx_j^{(i)}}\quad\textrm{and}\quad\boxed{\sigma_j^2=\frac{1}{m}\sum_{i=1}^m(x_j^{(i)}-\mu_j)^2}$$

- Adım 2: Hesaplama $\displaystyle\Sigma=\frac{1}{m}\sum_{i=1}^m\phi(x_i)\phi(x_i)^T\in\mathbb{R}^{n\times n}$, ki bu, gerçek özdeğerlerle simetriktir.

- Adım 3: Hesaplama $u_1, ..., u_k\in\mathbb{R}^n$ k'nin ortogonal ana özvektörleri, yani k en büyük özdeğerlerin ortogonal özvektörleri.

- Adım 4: $\textrm{span}_\mathbb{R}(u_1,...,u_k)$'daki verilerin izdüşümünü al. Bu prosedür, tüm k boyutlu uzaylar arasındaki farkı en üst düzeye çıkarır.