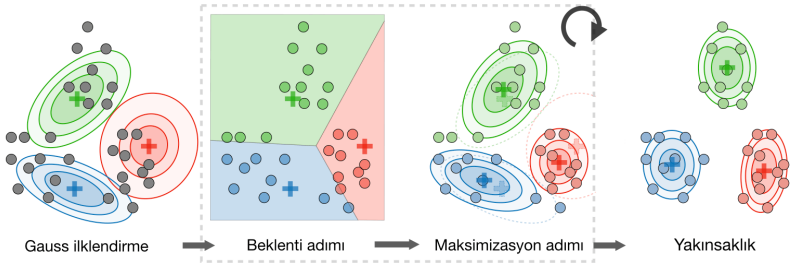
# Gözetimsiz Öğrenme El Kitabı VIP

Afshine Amidi ve Shervine Amidi

April 30, 2019

*Yavuz Kömeçoğlu ve Başak Buluz tarafından çevrilmiştir*




## Gözetimsiz Öğrenmeye Giriş

❒ **Motivasyon** – Gözetimsiz öğrenmenin amacı etiketlenmemiş verilerdeki gizli örüntüleri bulmaktır $\{x^{(1)},...,x^{(m)}\}$.

❒ **Jensen eşitsizliği** – $f$ bir konveks fonksiyon ve $X$ bir rastgele değişken olsun. Aşağıdaki eşitsizliklerimiz:

$$\boxed{E[f(X)] \geqslant f(E[X])}$$

## Beklenti-Ençoklama (Maksimizasyon)

❒ **Gizli değişkenler** – Gizli değişkenler, tahmin problemlerini zorlaştıran ve çoğunlukla $z$ olarak adlandırılan gizli/gözlemlenmemiş değişkenlerdir. Gizli değişkenlerin bulunduğu yerlerdeki en yaygın ayarlar şöyledir:

| Yöntem | Gizli değişken $z$ | $x\|z$ | Açıklamalar |
|---|---|---|---|
| $k$ Gaussianların birleşimi | Multinomial($\phi$) | $\mathcal{N}(\mu_j,\Sigma_j)$ | $\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$ |
| Faktör analizi | $\mathcal{N}(0,I)$ | $\mathcal{N}(\mu+\Lambda z,\psi)$ | $\mu_j \in \mathbb{R}^n$ |

❒ **Algoritma** – Beklenti-Ençoklama (Maksimizasyon) (BE) algoritması, $\theta$ parametresinin maksimum olabilirlik kestirimiyle tahmin edilmesinde, olasılığa ard arda alt sınırlar oluşturan (E-adımı) ve bu alt sınırın (M-adımı) aşağıdaki gibi optimize edildiği etkin bir yöntem sunar:

- E-adımı: Her bir veri noktasının $x^{(i)}$'in belirli bir kümeden $z^{(i)}$ geldiğinin sonsal olasılık değerinin $Q_i(z^{(i)})$ hesaplanması aşağıdaki gibidir:

$$\boxed{Q_i(z^{(i)}) = P(z^{(i)}|x^{(i)};\theta)}$$

- M-adımı: Her bir küme modelini ayrı ayrı yeniden tahmin etmek için $x^{(i)}$ veri noktalarındaki kümeye özgü ağırlıklar olarak $Q_i(z^{(i)})$ sonsal olasılıklarının kullanımı aşağıdaki gibidir:

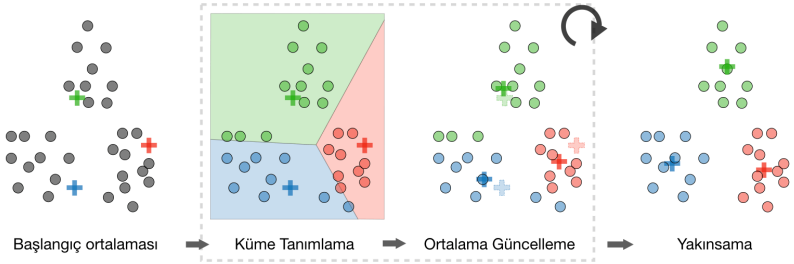
$$\boxed{\theta_i = \underset{\theta}{\text{argmax}} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log\left(\frac{P(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\right) dz^{(i)}}$$

## $k$-ortalamalar ($k$-means) kümeleme

$c^{(i)}$, $i$ veri noktasının bulunduğu küme olmak üzere, $\mu_j$ $j$ kümesinin merkez noktasıdır.

❒ **Algoritma** – Küme ortalamaları $\mu_1,\mu_2,...,\mu_k \in \mathbb{R}^n$ rasgele olarak başlatıldıktan sonra, $k$-ortalamalar algoritması yakınsayana kadar aşağıdaki adımı tekrar eder:

$$\boxed{c^{(i)} = \arg\min_j ||x^{(i)} - \mu_j||^2} \quad \text{and} \quad \boxed{\mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}}$$




❒ **Bozulma fonksiyonu** – Algoritmanın yakınsadığını görmek için aşağıdaki gibi tanımlanan bozulma fonksiyonuna bakarız:

$$\boxed{J(c,\mu) = \sum_{i=1}^m ||x^{(i)} - \mu_{c^{(i)}}||^2}$$

## Hiyerarşik kümeleme

❒ **Algoritma** – Ardışık olarak iç içe geçmiş kümelerden oluşturan hiyerarşik bir yaklaşıma sahip bir kümeleme algoritmasıdır.

❒ **Türler** – Aşağıdaki tabloda özetlenen farklı amaç fonksiyonlarını optimize etmeyi amaçlayan farklı hiyerarşik kümeleme algoritmaları vardır:

| Ward bağlantı | Ortalama bağlantı | Tam bağlantı |
|---|---|---|
| Küme mesafesi içinde minimize edin | Küme çiftleri arasındaki ortalama uzaklığı en aza indirin | Küme çiftleri arasındaki maksimum uzaklığı en aza indirin |

### Kümeleme değerlendirme metrikleri

Gözetimsiz bir öğrenme ortamında, bir modelin performansını değerlendirmek çoğu zaman zordur, çünkü gözetimli öğrenme ortamında olduğu gibi, gerçek referans etiketlere sahip değiliz.

❒ **Siluet katsayısı** – Bir örnek ile aynı sınıftaki diğer tüm noktalar arasındaki ortalama mesafeyi ve bir örnek ile bir sonraki en yakın kümedeki diğer tüm noktalar arasındaki ortalama mesafeyi not ederek, tek bir örnek için siluet katsayısı aşağıdaki gibi tanımlanır:

$$\boxed{s = \frac{b-a}{\max(a,b)}}$$

❒ **Calinski-Harabaz indeksi** – $k$ kümelerin sayısını belirtmek üzere $B_k$ ve $W_k$ sırasıyla, kümeler arası ve küme içi dağılım matrisleri olarak aşağıdaki gibi tanımlanır

$$B_k = \sum_{j=1}^{k} n_{c^{(i)}}(\mu_{c^{(i)}} - \mu)(\mu_{c^{(i)}} - \mu)^T, \qquad W_k = \sum_{i=1}^{m}(x^{(i)} - \mu_{c^{(i)}})(x^{(i)} - \mu_{c^{(i)}})^T$$

Calinski-Harabaz indeksi $s(k)$, kümelenme modelinin kümeleri ne kadar iyi tanımladığını gösterir, böylece skor ne kadar yüksek olursa, kümeler daha yoğun ve iyi ayrılır. Aşağıdaki şekilde tanımlanmıştır:

$$\boxed{s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1}}$$

### Temel bileşenler analizi

Verilerin yansıtılacağı yönleri maksimize eden varyansı bulan bir boyut küçültme tekniğinidir.

❒ **Özdeğer, özvektör** – Bir matris $A \in \mathbb{R}^{n\times n}$ verildiğinde $\lambda$'nın, özvektör olarak adlandırılan bir vektör $z \in \mathbb{R}^n \backslash \{0\}$ varsa, $A$'nın bir özdeğeri olduğu söylenir:

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

❒ **Spektral teorem** – $A \in \mathbb{R}^{n\times n}$ olsun. Eğer $A$ simetrik ise, o zaman $A$ gerçek ortogonal matris $U \in \mathbb{R}^{n\times n}$ n ile diyagonalleştirilebilir. $\Lambda = \mathrm{diag}(\lambda_1,...,\lambda_n)$ yazarak, bizde:

$$\boxed{\exists \Lambda \text{ diyagonal}, \quad A = U\Lambda U^T}$$

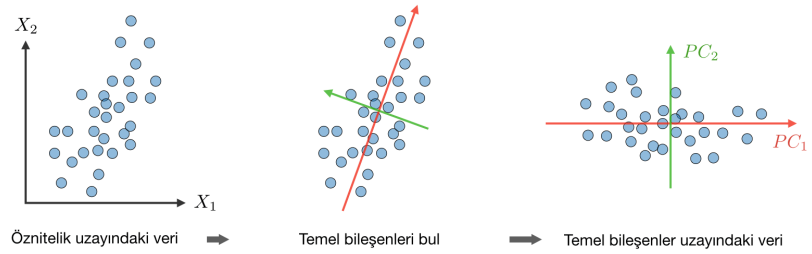
*Not: En büyük özdeğere sahip özvektör, matris A'nın temel özvektörü olarak adlandırılır.*

❒ **Algoritma** – Temel Bileşen Analizi (TBA) yöntemi, verilerin aşağıdaki gibi varyansı en üst düzeye çıkararak veriyi $k$ boyutlarına yansıtan bir boyut azaltma tekniğidir:

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

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

- Adım 2: Gerçek özdeğerler ile simetrik olan $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i)^T} \in \mathbb{R}^{n\times n}$ hesaplayın.
- Adım 3: $u_1, ..., u_k \in \mathbb{R}^n$ olmak üzere $\Sigma$ ort'nin ortogonal ana özvektörlerini, yani $k$ en büyük özdeğerlerin ortogonal özvektörlerini hesaplayın.
- Adım 4: $\mathrm{span}_{\mathbb{R}}(u_1,...,u_k)$ üzerindeki verileri gösterin. Bu yöntem tüm $k$-boyutlu uzaylar arasındaki varyansı en üst düzeye çıkarır.




### Bağımsız bileşen analizi

Temel oluşturan kaynakları bulmak için kullanılan bir tekniktir.

❒ **Varsayımlar** – Verilerin $x$'in $n$ boyutlu kaynak vektörü $s = (s_1,...,s_n)$ tarafından üretildiğini varsayıyoruz, burada $s_i$ bağımsız rasgele değişkenler, bir karışım ve tekil olmayan bir matris $A$ ile aşağıdaki gibi:

$$\boxed{x = As}$$

Amaç, işlem görmemiş matrisini $W = A^{-1}$ bulmaktır.

❒ **Bell ve Sejnowski ICA algoritması** – Bu algoritma, aşağıdaki adımları izleyerek işlem görmemiş matrisi $W$'yi bulur:

- $x = As = W^{-1}s$ olasılığını aşağıdaki gibi yazınız:

$$p(x) = \prod_{i=1}^{n} p_s(w_i^T x) \cdot |W|$$

- Eğitim verisi $\{x^{(i)}, i \in [\![1,m]\!]\}$ ve $g$ sigmoid fonksiyonunu not ederek log olasılığını yazınız:

$$l(W) = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} \log\left(g'(w_j^T x^{(i)})\right) + \log|W|\right)$$

Bu nedenle, rassal (stokastik) eğim yükselme öğrenme kuralı, her bir eğitim örneği için $x^{(i)}$, $W$'yi aşağıdaki gibi güncelleştiririz:

$$\boxed{W \longleftarrow W + \alpha\left(\begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} x^{(i)^T} + (W^T)^{-1}\right)}$$