

# Negatif Olmayan Matris Ayrışımı Modellerinde Varyasyonel Bayesci Çıkarım (Variational Nonnegative Matrix Factorisation)

A. Taylan Cemgil

Bilgisayar Mühendisliği Bölümü  
Boğaziçi Üniversitesi, 34342, Bebek, İstanbul, Türkiye

taylan.cemgil@boun.edu.tr

## Özetçe

Bu bildiriye KL ıraksayı (KL-divergence) uzaklık ölçüsü kullanılan Negatif Olmayan Matris Ayrışımına Bayesci istatistik çerçevesinde, problem hakkındaki önbilginin önsel dağılımlar yolu ile eklenebileceği bir yaklaşım öneriyoruz. Bu yaklaşımda, standart NMF eniyileme yöntemleri özel bir duruma dönüşmekte ve bir Kestirme-Enbüyütme (Expectation-Maximisation, EM) olarak da açıklanabilmektedir. Bu bakış açısından yola çıkarak Bayesci genellemelere gidebiliyor ve model seçimi gibi daha kapsamlı problemleri de klasik NMF hızında çözebiliyoruz. Yaklaşımımızı model boyutu kestirimi ve görüntü aradeğerleme için gösteriyoruz.

## Abstract

We describe non-negative matrix factorisation (NMF) in a statistical framework, with a hierarchical generative model consisting of an observation and a prior component. Omitting the prior leads to standard NMF algorithms as special cases, where maximum likelihood parameter estimation is carried out via the Expectation-Maximisation (EM) algorithm. Starting from this view, we develop Bayesian extensions that facilitate more powerful modelling and allow more sophisticated inference, such as Bayesian model selection. Our construction retains conjugacy and enables us to develop models that fit better to real data while retaining attractive features of standard NMF such as fast convergence and easy implementation. We illustrate our approach on model order selection and image reconstruction.

## 1. Giriş

Negatif Olmayan Matris Ayrışımı (Nonnegative Matrix Factorisation, NMF) ilk olarak Lee ve Seung [6] tarafından k-ortalama gruplandırma (K-means clustering) ve ana bileşenler ayrıştırılması (Principal Component Analysis) yöntemlerine bir alternatif olarak önerilmiştir. Bu modelde kaba olarak amaç, boyutları  $W \times K$  olan bir  $X = \{x_{\nu,\tau}\}$  matrisine iki negatif olamayan matrisin çarpımı cinsinden yaklaşımdır. Bir başka deyişle,  $\nu = 1:W$ ,  $\tau = 1:K$  ve  $i = 1:I$  olmak üzere

$$x_{\nu,\tau} \approx [TV]_{\nu,\tau} = \sum_i t_{\nu,i} v_{i,\tau}$$

özelliğini sağlayan  $T$  ve  $V$  matrisleri arıyoruz. Bu bildiriye,  $W \times I$  boyutundaki  $T$  matrisine *şablon matrisi*, and  $I \times K$  boyutundaki  $V$  matrisine de *katsayı matrisi* adını vereceğiz. NMF,

negatif olmama kısıtlamaları altında aşağıdaki enküçültme problemini çözer:

$$(T, V)^* = \arg \min_{T, V} D(X||TV) \quad (1)$$

Burada  $D$ , uygun bir hata fonksiyonudur. Sıkça kullanılan bir seçim Kullback-Leibler (KL) ıraksayıdır:

$$D(X||\Lambda) = - \sum_{\nu,\tau} \left( x_{\nu,\tau} \log \frac{\lambda_{\nu,\tau}}{x_{\nu,\tau}} - \lambda_{\nu,\tau} + x_{\nu,\tau} \right) \quad (2)$$

Jensen eşitsizliği [1] ve  $\log x$  dışbükeyliği (concavity) kullanılarak  $D(\cdot)$  her zaman sıfır veya pozitif olduğunu ve sadece  $X = \Lambda$  seçimi için  $D(X||\Lambda) = 0$  olduğunu gösterebiliriz. Bu denklemden (1) verilen amaç fonksiyonu uygun bir çok yöntemle eniyilenebilir. Lee ve Seung [6], bir çok uygulamada başarı ile kullanılan varyasyonel bir yöntem önermişlerdir.

Kanımızca NMF'e düşük seviyeli bir matris yaklaşımı olarak bakmak, örneğin SVD (singular value decomposition) yöntemine bir alternatif gibi, işe yarayan pratik bir algoritma geliştirmeye yetse de, verinin tam olarak nasıl modellendiğini anlamak için yeterli değildir. Burada amacımız  $X$ 'in istatistiksel özelliklerini daha net bir şekilde anlamak ve NMF'i sıradüzensel üreten bir model (hierarchical generative model) olarak betimlemektir. Bu bağlamda, NMF'in altında yatan model, koşullu olarak Poisson dağılım rastlantısal değişkenlerin yeğinlik katsayılarının birbirlerine bağlı olarak modellenmesi ile çıkmaktadır. Bu bakış açısının avantajı, veri genişletme (data augmentation) yöntemi ile klasik NMF algoritmalarını bir EM algoritması olarak görmenin mümkün olması, ve daha önemlisi Markov zinciri Monte Carlo (Markov chain Monte Carlo MCMC) veya varyasyonel (variational) ve ortalama alan (mean field) yöntemleri ile de çıkarım yapılabilmesidir. Bu da marjinal olabilirlik (marginal likelihood) hesabı ile otomatik ilgililik belirleme (automatic relevance determination) ile model seçimi veya düzenleştirmesi (regularisation) yapabilmemizi sağlamaktadır.

## 2. İstatistiksel perspektif

Aşağıdaki sıradüzensel modeli tanımlayalım:

$$T \sim p(T|\Theta^t) \quad V \sim p(V|\Theta^v) \quad (3)$$

$$s_{\nu,i,\tau} \sim \mathcal{PO}(s_{\nu,i,\tau}; t_{\nu,i} v_{i,\tau}) \quad x_{\nu,\tau} = \sum_i s_{\nu,i,\tau} \quad (4)$$

Burada

$$\mathcal{P}\mathcal{O}(s; \lambda) = \exp(s \log \lambda - \lambda - \log \Gamma(s + 1))$$

$s$  raslatsal değişkenini, yeğlilik katsayısı  $\lambda$  olan bir Poisson dağılımından gelmektedir ve gamma fonksyonu  $\Gamma(s + 1) = s!$  olarak gösterilmektedir. Önsel dağılımlar  $p(T|\cdot)$  ve  $p(V|\cdot)$  daha sonra tanımlanacaktır. Burada, *saklı kaynaklar* olarak adlandırdığımız  $S_i = \{s_{\nu,i,\tau}\}$  değişkenlerinin üstünden analitik olarak toplam olarak aşağıdaki marjinal dağılımı hesaplayabiliyoruz:

$$\begin{aligned} \log p(X|T, V) &= \log \sum_S p(X|S)p(S|T, V) \\ &= \log \prod_{\nu,\tau} \mathcal{P}\mathcal{O}(x_{\nu,\tau}; \sum_i t_{\nu,i}, v_{i,\tau}) \quad (5) \\ &= + \sum_{\nu} \sum_{\tau} (x_{\nu,\tau} \log [TV]_{\nu,\tau} - [TV]_{\nu,\tau}) \end{aligned}$$

Burada  $S = \{S_1 \dots S_I\}$  olarak tanımlandığında bu sonuç Poisson dağılımının üstüdüşüm [5] (superposition) özelliğinden kaynaklanmaktadır. Bir başka deyişle  $s_i \sim \mathcal{P}\mathcal{O}(s_i; \lambda_i)$  ve  $x = s_1 + s_2 + \dots + s_I$  olduğunda marjinal dağılım  $p(x) = \mathcal{P}\mathcal{O}(x; \sum_i \lambda_i)$  olarak belirlenmektedir. Kolayca görüleceği gibi, bu amaç fonksiyonunun enbüyütülmesi KL ıraksayımın (2) enküçültülmesine eşittir. Aslında orjinal NMF algoritmasının türetilmesinde, bu  $S$  değişkenlere eşdeğer değişkenler enbüyütme sırasında alt sınır oluşturmak için kullanılmaktadır. Burada, genişletilmiş bu değişkenlerin beklenen yeterli istatistiklerinin (expected sufficient statistics) de kolayca hesap edilebildiğini göstereceğiz:

$$\begin{aligned} p_{\nu,i,\tau} &= t_{\nu,i} v_{i,\tau} / \sum_{i'} t_{\nu,i'} v_{i',\tau} \\ \langle s_{\nu,i,\tau} \rangle &= x_{\nu,\tau} p_{\nu,i,\tau} \quad (6) \\ \log p(S|\cdot) &= \prod_{\nu,\tau} x_{\nu,\tau}! \left( \prod_i \frac{p_{\nu,i,\tau}^{s_{\nu,i,\tau}}}{s_{\nu,i,\tau}!} \right) \delta(x_{\nu,\tau} - \sum_i s_{\nu,i,\tau}) \\ &= \prod_{\nu,\tau} \mathcal{M}(s_{\nu,1:I,\tau}; x_{\nu,\tau}, p_{\nu,1:I,\tau}) \end{aligned}$$

hücre olasılıklarının  $p_i$  ve  $x = \sum_i s_i$  olduğu çokterimli dağılım (multinomial distribution) [5]  $\mathcal{M}(s_{1:I}, x, p_{1:I})$  olarak gösterilmektedir. Çokterimli dağılımın marginal dağılımları iki terimli (binomial) dağılımlar cinsinden yazılabilir ve bu dağılımlarında beklenen yeterli istatistikleri  $\langle s_i \rangle = x p_i$  olarak bulunur (burada  $\langle \cdot \rangle$  beklenen değeri belirtmektedir). EM algoritması, bir sabit nokta döngüsü ile aşağıdaki olabilirlik fonksiyonunu eniyiler:

$$\begin{aligned} \mathcal{L}_X(T, V) &\equiv \log \sum_S p(X|S)p(S|T, V) \quad (7) \\ &\geq \sum_S q(S) \log \frac{p(X, S|T, V)}{q(S)} \equiv \mathcal{B}_{EM}[q] \quad (8) \end{aligned}$$

burada  $q(S)$  herhangi bir dağılımdır,

$$\begin{aligned} \text{E:} \quad q(S)^{(n)} &= \arg \max_{q(S)} \mathcal{B}_{EM}[q] \\ &= p(S|X, T^{(n-1)}, V^{(n-1)}) \\ \text{M:} \quad (T^{(n)}, V^{(n)}) &= \arg \max_{T, V} \langle \log p(S, X|T, V) \rangle_{q(S)^{(n)}} \end{aligned}$$

M (eniyileme evresi) aşağıdaki gibidir:

$$\begin{aligned} &\langle \log p(S, X|T, V) \rangle_{p(S|X, T, V)} \\ &= \sum_{\nu} \sum_{\tau} \left( \sum_i \langle s_{\nu,i,\tau} \rangle \log(t_{\nu,i} v_{i,\tau}) - t_{\nu,i} v_{i,\tau} \right) + \text{const} \end{aligned}$$

Bu denklemde  $t|v$  ve  $v|t$  koşullu olarak gamma dağılımlıdır. Bu dağılımın doruk noktası türev sifra eşitlenerek

$$t_{\nu,i}^{(n+1)} = \sum_{\tau} \langle s_{\nu,i,\tau} \rangle^{(n)} / \sum_{\tau} v_{i,\tau}^{(n)} \quad (9)$$

$$v_{i,\tau}^{(n+1)} = \sum_{\nu} \langle s_{\nu,i,\tau} \rangle^{(n)} / \sum_{\nu} t_{\nu,i}^{(n)} \quad (10)$$

olarak bulunur. Bu denkleme 6 nolu denklemi yerleştirirsek, [6]'de betimlenen çarpansal güncelleme denklemlerini (multiplicative update equations) elde ederiz. Yani KL-NMF, veri genişletme ile oluşturduğumuz modelde EM algoritması ile enbüyük olabilirlik değeri aramaya eşittir. Teknik literatürde sıkça NMF'in EM'e benzediği söylenmektedir. Biz burada EM'e sadece benzemediğini, bir EM algoritması olduğunu gösterdik. Gördüğü gibi NMF'in avantajı  $W \times I \times K$  boyutundaki  $\langle S \rangle$  objesinin hiç bir zaman belirtik (explicit) olarak hesaplanmasının gerekmemesi ve sadece  $\tau$  ve  $\nu$  üzerinden alınan marjinallerinin hesaplamaya yetmesidir. Bir sonraki bölümde modeli geliştirip  $T$  ve  $V$  üzerinden noktasal kestirimler yapmak yerine bu değişkenler üzerinden integral hesaplayacağız.

## 2.1. Sıradüzensel Model

Olasılık modelinin genel yapısı anlaşıldıktan sonra belirli bir uygulamaya yönelik olarak modeli farklı önsel dağılımlar tanımlanarak geliştirmek mümkündür. Biz burada basit ve uyumlu (conjugate) bir yapı üzerinde duracağız:

$$t_{\nu,i} \sim \mathcal{G}(t_{\nu,i}; a_{\nu,i}^t, b_{\nu,i}^t / a_{\nu,i}^t), \quad v_{i,\tau} \sim \mathcal{G}(v_{i,\tau}; a_{i,\tau}^v, b_{i,\tau}^v / a_{i,\tau}^v)$$

buradaki indeks değişkenleri en genel hali yansıtıyor ve modeldeki her değişken için ayrı bir hiperparametre seçiliyor. Uygulamalarda bu hiperparametreleri birbirlerine bağlamak da mümkün. Verinin bir bölümü eksikse, yani  $x_{\nu,\tau}$  bazı elemanları gözlemlenmemişse bir *maske* matrisi tanımlıyoruz. Bu  $M = \{m_{\nu,\tau}\}$  olarak adlandırdığımız matris,  $X$  ile aynı boyutta ve  $x_{\nu,\tau}$  gözlemlenmemişse  $m_{\nu,\tau} = 0$  yoksa 1 olarak tanımlanıyor. Bu maskeyi kullanarak olabilirlik fonksiyonunu şu şekilde yazıyoruz:

$$p(X, S|T, V) = \prod_{\nu,\tau} (p(x_{\nu,\tau} | s_{\nu,1:I,\tau}) p(s_{\nu,1:I,\tau} | t_{\nu,1:I}, v_{1:I,\tau}))^{m_{\nu,\tau}}$$

## 2.2. Çıkarım

Bu bölümde Varyasyonel Bayes (VB) yönteminin ana hatlarını anlatacağız. VB [4, 1] metodu aslında marjinal olabilirlik fonksiyonunun bir alt sınırının eniyilenmesi üzerine kurulu bir yöntem

$$\mathcal{L}_X(\Theta) \equiv \log p(X|\Theta) \quad (11)$$

$$\geq \sum_S \int d(T, V) q \log \frac{p(X, S, T, V|\Theta)}{q} \quad (12)$$

$$= \langle \log p(X, S, V, T|\Theta) \rangle_q + H[q] \equiv \mathcal{B}_{VB}[q]$$

burada,  $q = q(S, T, V)$  herhangi bir dağılım ve  $H[q]$  bu dağılımın entropisidir. Gerçek sonsal dağılım için bu alt sınır marjinal olabilirlik fonksiyonuna eşittir:  $q(S, T, V) = p(S, T, V|X, \Theta)$ , ama buradaki zorluk bu şekilde seçilen  $q$  dağılımının çok karmaşık olmasıdır. Bunun yerine daha basit bir dağılım ailesi seçersek, örneğin çarpanlarına ayrılan

$$q(S, T, V) = q(S)q(T)q(V) \equiv \prod_{\alpha \in \mathcal{C}} q_{\alpha} \quad (13)$$

$$= \left( \prod_{\nu, \tau} q(s_{\nu, 1: I, \tau}) \right) \left( \prod_{\nu, i} q(t_{\nu, i}) \right) \left( \prod_{i, \tau} q(v_{i, \tau}) \right) \quad (14)$$

$\alpha \in \mathcal{C} = \{\{S\}, \{T\}, \{V\}\}$ . Böyle seçilen bir  $q$  dağılımı gerçek sonsal dağılımın barındırdığı yapıya tam olarak yaklaşmayacağı için alt sınır marjinal olabilirliğin altında kalacaktır. VB yöntemi bu alt sınırı en iyileyen bir yöntemdir. Bu durumda eniyileyen çözüm bir sabit nokta döngüsü sonucunda bulunabilir:

$$q_{\alpha}^{(n+1)} \propto \exp \left( \langle \log p(X, S, T, V | \Theta) \rangle_{q_{\alpha}^{(n)}} \right) \quad (15)$$

Burada  $q_{-\alpha} = q/q_{\alpha}$  olarak tanımlanmıştır. Bu döngü,  $q$  dağılımının çarpanlarını güncelleyerek alt sınırı her adımda eniyiler, dolayısıyla algoritma yöresel bir minimumda kalır; bir başka deyişle  $n = 1, 2, \dots$  ve verilen bir  $q^{(0)}$  için  $\mathcal{B}[q^{(n)}] \leq \mathcal{B}[q^{(n+1)}]$ . Sabit nokta döngüsü saklı kaynaklar  $S$  için ( $m_{\nu, \tau} = 1$ , ve katsayı matrisi  $V$  için aşağıdaki gibi bulunur

$$q(s_{\nu, 1: I, \tau}) = \mathcal{M}(s_{\nu, 1: I, \tau}; x_{\nu, \tau}, p_{\nu, 1: I, \tau}) \quad (16)$$

$$q(v_{i, \tau}) = \mathcal{G}(v_{i, \tau}; \alpha_{i, \tau}^v, \beta_{i, \tau}^v) \quad (17)$$

$$p_{\nu, i, \tau} = \frac{\exp(\langle \log t_{\nu, i} \rangle + \langle \log v_{i, \tau} \rangle)}{\sum_i \exp(\langle \log t_{\nu, i} \rangle + \langle \log v_{i, \tau} \rangle)} \quad (18)$$

$$\alpha_{i, \tau}^v = a_{i, \tau}^v + \sum_{\nu} m_{\nu, \tau} \langle s_{\nu, i, \tau} \rangle \quad (19)$$

$$\beta_{i, \tau}^v = \left( \frac{a_{i, \tau}^v}{b_{i, \tau}^v} + \sum_{\nu} m_{\nu, \tau} \langle t_{\nu, i} \rangle \right)^{-1} \quad (20)$$

Yaklaşım dağılımının parametreleri de  $q(t_{\nu, i}) = \mathcal{G}(t_{\nu, i}; \alpha_{\nu, i}^t, \beta_{\nu, i}^t)$  benzer şekilde bulunur. Yukarıdaki algoritmayı matris notasyonunda da yazmak mümkündür. Eleman eleman çarpma ve bölme operatorlarını sırasıyla  $*$  ve  $/$  olarak tanımlıyoruz. Yukarıda türettiğimiz *varyasyonel negatif olmayan matris ayrıştırma* algoritması aşağıda özetlenmiştir.

1: Tanımlar :

$$E_t = \langle t_{\nu, i} \rangle \quad L_t = \exp(\langle \log t_{\nu, i} \rangle) \quad \Sigma_t = \sum_{\tau} \langle s_{\nu, i, \tau} \rangle$$

$$A_t = \alpha_{\nu, i}^t \quad B_t = b_{\nu, i}^t \quad \alpha_t = \alpha_{\nu, i}^t \quad \beta_t = \beta_{\nu, i}^t$$

$$E_v = \langle v_{i, \tau} \rangle \quad L_v = \exp(\langle \log v_{i, \tau} \rangle) \quad \Sigma_v = \sum_{\nu} \langle s_{\nu, i, \tau} \rangle$$

$$A_v = a_{i, \tau}^v \quad B_v = b_{i, \tau}^v \quad \alpha_v = \alpha_{i, \tau}^v \quad \beta_v = \beta_{i, \tau}^v$$

2: Başla :

$$L_t^{(0)} = E_t^{(0)} \sim \mathcal{G}(\cdot; A_t, B_t ./ A_t)$$

$$L_v^{(0)} = E_v^{(0)} \sim \mathcal{G}(\cdot; A_v, B_v ./ A_v)$$

3: **for**  $n = 1 \dots$  MAXITER **do**

4: Kaynakların yeterli istatistikleri

$$\Sigma_t^{(n)} := L_t^{(n-1)} .* ((X .* M) ./ (L_t^{(n-1)} L_v^{(n-1)})) L_v^{(n-1) \top}$$

$$\Sigma_v^{(n)} := L_v^{(n-1)} .* (L_t^{(n-1) \top} ((X .* M) ./ (L_t^{(n-1)} L_v^{(n-1)})))$$

5: Ortalama değerler

$$E_t^{(n)} := \alpha_t^{(n)} .* \beta_t^{(n)}$$

$$\alpha_t^{(n)} = A_t + \Sigma_t^{(n)}$$

$$\beta_t^{(n)} = 1. / (A_t ./ B_t + M E_v^{(n-1) \top})$$

$$E_v^{(n)} := \alpha_v^{(n)} .* \beta_v^{(n)}$$

$$\alpha_v^{(n)} = A_v + \Sigma_v^{(n)}$$

$$\beta_v^{(n)} = 1. / (A_v ./ B_v + E_t^{(n) \top} M)$$

6: (İstenirse) alt sınırı hesapla ([2])

7: Logların ortalamaları

$$L_t^{(n)} = \exp(\Psi(\alpha_t^{(n)})) .* \beta_t^{(n)}$$

$$L_v^{(n)} = \exp(\Psi(\alpha_v^{(n)})) .* \beta_v^{(n)}$$

8: (İstenirse) hiperparametreleri güncelle ([2])

9: **end for**

Benzer bir şekilde döngülü koşullu doruklar (iterative conditional modes (ICM)) veya en büyük sonsal olasılık çözümü de (maximum a-posteriori (MAP)) bulunabilir:

$$V := (A_v + V .* (T^{\top} ((M .* X) ./ (TV)))) ./ (A_v ./ B_v + T^{\top} M)$$

$$T := (A_t + T .* ((M .* X) ./ (TV)) V^{\top}) ./ (A_t ./ B_t + M V^{\top})$$

Bu denklemlerden de görüneceği gibi,  $A_t, A_v \rightarrow \mathbf{0}$  olduğunda orijinal NMF algoritmasını buluruz.

### 3. Benzetim Çalışmaları

Yaklaşımımızı öncelikle bir model seçme probleminde göstereceğiz ve varyasyonel algoritmayı bir Gibbs örnekleyicisi ile karşılaştıracacağız. Yer darlığından dolayı Gibbs örnekleyicisini ve Chib metodu ile [3] marjinal olabilirlik hesabını burada türetmiyoruz. Detaylar bu bildirinin daha kapsamlı bir sunumunda bulunabilir [2].

**Model seçimi:** Burada yöntemimizi (4) numaralı denklemdaki modelden sentetik olarak üretilmiş veri üzerinde deniyoruz. Burada  $W = 16$ ,  $K = 10$  ve kaynakların sayısı  $I_{\text{true}} = 5$ . Çıkarımın amacı sadece  $X$  verildiğinde gerçek modeli bulmak. Gerçek modelin hiperparametreleri  $a_{\nu, i}^t = a^t = 10$ ,  $b_{\nu, i}^t = b^t = 1$ ,  $a_{i, \tau}^v = a^v = 1$ ,  $b_{i, \tau}^v = b^v = 100$  olarak alındı. İlk deneyde hiperparametrelerin bilindiğini varsayıyoruz. İkinci deneyde ise bu parametreleri de veriden buluyoruz. İlk deneyde şablon sayısını  $I = 1 \dots 10$  arasında değiştirerek her model için marjinal olabilirliği Gibbs örnekleyicisi ve VNMF ile kestirdik. Gibbs örnekleyicisini 5000 adımlık bir ilk ısınma (burn-in) devresinden sonra MAXITER = 10000 adım koşturduk. Daha sonra kaynakları ( $S$ ) sabitleyip benzetime 10000 adım daha devam ettik. Varyasyonel algoritmayı isel en fazla 10000 adım koşturduk. Şekil 1'de, Gibbs örnekleyicisi ile VNMF'i karşılaştırıyoruz. Burada Gibbs sonuçları 5 değişik benzetim çalışmasının ortalaması olarak hesaplandı.

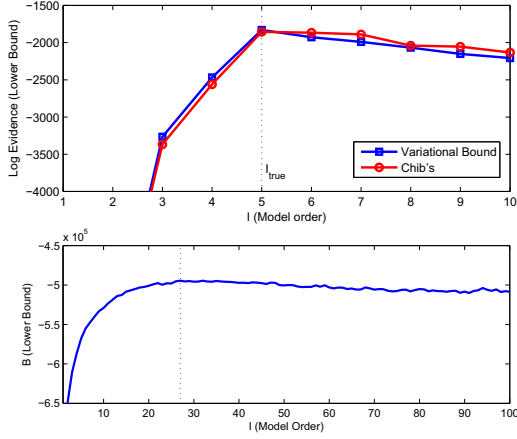


Figure 1: (Yukarıdan Aşağıya) Model seçimi karşılaştırmaları. Varyasyonel alt sınır (kareler) ve Chib's metodundan gelen marjinal olabilirlik kestirimi (daireler). Burada hiperparametrelerin bilindiği varsayılıyor. Sonuçlar, hiperparametreler bilinmediğinde de benzer çıkıyor. Yüz görüntüleri ( $16 \times 16$ ) da  $I^* = 27$  ve  $32 \times 32$  de  $I^* = 42$ .

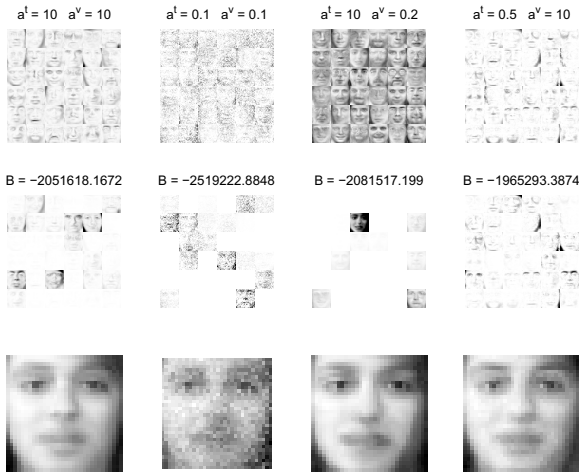


Figure 2: Şablon ve Katsayı matrislerinin belirli bir örnek için farklı hiperparametreler ile bulunan değerleri. B bütün veri tabanı kullanılarak bulunan alt sınır.

Şekil 1 de ise, hiperparametreleri de eniyilediğimiz durumda bulduğumuz alt sınırı model büyüklüğü  $I'$ e bağlı olarak çizdirdik. Bu sonuçlar, gerçek  $a_t, b_t, a_v$  ve  $b_v$  hiperparametrelerini bilmesek bile model büyüklüğünü hesaplayabileceğimizi gösteriyor. Bu haliyle yöntemin gerçek veri üzerinde uygulanabilmesinde önemli bir nokta.

Gerçek veri olarak da Olivetti yüz veri tabanını kullandık ( $64 \times 64$  piksellik  $K = 400$  resim, <http://www.cs.toronto.edu/~roweis/data/olivettifaces.mat>). Görüntüleri,  $16 \times 16$  veya  $32 \times 32$  boyutuna indirdik. Bu durumda veri matrisi  $X$   $16^2 \times 400$  veya  $32^2 \times 400$  boyutlarında oldu. Hiperparametreleri  $a_{\nu,i}^t = a^t$ ,  $b_{\nu,i}^t = b^t$ ,  $a_{i,\tau}^v = a^v$  ve  $b_{i,\tau}^v = b^v$  şeklinde birbirlerine

bağladık ve veriden beraber kestirdik. Şekil 1 alt kısımda bulunan model büyüklüğünü gösteriyoruz. Burada VNMF ile her model büyüklüğünün marjinal olabilirliğini ayrı ayrı kestirdik. Gibbs örnekleyicisi, bu boyuttaki bir problem için çok pratik olmadığından kullanılmadı. Buradaki gözlemimiz, varyasyonel alt sınırın marjinal olabilirliğe benzer bir şekilde hareket ediyor olması: model gerektiğinden çok küçük veya çok büyükse marjinal olabilirlikten beklenildiği gibi düşük kalıyor. Çözünürlük arttığında tahmin edilebileceği gibi daha çok şablon kullanmak gerekiyor. Ne yazık ki yer darlığından dolayı sonuçları buraya ekleyemiyoruz. İlgilenen okuyucuyu, bu ve benzer sonuçların daha detaylı tartışıldığı bir teknik rapora bakmaya davet ediyoruz [2]. Bulunan şablon ve katsayı matrislerinin yapısı hiperparametrelere göre epey değişiyor (Şekil 2). Bunu göstermek için öncelikle  $(a^t, a^v) = [(10, 10), (0.1, 0.1), (10, 0.2), (10, 0.5)]$  olarak sadece  $b^t$  ve  $b^v$  veriden kestirdik. Burada büyük  $a$  değerleri,  $t$  ve  $v$  matrislerini kıt olmayan (non-sparse), küçük değerler ise daha kıt (sparse) çözümlere itiyor.

### 3.1. Sonuç ve Yorumlar

Bu çalışmada, sıkça kullanılan KL-NMF algoritmasının aslında ilgili bir sıradüzensel modelde kestirim yapan bir EM algoritması olduğunu gösterdik. Buradan yola çıkışla da model mertebesi kestirimi problemine bir çözüm önerdik. Benzetim sonuçları, önerdiğimiz yöntemin varyasyonel alt sınırın marjinal olabilirliğe makul bir yaklaşım sağladığını ve model seçimi için yararlı olabileceğini gösteriyor. Modelleme açısından yaklaşımımızın bir kaç avantajı var. Ön bilgi kolay bir şekilde entegre edilebiliyor veya eniyilenmiş kıtlık ölçütünü de (optimal sparseness criteria) veriden otomatik olarak öğrenebiliyoruz. Burada önemli bir nokta bulunan yöntemin hesap ağırlığının orjinal NMF'e yakın olması. Belki de daha önemlisi, yöntemi matlab gibi matris tabanlı sistemlerde gerçekleştirmek orjinal algoritma kadar kolay.

## 4. Kaynakça

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [2] A. T. Cemgil. Bayesian inference in non-negative matrix factorisation models. Technical Report CUED/INFENG/TR.609, University of Cambridge, July 2008. Submitted for publication to Computational Intelligence and Neuroscience.
- [3] S. Chib. Marginal likelihood from the gibbs output. *JASA*, 90(432):1313–1321, Dec. 1995.
- [4] Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In *Neural Information Processing Systems 13*, 2000.
- [5] J. F. C. Kingman. *Poisson Processes*. Oxford Science Publications, 1993.
- [6] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, 1999.