Bayesian Segmentation of Natural Scenes using Dependent Pitman Yor Processes

Sebastian Thiel

January 6, 2011

Contents

1 Motivation: **2**

2 Introduction: **3**

3 Preprocessing/Superpixels **3**
3.1 From NCut to Eigenproblems . . . 3
3.2 How to get the affinity matrix W? . . . 7
3.2.1 comparing pixels by color . . . 7
3.2.2 comparing pixels by texture . . . 7
3.2.3 Comparing pixels by intervening contour . . . 8
3.3 Putting the weights together.. . . . 8
3.4 Superpixels: Results . . . 9

4 Technical Details/Notation **9**

5 Hirarchical Pitman-Yor Model Without Spatial Dependencies **10**
5.1 Introduction . . . 10
5.2 The model: . . . 11
5.3 Variational Inference . . . 12
5.3.1 Factorization . . . 12
5.3.2 Summary Of Results: . . . 13
5.3.3 Towards Algorithm . . . 14
5.3.4 Algorithm . . . 17
5.4 Comment . . . 18

6 Dependent Pitman-Yor Model For One Image **18**
6.1 Introduction: . . . 19
6.1.1 Variable transformation . . . 20
6.2 The model: . . . 21
6.3 Variational Inference . . . 21
6.3.1 Factorization . . . 21
6.3.2 Lower bound . . . 22

6.3.3 Calculating the free form distribution of $q(\theta)$ . . . 23
6.3.4 Calculating the distributions $q(\bar{\nu}_k)\prod_{i=1}^{N} q(u_{ki})$ . . . 25
6.4 Results: . . . 29
6.4.1 Comparison to KMeans clustering . . . 29

7 Dependent Pitman-Yor Model on multiple images **30**
7.1 Introduction . . . 30
7.2 The model . . . 30

8 Conclusion **31**

9 Acknowledgements **32**

10 Appendix **32**
10.0.1 Completing the Square . . . 32
10.1 Factorized distributions . . . 32
10.2 Properties of factorized approximations . . . 33
10.2.1 Results: . . . 34
10.2.2 Comparison: reverse Kullback-Leibler divergence: minimize $\mathbf{KL}(p \parallel q)$ instead of $\mathbf{KL}(q \parallel p)$ . . . 34
10.3 HPY model (first model): Calculation of the factorized distributions . . . 35
10.3.1 Calculating $q(\mathbf{t})$ . . . 35
10.3.2 Calculating $q(\nu)$ . . . 36
10.3.3 Calculating $q(k)$ . . . 37
10.3.4 Calculating $q(\omega)$ . . . 37
10.3.5 Calculating $q(\theta)$ . . . 38
10.4 DPY model (second model): Calculation of Lowerbound . . . 39

11 References **44**

1 Motivation:

Image Segmentation is often used as the first step to perform image recognition. Segmentation separates the image in the distinct objects or topics (e.g. sky, tree, car, ground) in an image without naming/recognizing the objects. Image segmentation as a preprocessing step during recognition is necessary when contour based descriptors like Fourier Descriptors of Template Descriptors fail, because the objects to be recognized do not follow rigid forms. Image recognition of natural scenes is such a case, as trees, brushes, clouds etc. have quite varying forms or contours. To segment such images nevertheless, it was necessary to develope new strategies: Instead of recognizing the object as a whole directly, the idea is to build the objects out of atomic structures called superpixels, merged patches of adjacent pixels, which have similar texture or color. Therein it is used, that most objects consist of connected, smoothly changing parts and that the distribution of area for particular objects in an image follows power law distributions. The first property is modelled by a Gaussian process over all datapoints (superpixels), while the second is governed by a Pitman-Yor process, a generalization of a Dirichlet process.

2 Introduction:

The objective of my diploma thesis is to implement the algorithm suggested in the paper "Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes" by Erik Sudderth and Michael I. Jordan of late 2008, where unsupervised image segmentation is done. In addition to the restricted length of the paper the author refers to his Technical Report, where further details to the complex algorithm and its implementation were outspread. Unfortunately this report did not exist till the end of the diploma thesis, which changed the objective of my thesis to the derivation *and* implementation of the paper.

The work is distributed the following: I start with the preprocessing step, where the image is subdivided in superpixel. After this I move on with a Hierarchical Pitman-Yor model without spatial dependencies of the superpixels for multiple images, where the power law distribution of area in an image as well as the relative frequencies of occurance of an image category are modelled by Pitman-Yor Processes. Then I discuss a Dependent Pitman-Yor model for one image, where spatial dependencies are modelled my Gaussian Processes. The latter model is implemented in this thesis and will be illuminated in most detail. An outview is done in the next part, where the spatial dependent model is extended to multiple images. Finally results and comments conclude the work.

3 Preprocessing/Superpixels

We want to subdivide the image in superpixels, locally affine, coherent parts to get a first (over) segmentation. Later we will merge these superpixels to form our segments, which is done using different stochastic processes.

The superpixel calculation is at first approached by connecting the pixels of the image in an affinity graph G={V,E}, where the edge weights represent the pairwise affinity between two pixels. In this work affinity is investigated by comparing HSV color histograms of both pixel and their particular neighbours, texton histograms and by checking intervening contours between the pixels. Once we have found appropriate edge weights, we can see the problem of subdividing the pixels in affine fields as the problem of minimizing the cut cost in this affinity graph G, while at the same time keeping the superpixels over a certain minimum size. Minimizing the cut cost in the graph means, in this case, separating pixels with low edge weights, respectively pixels with a low affinity, while at the same time keeping those pixels together, that have very similar properties. I will show now, why solving an eigenproblem of a normalized graph Laplacian is equivalent to solving the above problem:

3.1 From NCut to Eigenproblems

Consider first the case of two groups , k=2: We want to minimize the cut between two partitions in a graph G={V,E}, or: solve the optimization problem

$$\underset{A \subset V}{min}\, Ncut(A, \bar{A}) \quad = \quad \underset{A \subset V}{min}\, \frac{cut(A, \bar{A})}{Vol(A)} + \frac{cut(\bar{A}, A)}{Vol(\bar{A})},$$

where $cut(A, \bar{A}) = \sum_{i \epsilon A, j \epsilon \bar{A}} W_{ij}$ and $Vol(A) = \sum_{i \epsilon A} d_i$ and $d_i = \sum_{j=1}^{n} W_{ij}$.

In this case W_{ij} is the edge weight between points i and j, N is the total number of points. Hence, d_i is the *degree* of a vertex i, the sum of all adjacent edges, and $Vol(A)$ a measure for the *connectivity* of the group A, the sum of all edges adjacent to nodes of A. With these measures the $Ncut(A, \bar{A})$, the *normalized cut,*

deserves his name to be a *normalized* version of the simple $Cut(A,\bar{A})$, as the denominators $Vol(A), Vol(\bar{A})$ work balancing concerning the size of each group. Let's define a cluster indicator vector f:

$$f_i = \begin{cases} \sqrt{(\frac{Vol(\bar{A})}{Vol(A)})} & i \epsilon A \\ -\sqrt{(\frac{Vol(A)}{Vol(\bar{A})})} & i \epsilon \bar{A} \end{cases},$$

where the indicator values have certain values, that will turn out to be useful later.

I will now show the close relationship of the $Ncut$ to the graph Laplacian L(v.Luxburg,2006)):

$$\begin{aligned} f^T L f &= \sum_{i,j=1}^{n} W_{ij}(f_i - f_j)^2 \\ &= \sum_{i \epsilon A, j \epsilon \bar{A}} W_{ij}(\sqrt{\frac{Vol(\bar{A})}{Vol(A)}} + \sqrt{\frac{Vol(A)}{Vol(\bar{A})}})^2 + \sum_{i \epsilon \bar{A}, j \epsilon A} W_{ij}(-\sqrt{\frac{Vol(\bar{A})}{Vol(A)}} - \sqrt{\frac{Vol(A)}{Vol(\bar{A})}})^2 \\ &= \sum_{i \epsilon A, j \epsilon \bar{A}} W_{ij}(\frac{Vol(\bar{A})}{Vol(A)} + 2 + \frac{Vol(A)}{Vol(\bar{A})}) + \sum_{i \epsilon \bar{A}, j \epsilon A} W_{ij}(\frac{Vol(\bar{A})}{Vol(A)} + 2 + \frac{Vol(A)}{Vol(\bar{A})}) \\ &= 2 * \sum_{i \epsilon A, j \epsilon \bar{A}} W_{ij}(\frac{Vol(\bar{A}) + Vol(A)}{Vol(A)} + \frac{Vol(\bar{A}) + Vol(A)}{Vol(\bar{A})}) \\ &= 2 * (Vol(\bar{A}) + Vol(A)) \sum W_{ij}(\frac{1}{Vol(A)} + \frac{1}{Vol(\bar{A})}) \\ &= 2 * Vol(V) * Ncut(A, \bar{A}) \end{aligned}$$

So, as $Vol(V)$ is a constant, the problem of minimizing $Ncut(A,\bar{A})$ is equivalent to minimizing $f^T L f$. Furthermore one can show (see v.Luxburg,'06 for details), that

1. $(Df)^T * ones(N) = 0$, where $D = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}$ and $ones(n) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \} n$
2. $f^T D f = Vol(V)$

Thus, we can rewrite the problem of minimizing $Ncut(A,\bar{A})$ by the equivalent problem:

$$\underset{A \subset V}{min} f^T L f \text{ s.t.}$$

- f defined as above
- $Df \perp ones(n)$
- $f^T D f = Vol(V)$

Now, relax the problem, allow f to be real valued, to avoid solving an NP hard problem caused by the balancing, as v. Luxburg remarks, and subsitute

$$\begin{aligned} g &= D^{\frac{1}{2}} f \\ \Longleftrightarrow f &= D^{-\frac{1}{2}} g \end{aligned}$$

Now we get:

$$\min_{g \epsilon \mathbb{R}^n} g^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} g$$

s.t.

1. $Df \perp ones(n) \Leftrightarrow DD^{-\frac{1}{2}} g \perp ones(n) \Leftrightarrow D^{\frac{1}{2}} g \perp ones(n) \Leftrightarrow g \perp D^{\frac{1}{2}} ones(n)$
2. $f^T Df = Vol(V) \Leftrightarrow g^T D^{-\frac{1}{2}} DD^{-\frac{1}{2}} g = Vol(V) \Leftrightarrow g^T g = Vol(V)$

$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} =: L_{sym}$ is the *normalized graph Laplacian* and v. Luxburg showed that L_{sym} has got the smallest eigenvector $D^{\frac{1}{2}} ones(n)$ and $Vol(V)$ is constant. We know that eigenvectors are orthogonal to one another, which would make the second smallest eigenvector of L_{sym} a solution to the problem, according to constraint 1), while constraint 2) doesn't play an important role, as eigenvectors can be scaled without changing the meaning. But why can our problem be transferred in an eigenproblem?

$$\min_{g \epsilon \mathbb{R}^n} g^T L_{sym} g = \lambda$$

be the minimum value of our problem.

$$\Leftrightarrow L_{sym} g = \lambda g$$

is the standard eigenproblem with eigenvalues λ to the eigenvectors g. Certainly we are interested in small values of λ. As we know that all eigenvalues of $L_{sym}$are sorted, the smallest eigenvector $D^{\frac{1}{2}} ones(n)$ is useless for our purposes, because of lack of separation power and constraint 1). So the best solution to our problem is the second smallest eigenvalue to the according eigenvector (Rayleigh-Ritz problem). Now we resubstitude

$$\begin{aligned} f &= D^{-\frac{1}{2}} g \\ \Leftrightarrow g &= D^{\frac{1}{2}} f \end{aligned}$$

:

$$\begin{aligned} L_{sym} D^{\frac{1}{2}} f &= \lambda D^{\frac{1}{2}} f \\ \Leftrightarrow D^{-\frac{1}{2}} L D^{-\frac{1}{2}} D^{\frac{1}{2}} f &= \lambda D^{\frac{1}{2}} f \\ \Leftrightarrow D^{-\frac{1}{2}} L f &= \lambda D^{\frac{1}{2}} f \\ \Leftrightarrow D^{-1} L f &= \lambda f \\ \Leftrightarrow L f &= \lambda D f \end{aligned}$$

which is the generalized eigenproblem. So the solution for f is the second smallest eigenvector of the generalized eigenproblem above.

Generalization to $k > 2$: We define indicator vectors $h_j = (h_{1,j}, \cdots, h_{n,j})$ by

$$h_{i,j} = \begin{cases} 1/\sqrt{Vol(A_j)} & i \epsilon A_j \\ 0 & else \end{cases}$$

Then we set H to be the matrix containing the k indicator vectors h_j

$$H := \left[\begin{pmatrix} \vdots \\ h_1 \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ h_2 \\ \vdots \end{pmatrix} \cdots \begin{pmatrix} \vdots \\ h_j \\ \vdots \end{pmatrix} \right]$$

by definition the columns are orthogonal and hence

$$H^T H = I$$

Let's consider:

$$\begin{aligned} h_j^T D h_j &= \sum_{i=1}^{n} d_i h_{i,j}^2 \\ &= \sum_{i \epsilon A_j} d_i * \frac{1}{Vol(A_j)} + \sum_{i \epsilon \bar{A} j} d_i * 0 \\ &= \frac{\sum\limits_{i \epsilon A_j} d_i}{Vol(A_j)} = \frac{Vol(A_j)}{Vol(A_j)} = 1 \end{aligned}$$

and

$$\begin{aligned} h_j^T L h_j &= \sum_{l,i=1}^{n} W_{l,i}(h_{j,l} - h_{j,i})^2 \\ &= 2\frac{cut(A_j, \bar{A_j})}{Vol(A_j)} \\ &= 2Ncut(A_j, \bar{A_j}) \end{aligned}$$

So we get :

$$Ncut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{j=1}^{k} \frac{cut(A_j, \bar{A_j})}{Vol(A_j)} = \frac{1}{2} H^T L H = \frac{1}{2} TR(H^T L H)$$

Let us finally rewrite our optimization problem for k cuts:

$$\min_{A_1, \cdots, A_k} TR(H^T L H)$$

s.t.

- $H^T DH = I$
- H defined as above

Relax the discreteness condition and substituting

$$\begin{aligned} U &= D^{\frac{1}{2}} H \\ \iff H &= D^{-\frac{1}{2}} U \end{aligned}$$

we get:

$$\underset{U \in \mathbb{R}^{nxk}}{min} TR(UD^{-\frac{1}{2}} LD^{-\frac{1}{2}} U)$$

s.t.

- $U^T U = I$

Very analogue to the problem of k=2, resubstituting $U = D^{\frac{1}{2}} H$ and using similar properties as above, v. Luxburg showed, that the solution H consists of the first k eigenvectors of the *generalized eigenproblem*

$$Lv = \lambda Dv$$

as columns. Mitendra Malik and Jianbo Shi called this 2000 *normalized cuts spectral clustering.*

Let us interprete this result: Each column of the matrix H is an indicator vector for the datapoints (pixels) to one group. A value bigger than a certain threshold indicates membership to the group, while a value beneath means no membership to the group. So the first k eigenvectors are indicator vectors to k different groups. *K means clustering* applied rowwise will help us to cluster these indicator information in a k dimensional space in desired many groups. These groups are our superpixels.

3.2 How to get the affinity matrix W?

3.2.1 comparing pixels by color

Color affinity between two pixels is estimated by comparing *kernel density estimate* histograms of the three color channels in hsv color space. Regarding performance Kernel density estimates are implemented as gaussian blobs centered on the corresponding h, s, v value with a certain variance to allow unsharpness of the color value. The final *kde* histogram is a normalized summation of the *kdes* of the pixel an the pixels in a certain neighborhood. How big this neighbourhood is depends on the periodicity of the texture the pixel is in. The histograms of the two pixels are compared by estimating their X^2distance.

3.2.2 comparing pixels by texture

Texture affinity between two pixels is calculated by comparing soft binning *texton* histograms.

What are *textons* ? When we investigate the texture of a pixel, it makes sence to take its surrounding into account. We convolute an image with a certain filter and take the filter response value at the desired pixel. This pixel value has the desired property, as the value is influenced by the surrounding pixels. If another pixel convoluted with the same filter has a similar value, it is quite probable that the two pixels have a similar texture. If we now convolute an image with several (like 40) very different filters and compare their (in this case 40 dimensional) filter response vector by calculating its distance in a 40 dim. vector space, we will find similar textures when the vectors have low distance.

Of course, saving a 40 dimensional filter response vector for every pixel in an image needs a lot of resources. To avoid that we use K means to cluster the filter response vectors in like 100 groups, and assign each pixel to the group, whose mean vector is closest to the filter response vector of the pixel, which is automatically done by the Kmeans algorithm. These groups are our *textons*.

texton histograms In my work, like proposed by Malik et al.(2001), I convoluted the image with *Oriented Even Symmetric Filters, Oriented Odd Symmetric Filters* and *Difference of Gaussian Filters* at various scales and orientations. After having assigned each pixel to one *texton* group, I additionally saved a *texton histogram* h_j for each texton group. The histogram in the above case is a 100 bin histogram, where the value of bin i in h_j is the euclidian distance of the texton mean vectors of group i and j. This technique is called soft binning. Comparing *texton histograms* it creates a continous distance measure between two textures.

Again the texture affinity is calculated by estimating X^2 distance between the two *texton histograms.*

3.2.3 Comparing pixels by intervening contour

If two pixels are separated by a strong contour, e.g. a sudden change of colour or lightness, it is quite probable that they don't belong to the same group. Exceptions are contours inside a texture, which should be ignored.

So after having found contours by convoluting the image with *Odd Symmetric Filters* at various orientations and scales and applying *non-maximum- suppression* on pixels near a contour maximum, I additionally applied a *textureness measure*, also proposed by Malik et al(2001), to filter out those edges or contours inside a texture :If on both sides perpendicular to an edge the the texture histograms are similar, we are most probably inside a texture. So this edge should not be used as a separation criterion. So, *textureness* of a pixel suppresses *contourness* which is reflected in the following formula:

$$p_{contour} = (1 - p_{texture}) * p_{contour+texture}$$

Now, in this contour image, pixel affinity is calculated by getting the contour maximum on a straight line between the two investigated pixels. The bigger the maximum, the less the weight $W_{ij,IC}$between the two pixels.

3.3 Putting the weights together..

Alltogether we now have:

- $$W_{HSV} = exp(-(\frac{distance_h}{\sigma_h} + \frac{distance_s}{\sigma_s} + \frac{distance_v}{\sigma_v}))$$
- $$W_{TEXTURE} = exp(-\frac{distance_{tex}}{\sigma_{tex}})$$

- $$W_{InterveningContour} = \max_{m \in G_{ij}} contourvalue(m)$$

  , where $G_{ij}$are all points on a straight line connecting pixel i and j

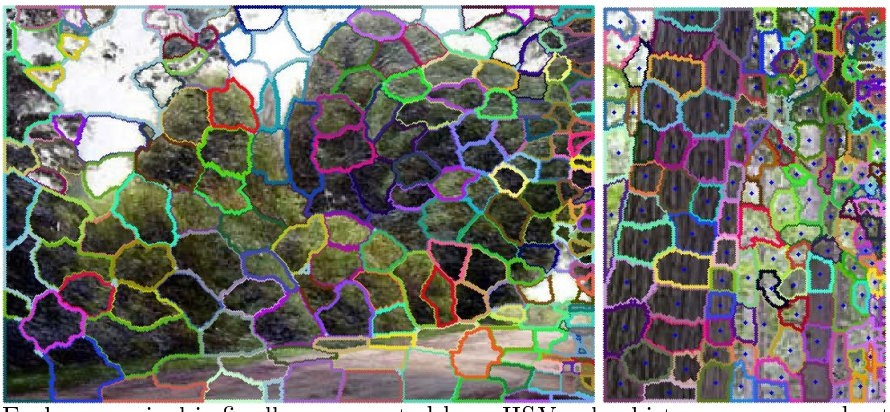
I combine these weights by

$$W = W_{IC} * (W_{HSV} + W_{TEXTURE})$$

This formulation supresses high HSV and TEXTURE values, whenever a strong contour is between the pixels. This dominant role of W_{IC} makes sense regarding the fact that W_{IC} is exempted by contours inside a texture.

3.4 Superpixels: Results

As one can see in the below image, locally affine pixels are merged to a superpixel, while sharp edges/contours separate into distinct superpixels.



Each superpixel is finally represented by a HSV color histogram x_{col} and a texton histogram x_{tex}

4 Technical Details/Notation

In our now treated Hierarchical Bayesian models , we denote the set of all i.i.d. latent variables/parameters by $\mathbf{Z} = \{z_1, ...,z_N\}$, all i.i.d. observed variables by $\mathbf{X} = \{x_1, ..., x_N\}$.

In the following chapters we build intuitive models, whose output data are our superpixels X. We want to train the model's parameters Z as good as possible to fit the data. As the true distributions p are intractable to work with, instead we try to approximate it by factorized distributions q. The gap/difference between two distributions p and q is well explained my the Kullback-Leibler Divergence $\mathbf{KL}(q\|p)$. We are interested in the posterior $p(\mathbf{Z}|\mathbf{X})$ as well as in the model evidence $p(\mathbf{X})$. We can decompose the log of the evidence in a sum of the lower bound and the KL divergence:

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathbf{KL}(q\|p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) ln\{\frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})}\} d\mathbf{Z}$$

$$\mathbf{KL}(q\|p) = -\int q(\mathbf{Z}) ln\{\frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})}\} d\mathbf{Z}$$

Proof:

1.

$$\begin{aligned} p(x,z) &= p(z|x) * p(x) \ln p(z|x) + \ln p(x) \\ \Leftrightarrow \ln p(x,z) &= \ln p(z|x) + \ln p(x) \end{aligned}$$

2.

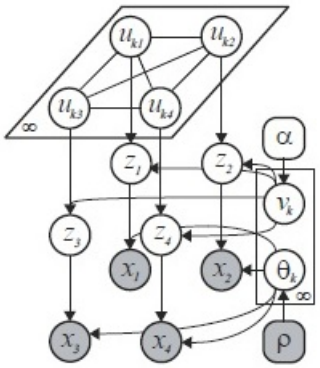
$$\mathcal{L}(q) = \int q(\mathbf{z}) \ln\{\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\} d\mathbf{z} = \int q(z) \ln p(x,z) d\mathbf{z} - \int q(z) \ln q(z) dz$$

We now use Result 1:

$$\begin{aligned} \mathcal{L}(q) &= \int q(z) \ln p(z|x) dz + \int q(z) \ln p(x) dz - \int q(z) \ln q(z) dz \\ &= \int q(z) \ln\{\frac{p(z|x)}{q(z)}\} dz + \ln p(x) \int q(z) dz \\ &= \mathbf{KL}(q\|p) + \ln p(x) \end{aligned}$$

As $lnp(\mathbf{X})$ is constant, we can maximize the *lower bound* $\mathcal{L}(q)$ w.r.t $q(z)$, which is equivalent to minimize the Kullback Leibler divergence $\mathbf{KL}(q\|p)$. Of course, if any choice for $q(z)$ is possible, the optimal solution occurs when $p(z|x) = q(z)$. Instead we consider a restricted family of factorized distribution $q(z)$ and choose the one minimizing the $\mathbf{KL}$.

5 Hirarchical Pitman-Yor Model Without Spatial Dependencies



5.1 Introduction

We first investigate J images and segment them jointly, neglecting all spatial dependencies between the superpixels. We assume that the different areas in the images are instances of potentially infinite many global object categories ($k = 1...K$). Each of these occurs with frequency/probability

$$\varphi \sim GEM(\gamma_a, \gamma_b),$$

where $GEM(\gamma_a, \gamma_b)$ is a Pitman-Yor Process constructed from stick proportions

$$\omega_k \sim Beta(1 - \gamma_a, \gamma_b + k\gamma_a)$$

The advantage of using such a stochastic process is, that we avoid to choose a fixed number of categories, but implicitely learning the number of categories by only regarding those categories over a certain frequency.

Each category k has an associated appearance model:

$$\theta_k = (\theta_k^{tex}, \theta_k^{col})$$

are the parameters of Multinomial distributions on the superpixels x^{tex}, x^{col}. Because of conjugacy between Multinomials and Dirichlet distributions the parameter have priors

$$\theta_k^{tex} \sim Dir(\rho^{tex}), \theta_k^{col} \sim Dir(\rho^{col})$$

We can now imagine region t in image j beeing produced by category k_{jt} as a sample from φ:

$$k_{jt} \sim \varphi = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_K \end{pmatrix}$$

Each image j is subdivided in a potentially infinite set of segments/regions ($t = 1...T$). Region t occupies the random proportion $\pi_{jt}$of the area in image j:

$$\pi_j \sim GEM(\alpha_a, \alpha_b),$$

which is another Pitman-Yor Process with stick proportions

$$\nu_{jt} \sim Beta(1 - \alpha_a, \alpha_b + k\alpha_a)$$

Again we use a stochastic process to work around the model selection task to choose the number of segments in an image. Instead the model subdivides the area in an image according to probabilities, which implicitely sets the number of segments in the image.

Proportional to the region size is the probability of a superpixel i in image j belonging to region t_{ji}:

$$t_{ji} \sim \pi_j = \begin{pmatrix} \pi_{j1} \\ \vdots \\ \pi_{jT} \end{pmatrix}$$

5.2 The model:

From these intuitions we build a fully hierarchical Bayesian model

$$p(\mathbf{x}, \mathbf{t}, \nu, \mathbf{k}, \omega, \theta) = p(\mathbf{x}|\mathbf{t}, \mathbf{k}, \theta)p(\mathbf{t}|\nu)p(\mathbf{k}|\omega)p(\nu)p(\omega)p(\theta),$$

where

$$p(\mathbf{x}|\mathbf{t},\mathbf{k},\theta) = \prod_{j=1}^{J}\prod_{i=1}^{N_j}[\prod_{t=1}^{T}\prod_{k=1}^{K}(Mult(x_{ji}^{tex}|\theta_k^{tex})Mult(x_{ji}^{col}|\theta_k^{col}))^{1_{\{k_{ji}=k\}}1_{\{t_{ji}=t\}}}]$$

$$p(\mathbf{t}|\nu) = \prod_{j=1}^{J}\prod_{i=1}^{N_j}(\prod_{t=1}^{T}(\nu_{jt}\prod_{s=1}^{t-1}(1-\nu_{js})))^{1_{\{t_{ji}=t\}}}$$

$$p(\mathbf{k}|\omega) = \prod_{j=1}^{J}\prod_{t=1}^{T}(\prod_{k=1}^{K}(\omega_k\prod_{l=1}^{k-1}(1-\omega_l))^{1_{\{k_{jt}=k\}}})$$

$$p(\nu|\alpha) = \prod_{j=1}^{J}\prod_{t=1}^{T}Beta(1-\alpha_a,\alpha_b+t\alpha_a)$$

$$p(\omega|\gamma) = \prod_{k=1}^{K}Beta(1-\gamma_a,\gamma_b+k\gamma_a)$$

$$p(\theta|\rho) = \prod_{k=1}^{K}Dir(\rho^{tex})Dir(\rho^{col})$$

5.3 Variational Inference

5.3.1 Factorization

We will perform the standard mean field approach to calculate the factorized distributions q

$$q(\mathbf{t},\nu,\mathbf{k},\omega,\theta) = \prod_{k=1}^{K}q(\omega_k)q(\theta_k)\prod_{j=1}^{J}[\prod_{t=1}^{T}q(\nu_{jt})q(k_{jt})\prod_{i=1}^{N_j}q(t_{ji})]$$

It is possible to perform free form approximations for all variables. This means that any distribution is allowed for the distributions q. We see now how maximizing the lower bound in a free-form approximation naturally leads to a typical form of equation.

$$\begin{aligned}\mathcal{L}(q) &= \int q(\mathbf{\Theta})ln\{\frac{p(\mathbf{X},\mathbf{\Theta})}{q(\Theta)}\}d\mathbf{\Theta} \\ &= \int \underbrace{\prod_{j=1}^{M}}_{\#\text{ of all variables}} q_j(\Theta_j)\{p(\mathbf{X},\mathbf{\Theta}) - \sum_{j=1}^{M}lnq_i(\Theta_i)\}d\mathbf{\Theta}\end{aligned}$$

Let us now marginalize out for one variable:

$$\begin{aligned}\mathcal{L}(q_j) &= \int q_j(\Theta_j)[\underbrace{\int ln[p(\mathbf{X},\mathbf{\Theta})]\prod_{i\neq j} q_i(\Theta_i)d\Theta_i]}_{\substack{=E_{\prod_{i\neq j} q(\Theta_i)}[ln[p(\mathbf{X},\mathbf{\Theta})]] \\ =:ln\tilde{p}(\mathbf{x},\mathbf{\Theta_j})}} d\Theta_j - \int q_j(\Theta_j) lnq_j(\Theta_j)d\Theta_j \\ &\quad \underbrace{-\sum_{i\neq j}^{M}\int q_i(\Theta_i)lnq_i(\Theta_i)d\Theta_i}_{=const w.r.t.\Theta_j} \\ &= \int q_j(\Theta_j) ln[\frac{\tilde{p}(\mathbf{x},\mathbf{\Theta_j})}{q_j(\Theta_j)}]d\Theta_j - const\end{aligned}$$

We see that for a variable Θ_j the optimal distribution which maximizes the lower bound is at:

$$ln(q^*(\Theta_j)) = E_{\prod_{j\neq i} q(\Theta_i)}[ln(p(x,\Theta))]$$

In a free-form approximation this form is valid for all variables Θ_j . We know calculate the distributions for the different variables of the models according to this pattern. See the Appendix for details.

5.3.2 Summary Of Results:

1.

$$q*(t) = \prod_{j=1}^{J}\prod_{i=1}^{N_j}\prod_{t=1}^{T} r_{jit}^{t_{jit}},$$

where

$$\begin{aligned} r_{jit} &= \frac{\rho_{jit}}{\sum_{s=1}^{T}\rho_{jis}} \\ ln\rho_{jit} &= E_{q(\nu_j)}[ln(\nu_{jt})] + \sum_{s=1}^{t-1} E_{q(\nu_j)}[ln(1-\nu_{js})] \\ &\quad + \sum_{k=1}^{K} E_{q(k)}[k_{jtk}][\sum_{m=1}^{M_1} x_{jim}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M_2} x_{jin}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]]\end{aligned}$$

2.

$$q*(\nu) = \prod_{j=1}^{J}\prod_{t=1}^{T-1}\frac{\Gamma(\beta_{ajt}+\beta_{bjt})}{\Gamma(\beta_{ajt})\Gamma(\beta_{bjt})}\nu_{jt}^{(\beta_{ajt-1})}(1-\nu_{jt})^{(\beta_{bjt}-1)},$$

where

$$\begin{aligned}\beta_{ajt}: &= \sum_{i=1}^{N_j} r_{jit} - \alpha_a + 1 \\ \beta_{bjt} &:= \sum_{i=1}^{N_j}\sum_{s=t+1}^{T-1} r_{jis} + \alpha_b + t\alpha_a\end{aligned}$$

3.

$$q*(k) \quad = \quad \prod_{j=1}^{J}\prod_{t=1}^{T}\prod_{k=1}^{K} d_{jtk}^{k_{jtk}},$$

where

$$\begin{aligned} d_{jtk} &:= \frac{\delta_{jtk}}{\sum_{l=1}^{K}\delta_{jtl}} \\ ln\delta_{jtk} &:= E_{q(\omega_k)}[ln\omega_k] + \sum_{l=1}^{k-1} E_{q(\omega_l)}[ln(1-\omega_l)] \\ &\quad + \sum_{i=1}^{N_j} r_{jit}[\sum_{m=1}^{M_1} x_{jim}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M_2} x_{jin}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]] \end{aligned}$$

4.

$$q*(\omega) \quad = \prod_{k=1}^{K-1} \quad \frac{\Gamma(\varepsilon_{ak}+\varepsilon_{bk})}{\Gamma(\varepsilon_{ak})\Gamma(\varepsilon_{bk})}\omega_k^{(\varepsilon_{ak}-1)}(1-\omega_k)^{(\varepsilon_{bk}-1)},$$

where

$$\begin{aligned} \varepsilon_{ak} : &= \sum_{j=1}^{J}\sum_{t=1}^{T} d_{jtk} - \gamma_a + 1 \\ \varepsilon_{bk} &:= \sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{l=1}^{K-1} d_{jtl} + \gamma_b + k\gamma_a \end{aligned}$$

5.

$$\begin{aligned} q*(\theta) \quad = \quad & \prod_{k=1}^{K} Dir(\theta_k^{tex}|(...., \underbrace{\underbrace{\rho^{tex} + \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} r_{jit} d_{jtk} x_{jim}^{tex}}_{\text{m-th entry in M1-dim parameter vector}}}_{\phi_{km}^{tex}}, ...)) \\ & *Dir(\theta_k^{col}|(...., \underbrace{\underbrace{\rho^{col} + \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} r_{jit} d_{jtk} x_{jin}^{col}}_{\text{n-th entry in M2-dim parameter vector}}}_{\phi_{kn}^{col}}, ...)) \end{aligned}$$

5.3.3 Towards Algorithm

The algorithm yields to approach p by q during a learning process, which is done by maximizing for the variables (M-Step) and evaluation of the Lower bound based on the current variable values(E-Step). To be able to maximize for the variables, we first wish to evaluate certain terms contained in $q*(t)$:

$$
\begin{aligned}
ln\tilde{\nu}_{jt1} &:= E_{q(\nu_{jt})}[ln\nu_{jt}] = \psi(\beta_{ajt}) - \psi(\beta_{ajt} + \beta_{bjt}) \\
ln\tilde{\nu}_{jt2} &:= E_{q(\nu_{jt})}[ln(1-\nu_{jt})] = \psi(\beta_{bjt}) - \psi(\beta_{ajt} + \beta_{bjt}) \\
ln\tilde{\omega}_{k1} &:= E_{q(\omega_k)}[ln\omega_k] = \psi(\varepsilon_{ak}) - \psi(\varepsilon_{ak} + \varepsilon_{bk}) \\
ln\tilde{\omega}_{k2} &:= E_{q(\omega_k)}[ln(1-\omega_k)] = \psi(\varepsilon_{bk}) - \psi(\varepsilon_{ak} + \varepsilon_{bk}) \\
ln\tilde{\theta}^{tex}_{km} &:= E_{q(\theta^{tex}_{km})}[ln\theta^{tex}_{km}] = \psi(\phi^{tex}_{km}) - \psi(\sum_{m=1}^{M_1} \phi^{tex}_{km}) \\
ln\tilde{\theta}^{col}_{kn} &:= E_{q(\theta^{col}_{kn})}[ln\theta^{col}_{kn}] = \psi(\phi^{col}_{kn}) - \psi(\sum_{n=1}^{M_2} \phi^{col}_{kn}),
\end{aligned}
$$

where ψ is the Digamma function. Because

$$E_{q(k_{jkt})}[k_{jtk}] = d_{jtk}$$

$$E_{q(t)}[t_{jit}] = r_{jit}$$

We see, that $\mathbf{d}, \mathbf{r}$ are nothing but (softbinning) category and region assignments. We will initialize them randomly or, to speed convergence up, by a lower level segmentation algorithm like K Means.

We are now able to evaluate r_{jit}:

$$r_{jit} \propto exp(ln\tilde{\nu}_{jt1} + \sum_{s=1}^{t-1} ln\tilde{\nu}_{js2} + \sum_{k=1}^{K} d_{jtk}\{\sum_{m=1}^{M_1} x^{tex}_{jim} ln\tilde{\theta}^{tex}_{km} + \sum_{n=1}^{M_2} x^{col}_{jin}\tilde{\theta}^{col}_{kn}\})$$

Next we evaluate the lower bound $\mathcal{L}$:

$$
\begin{aligned}
\mathcal{L}(q) \quad = \quad & \underbrace{\sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T}\sum_{k=1}^{K} d_{jtk} r_{jit}\{\sum_{m=1}^{M_1} x_{jim}^{tex} ln\tilde{\theta}_{km}^{tex} + \sum_{n=1}^{M_2} x_{jin}^{col} ln\tilde{\theta}_{kn}^{col}\}}_{E_q[ln(p(x|t,k,\theta))]} \\
& + \underbrace{\sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T-1} r_{jit}\{ln\tilde{\nu}_{jt1} + \sum_{s=1}^{t-1} ln\tilde{\nu}_{js2}\}}_{E_q[ln(p(t|\nu))]} \\
& + \underbrace{\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{k=1}^{K-1} d_{jtk}\{ln\tilde{\omega}_{k1} + \sum_{l=1}^{k-1} ln\tilde{\omega}_{l2}\}}_{E_q[ln(p(k|\omega))]} \\
& + \underbrace{\sum_{j=1}^{J}\sum_{t=1}^{T-1}(ln\frac{\Gamma(1-\alpha_a+\alpha_b+k\alpha_a)}{\Gamma(1-\alpha_a)\Gamma(\alpha_b+k\alpha_a)}) + (-\alpha_a)ln\tilde{\nu}_{jt1} + (\alpha_b+k\alpha_a-1)ln\tilde{\nu}_{jt2}}_{E_q[ln(p(\nu))]} \\
& + \underbrace{\sum_{k=1}^{K-1}\{(-\gamma_a)ln\tilde{\omega}_{k1} + (\gamma_b+k\gamma_a-1)ln\tilde{\omega}_{k2} + ln(\frac{\Gamma(1-\gamma_a+\gamma_b+k\gamma_a)}{\Gamma(1-\gamma_a)\Gamma(\gamma_b+k\gamma_a)})\}}_{E_q[ln(p(\omega))]} \\
& + \underbrace{\sum_{k=1}^{K}\{\sum_{m=1}^{M_1}\rho^{tex} ln\tilde{\theta}_{km}^{tex} + \sum_{n=1}^{M_2}\rho^{col} ln\tilde{\theta}_{kn}^{col}\}}_{E_q[ln(p(\theta))]} \\
& - \underbrace{\sum_{j=1}^{J}\sum_{t=1}^{T-1}\{ln(\frac{\Gamma(\beta_{ajt}+\beta_{bjt})}{\Gamma(\beta_{ajt})\Gamma(\beta_{bjt})}) + (\beta_{ajt}-1)ln\tilde{\nu}_{jt1} + (\beta_{bjt}-1)ln\tilde{\nu}_{jt2}\}}_{E_q[ln(q(\nu))]} \\
& - \underbrace{\sum_{k=1}^{K-1}\{ln(\frac{\Gamma(\varepsilon_{ak}+\varepsilon_{bk})}{\Gamma(\varepsilon_{ak})\Gamma(\varepsilon_{bk})}) + (\varepsilon_{ak}-1)ln\tilde{\omega}_{k1} + ((\varepsilon_{bk}-1)ln\tilde{\omega}_{k2})\}}_{E_q[ln(q(\omega))]} \\
& - \underbrace{\sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T-1} r_{jit} lnr_{jit}}_{E_q[ln(q(t))]} \\
& - \underbrace{\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{k=1}^{K-1} d_{jtk} lnd_{jtk}}_{E_q[ln(q(k))]} \\
& - \underbrace{\sum_{k=1}^{K}\{\sum_{m=1}^{M_1}\phi_{km}^{tex} ln\tilde{\theta}_{km}^{tex} + \sum_{n=1}^{M_2}\phi_{kn}^{col} ln\tilde{\theta}_{kn}^{col}\}}_{E_q[ln(q(\theta))]}
\end{aligned}
$$

5.3.4 Algorithm

1. set lower bound : $\mathcal{L}_0(q) = -\infty$, set ε to small value, set $\alpha$and γ to fixed values according to power law statistics, initialize $\mathbf{d}, \mathbf{r}$

   (a) d_{jt} : membership of region t in image j to category d_{jt}

   (b) r_{ji}: membership of superpixel i in image j to region r_{ji}

2. do the following steps, while $\mathcal{L}_i(q) - \mathcal{L}_{i-1}(q) > \varepsilon$

   (a) Evaluate parameters of $q(\nu)$:

$$\begin{aligned} \beta_{ajt} &= \sum_{i=1}^{N_j} r_{jit} - \alpha_a + 1 \\ \beta_{bjt} &= \sum_{i=1}^{N_j} \sum_{s=t+1}^{T-1} r_{jis} + \alpha_b + t\alpha_a \end{aligned}$$

   (b) Evalute parameters of $q(\omega)$:

$$\begin{aligned} \varepsilon_{ak} &= \sum_{j=1}^{J} \sum_{t=1}^{T} d_{jtk} - \gamma_a + 1 \\ \varepsilon_{bk} &= \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{l=1}^{K-1} d_{jtl} + \gamma_b + k\gamma_a \end{aligned}$$

   (c) Evaluate parameters of $q(\theta)$:

$$\begin{aligned} \phi_{km}^{tex} &= \rho^{tex} + \sum_{j=1}^{J} \sum_{i=1}^{N_j} \sum_{t=1}^{T} r_{jit} d_{jtk} x_{jim}^{tex} \\ \phi_{kn}^{col} &= \rho^{col} + \sum_{j=1}^{J} \sum_{i=1}^{N_j} \sum_{t=1}^{T} r_{jit} d_{jtk} x_{jin}^{col} \end{aligned}$$

   (d) Evaluate parameters of $q(t)$:

$$r_{jit} \propto exp(ln\tilde{\nu}_{jt1} + \sum_{s=1}^{t-1} ln\tilde{\nu}_{js2} + \sum_{k=1}^{K} d_{jtk} \{ \sum_{m=1}^{M_1} x_{jim}^{tex} ln\tilde{\theta}_{km}^{tex} + \sum_{n=1}^{M_2} x_{jin}^{col} \tilde{\theta}_{kn}^{col} \})$$

   i.
$$ln\tilde{\nu}_{jt1} := \psi(\beta_{ajt}) - \psi(\beta_{ajt} + \beta_{bjt})$$

   ii.
$$ln\tilde{\nu}_{js2} := \psi(\beta_{bjs}) - \psi(\beta_{ajs} + \beta_{bjs})$$

iii.

$$ln\tilde{\theta}^{tex}_{km} := \psi(\phi^{tex}_{km}) - \psi(\sum_{m=1}^{M_1} \phi^{tex}_{km})$$

iv.

$$ln\tilde{\theta}^{col}_{kn} := \psi(\phi^{col}_{kn}) - \psi(\sum_{n=1}^{M_2} \phi^{col}_{kn})$$

,where r_{jit}has to be normalized for each image j

(e) Evaluate parameters of $q(k)$:

$$d_{jik} \propto exp(\quad ln\tilde{\omega}_{k1} + \sum_{l=1}^{k-1} ln\tilde{\omega}_{l2} + \sum_{i=1}^{N_j} r_{jit}[\sum_{m=1}^{M_1} ln\tilde{\theta}^{tex}_{km} + \sum_{n=1}^{M_2} x^{col}_{jin} ln\tilde{\theta}^{col}_{kn}])$$

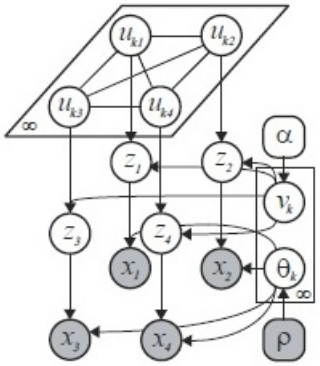
, which again has to be normalized.

(f) Evaluate lower bound $\mathcal{L}$ with actual parameters

5.4 Comment

We derived an algorithm to train our model with closed form parameter updates, which will enable a quick fitting of our model to the data. We also expect the model to build shared appearance categories like sky, trees, mountains... The main disadvantage of this model, beside the very huge number of equations, which are rich sources of errors during implementation, is that one of the most important properties of objects, their spatial continuity, is not taken into account at all. In our next model we first abandon to consider multiple images jointly, but we take the spatial aspect of objects in a single image into account.

6 Dependent Pitman-Yor Model For One Image



6.1 Introduction:

We now want to take into account the spatial relationship of nearby superpixels. Samples from Gaussian Processes of dimension N (= #superpixels) are smooth mountains over the superpixel centers. Sudderth et. al. proposed to assign a superpixel i to a category k, if the value of the category-specific Gaussian Process at datapoint i falls under a category-specific threshold $\bar{\nu}_k = \phi^{-1}(\nu_k)$. Nearby superpixels are highly likely to have similar values in a GP, and are therefore probable to be assigned to the same category. But how do we transform this intuition into a Bayesian model? First we assume the GPs to be zero mean with a kernel $\mathcal{K}$:

$$\mathbf{u_k} \sim \mathbf{GP}(\mathbf{0}, \mathcal{K})$$

From this follows that the value of a GP at the location of the center of superpixel i is distributed

$$u_{ki} \sim N(0,1)$$

An image potentially consists of infinite categories, in Bayesian models often described via a stochastic process as mentioned above, we again use a Pitman-Yor Process with a stick breaking construction. Here,

$$\varphi_k = \nu_k * \prod_{l=1}^{k-1}(1-\nu_l)$$

is the proportion category k takes within the image. But this is equivalent to the probability to assign a superpixel i to this category , $P(Z_i = k)$. Beginning with $k = 1$ we get the first *stick breaking proportion*

$$\nu_1 = \varphi_1 = P(z_i = 1)$$

For $k = 2$:

$$\begin{aligned} \varphi_2 &= P(z_i = 2) = & \nu_2 * (1-\nu_1) \\ &= & \nu_2 * (P(z_i \neq 1)) \\ \iff \nu_2 &= & \frac{P(z_i = 2)}{P(z_i \neq 1)} \\ &= & P(z_i = 2 | z_i \neq 1) \end{aligned}$$

Inductively follows that for an arbitrary category k we have:

$$\nu_k = P(z_i = k | z_i \neq k-1, ...1)$$

The stick breaking proportion ν_k is thus the conditional probability of assigning a superpixel to category k having rejected categories $1, ..., k-1$. But how can we bring this together with our first intuition about thresholded category assignment? We wanted to assign a superpixel i to a category k, if the value of the GP of category k at the position of i, $u_{ki} \sim N(0,1)$ falls below $\phi^{-1}(\nu_k)$. Let's again begin with $k = 1$:

$$\begin{aligned} P(z_i = 1) &= P(u_{1i} < \phi^{-1}(\nu_1)) \\ &= P(\phi(u_{1i}) < \nu_1) \\ &= \nu_1 \end{aligned}$$

since $\phi(u_{ki})$ is uniformly distributed on $[0,1]$, the range which is relevant for ν. This matches our previous results about category probability of $k = 1$.

For $k = 2$ we get:

$$\begin{aligned} P(z_i = 2) &= \nu_2 * (1 - \nu_1) \\ &= \nu_2 * (1 - P(\phi(u_{1i}) < \nu_1)) \\ &= \nu_2 * (P(\phi(u_{1i}) \geq \nu_1)) \\ &= P(\phi(u_{2i}) < \nu_2) * (P(\phi(u_{1i}) \geq \nu_1)) \\ &= P(u_{2i} < \phi^{-1}(\nu_2)) * P(u_{1i} \geq \phi^{-1}(\nu_1)) \\ &= P(u_{2i} < \bar{\nu}_2) * P(u_{1i} \geq \bar{\nu}_1) \end{aligned}$$

again because of the property of $\phi(u_{ki})$ being uniformly distributed on $[0, 1]$. Mathematically this now meets our intuition about the thresholded assignment, as the probability of a superpixel i being assigned to category $k = 2$, is the probability of u_{1i} being over the threshold $\bar{\nu}_1$ while u_{2i} having a value smaller $\bar{\nu}_1$. By induction this again can be generalized to the case of an arbitrary category k:

$$\begin{aligned} P(z_i = k) &= \nu_k \prod_{l=1}^{k-1} (1 - \nu_l) \\ &= P(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} P(u_{li} \geq \bar{\nu}_l) \end{aligned}$$

We have found that the probability of superpixel i being assigned to an arbritrary category k is the probability of category k to be the first GP at position i smaller than its category threshold $\bar{\nu}_k$. In our model the conditional distribution $p(\mathbf{z}|\bar{\nu},\mathbf{u})$ becomes a $0-1$ distribution, because the assignment z_i is perfectly determined having given $\bar{\nu}, \mathbf{u}$. Furthermore in contrast to the Dependent Pitman Yor Model treated in the former chapter, we do not distinguish between regions an categories, it is the same here. The reason is that when comparing multiple images we would like to recover shared properties, like the categories sky, trees,.. being discovered in most natural scenes. Investigating only one image has the focus to segment it only by its area distribution. This makes only one stochastic process (Pitman Yor process) necessary.

6.1.1 Variable transformation

We transform the Beta distributed variable $\nu_k \sim Beta(.|1-\alpha_a, \alpha_b + k\alpha_a)$ into a random variable $\bar{\nu}_k = \phi^{-1}(\nu_k)$ representing our class assignment threshold. We know that before and after the variable transformation a differential area should have the same value in both variable spaces:

$$\begin{aligned} p(\phi^{-1}(\nu_k)) * \partial\phi^{-1}(\nu_k) &= p(\nu_k) * \partial\nu_k \\ p(\phi^{-1}(\nu_k)) &= p(\nu_k) * \frac{\partial\nu_k}{\partial\phi^{-1}(\nu_k)} \\ p(\bar{\nu}_k) &= p(\phi(\bar{\nu}_k)) * \frac{\partial\phi(\bar{\nu}_k)}{\partial\bar{\nu}_k} \\ &= Beta(\phi(\bar{\nu}_k)|1 - \alpha_a, \alpha_b + k\alpha_a) * \frac{\partial\phi(\bar{\nu}_k)}{\partial\bar{\nu}_k} \end{aligned}$$

Knowing that the derivative of the normal cumulative distribution function is a normal distribution $N(.|0,1)$ of the argument, yields:

$$p(\bar{\nu}_{\mathbf{k}}|\alpha) = Beta(\phi(\bar{\nu}_k)|1-\alpha_a, \alpha_b + k\alpha_a)N(\bar{\nu}_k|0,1)$$

6.2 The model:

$$p(\mathbf{x},\mathbf{z},\bar{\nu},\mathbf{u},\theta) = p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{z}|\nu,\mathbf{u})p(\bar{\nu}|\alpha)p(\mathbf{u})p(\theta|\rho), \tag{1}$$

where

$$p(\mathbf{x}|\mathbf{z},\theta) = \prod_{i=1}^{N}(\prod_{k=1}^{K}[Mult(x_i^{tex}|\theta_k^{tex})Mult(x_i^{col}|\theta_k^{col})]^{1_{\{z_i=k\}}}) \tag{2}$$

$$p(\mathbf{z}|\bar{\nu},\mathbf{u}) = \prod_{i=1}^{N}(\prod_{k=1}^{K-1}[1_{\{u_{ki}<\bar{\nu}_k\}}]^{1_{\{k=z_i\}}}[1_{\{u_{ki}>\bar{\nu}_k\}}]^{1_{\{k<z_i\}}}) \tag{3}$$

$$p(\bar{\nu}|\alpha) = \prod_{k=1}^{K-1} Beta(\phi(\bar{\nu}_k)|1-\alpha_a, \alpha_b + k\alpha_a)N(\bar{\nu}_k|0,1) \tag{4}$$

$$p(\mathbf{u}|\mathcal{K}) = \prod_{k=1}^{K-1} GP(0,\mathcal{K}_k) \tag{5}$$

$$p(\theta|\rho) = \prod_{k=1}^{K} Dir(\theta^{tex}|\underbrace{\rho^{tex}}_{\text{M1-dim.}})Dir(\theta^{col}|\underbrace{\rho^{col}}_{\text{M2-dim.}}) \tag{6}$$

6.3 Variational Inference

6.3.1 Factorization

Unfortunately it is impossible to calculate free- form approximations for all variables. Sudderth proposes to put Gaussians on $\bar{\nu}$ and $\mathbf{u}$. Furthermore the factorized distribution of the label, $q(\mathbf{z})$, has the same Heaviside construction as the real distribution $p(\mathbf{z}|\bar{\nu},\mathbf{u})$. Later we will see that the two distributions on z will cancel in the lower bound. Because of the truncation, $q(\bar{\nu}_K) = 1$, this makes the K-th hidden surface superfluous. The factorization of the approximate distributions becomes.:

$$q(\mathbf{z},\bar{\nu},\mathbf{u},\theta) = \prod_{i=1}^{N}\underbrace{q(z_i|\bar{\nu},\mathbf{u})}_{=p(z_i|\bar{\nu},u)}\prod_{k=1}^{K}\underbrace{q(\theta_k^{tex}|\eta_k^{tex})q(\theta_k^{col}|\eta_k^{col})}_{free-form}\prod_{k=1}^{K-1}\{\underbrace{q(\bar{\nu}_k)}_{N(\bar{\nu}_k|v_k,\delta_k)}\prod_{i=1}^{N}\underbrace{q(u_{ki})}_{N(u_{ki}|\mu_{ki},\lambda_{ki})}\}$$

6.3.2 Lower bound

As shown above, we wish to maximize the lower bound in order to approximate p by q. Here, Θ stands for the variables and X for the data. We will now see (in line 3) how $p(\mathbf{z}|\bar{\nu},\mathbf{u})$ cancels against $q(\mathbf{z}|\bar{\nu},\mathbf{u})$ in the ln braces, while being retained in the expectation:

$$
\begin{aligned}
\mathcal{L}(q) &= \int q(\mathbf{\Theta}) ln\{\frac{p(\mathbf{X},\mathbf{\Theta})}{q(\Theta)}\} d\mathbf{\Theta} && (7)\\
&= E_{q(\mathbf{z},\bar{\nu},\mathbf{u},\theta)}[ln(\frac{p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{z}|\bar{\nu},\mathbf{u})p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)}{q(\mathbf{z}|\bar{\nu},\mathbf{u})\prod_{k=1}^{K-1}\{q(\bar{\nu}|v_k,\delta_k)\prod_{i=1}^{N}q(u_{ki}|\mu_{ki},\lambda_{ki})\})q(\theta)})]\\
&= E_{q(\mathbf{z},\bar{\nu},\mathbf{u},\theta)}[ln(\frac{p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)}{\prod_{k=1}^{K-1}\{q(\bar{\nu}|v_k,\delta_k)\prod_{i=1}^{N}q(u_{ki}|\mu_{ki},\lambda_{ki})\})q(\theta)})]\\
&= \int \underbrace{\prod_{j=1}^{M}}_{\#\text{ of all variables}} q_j(\Theta_j)\{lnp(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho) - \underbrace{\sum_{i=1}^{M} lnq_i(\Theta_i)}_{\Theta_i\neq\mathbf{z}}\}d\mathbf{\Theta}
\end{aligned}
$$

Let us again marginalize out for one variable:

$$
\begin{aligned}
\mathcal{L}(q_j) &= \int q_j(\Theta_j)[\underbrace{\int ln[p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)]\prod_{i\neq j}q_i(\Theta_i)d\Theta_i]}_{\substack{=E_{\prod_{i\neq j}q(\Theta_i)}[ln[p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)]]\\ =:ln\tilde{p}(\mathbf{x},\mathbf{\Theta_j})}} d\Theta_j - \int q_j(\Theta_j)\underbrace{lnq_j(\Theta_j)}_{=0\Leftrightarrow\Theta_j=z_i} d\Theta_j\\
&\quad \underbrace{-\sum_{i\neq j}^{M}\int q_i(\Theta_i)lnq_i(\Theta_i)d\Theta_i}_{=const w.r.t.\Theta_j} && (8)\\
&= \int q_j(\Theta_j)ln[\frac{\tilde{p}(\mathbf{x},\mathbf{\Theta_j})}{q_j(\Theta_j)}]d\Theta_j - const && (9)
\end{aligned}
$$

Marginalizing for several variables $\hat{\Theta}$is analogue:

$$
\begin{aligned}
\mathcal{L}(q(\hat{\Theta})) &= \int \prod_{\Theta_j\in\hat{\Theta}} q_j(\Theta_j)[\underbrace{\int ln[p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)]\prod_{\Theta_i\notin\hat{\Theta}}q_i(\Theta_i)d\Theta_i]}_{\substack{=E_{\prod_{\Theta_i\notin\hat{\Theta}}q(\Theta_i)}[ln[p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)]]\\ =:ln\tilde{p}(\mathbf{x},\hat{\mathbf{\Theta}})}} d\Theta_j\\
&\quad -\int \prod_{\Theta_j\in\hat{\Theta}} q(\Theta_j)\underbrace{lnq(\Theta_j)}_{=0\Leftrightarrow\Theta_j=z_i} d\Theta_j - const && (10)\\
&= \int q(\hat{\Theta})ln[\frac{\tilde{p}(\mathbf{x},\hat{\mathbf{\Theta}})}{q(\hat{\Theta})}]d\hat{\Theta} - const && (11)
\end{aligned}
$$

,which is only defined for parameter sets $\hat{\Theta}$ not including z_i.

We see that marginalizing out the lower bound for one or several variables results in negative Kullback-Leibler divergences between distributions $\tilde{p}$, which are expectations of the distributions $p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho)$

over all but the variables to be optimized, and the joint distribution q of the parameter/-s to be optimized. Our goal was to approximate p by q as good as possible, which is achieved if the negative KL is zero. This is the case if $lnq^* = ln\tilde{p}$. This technique of optimization is called free form optimization. Unfortunately this is not always possible, as some integrals/expectations of $\tilde{p}$ are hard to calculate. Instead, we assume some variables to have a certain distribution (because of simplicity often Gaussians). Then all what is left to do is to optimize the lower bound w.r.t. the parameters of these distributions (e.g. for Gaussians: moments). In our case, we apply free form optimization on $\theta^{tex}, \theta^{col}$and the discribed strategy on the moments of the Gaussian distributions of $\bar{\nu}, \mathbf{u}$.

6.3.3 Calculating the free form distribution of $q(\theta)$

Fortunately at least for θ it is possible to calculate a free form distribution. To do so, we use the result from above and marginalize out the lower bound for the variables θ:

$$\mathcal{L}(q(\theta)) \propto \int q(\theta) ln[\frac{\tilde{p}(\mathbf{x}, \theta)}{q(\theta)}] d\theta$$

We again see, that this is the negative $KL(q(\theta)||\tilde{p}(\mathbf{x}, \theta))$, which is because of the negative algebraic sign ≤ 0 . The maximum is hence at zero, when:

$$\begin{aligned}
lnq^*(\theta) &= ln\tilde{p}(\mathbf{x}, \theta) \\
&\propto E_{q(z|\bar{\nu},u)q(u)q(\bar{\nu})}[lnp(x|z,\theta)] + lnp(\theta|\rho)] \\
&\propto \sum_{i=1}^{N}\sum_{k=1}^{K} E_{q(z|\bar{\nu},u)q(u)q(\bar{\nu})}[z_{ik}] * \{\sum_{m=1}^{M_1} x_{im} ln\theta_{km}^{tex} + \sum_{n=1}^{M_2} x_{in} ln\theta_{kn}^{col}\} \\
&\quad + \sum_{k=1}^{K}\sum_{m=1}^{M_1} (\rho_m^{tex} - 1) ln\theta_{km}^{tex} + \sum_{k=1}^{K}\sum_{n=1}^{M_2} (\rho_n^{col} - 1) ln\theta_{kn}^{col}
\end{aligned}$$

To solve the first part of the right hand side we use the assumption that

$$q(u_{ki}) = N(u_{ki}|\mu_{ki}, \lambda_{ki})$$

$$q(\bar{\nu}_k) = N(\bar{\nu}_k|v_k, \delta_k)$$

Under this assumption the probability $q(u_{ki} < \bar{\nu}_k)$ can be calculated:

$$q(u_{ki} < \bar{\nu}_k) = q(\underbrace{\bar{\nu}_k - u_{ki}}_{\sim N(.|v_k - \mu_{ki}, \delta_k + \lambda_{ki})} > 0)$$

Be

$$X \sim N(.|0, 1)$$

Then we have

$$\sqrt{\delta_k + \lambda_{ki}} * X + (v_k - \mu_{ki}) \sim N(.|v_k - \mu_{ki}, \delta_k + \lambda_{ki})$$

From this follows that

$$\begin{aligned} q(u_{ki} < \bar{\nu}_k) &= q(\sqrt{\delta_k + \lambda_{ki}} * X + (v_k - \mu_{ki}) > 0) \\ &= q(X > -\frac{(v_k - \mu_{ki})}{\sqrt{\delta_k + \lambda_{ki}}}) \\ &= 1 - q(X \leq -\frac{(v_k - \mu_{ki})}{\sqrt{\delta_k + \lambda_{ki}}}) \\ &= 1 - \phi(-\frac{(v_k - \mu_{ki})}{\sqrt{\delta_k + \lambda_{ki}}}) \\ &= \phi(\frac{v_k - \mu_{ki}}{\sqrt{\delta_k + \lambda_{ki}}}) \end{aligned}$$

Using this and the symmetry of the normal cummulative distribution function, saying $1 - \phi(x) = \phi(-x)$, we can solve:

$$\begin{aligned} E_{q(z|\bar{\nu},u)q(u)q(\bar{\nu})}[z_{ik}] &= E_{q(u)q(\bar{\nu})}[1_{\{u_{ki} < \bar{\nu}_k\}} \prod_{l=1}^{k-1} 1_{\{u_{li} \geq \bar{\nu}_l\}}] \\ &= q(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} q(u_{li} > \bar{\nu}_l) \\ &= \phi(\frac{v_k - \mu_{ki}}{\sqrt{\delta_k + \lambda_{ki}}}) \prod_{l=1}^{k-1} \phi(-\frac{v_l - \mu_{li}}{\sqrt{\delta_l + \lambda_{li}}}), \end{aligned}$$

where, because of truncation according to Blei and Jordan, we have

$$q(u_{Ki} < \bar{\nu}_K) = 1$$

The rest is just sorting for the variables θ:

$$
\begin{aligned}
ln q^*(\theta) &\propto \sum_{i=1}^{N}\sum_{k=1}^{K} q(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} q(u_{li} > \bar{\nu}_l) * \{\sum_{m=1}^{M_1} x_{im} ln\theta_{km}^{tex} + \sum_{n=1}^{M_2} x_{in} ln\theta_{kn}^{col}\} \\
&\quad + \sum_{k=1}^{K}\sum_{m=1}^{M_1} (\rho_m^{tex} - 1) ln\theta_{km}^{tex} + \sum_{k=1}^{K}\sum_{n=1}^{M_2} (\rho_n^{col} - 1) ln\theta_{kn}^{col} \\
&\propto \sum_{k=1}^{K}\sum_{m=1}^{M_1} ln\theta_{km}^{tex} \{\sum_{i=1}^{N} x_{im}^{tex} * q(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} q(u_{li} > \bar{\nu}_l) + \rho_m^{tex} - 1\} \\
&\quad + \sum_{k=1}^{K}\sum_{n=1}^{M_2} ln\theta_{kn}^{tex} \{\sum_{i=1}^{N} x_{in}^{tex} * q(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} q(u_{li} > \bar{\nu}_l) + \rho_n^{col} - 1\} \\
q^*(\theta) &= \prod_{k=1}^{K} Dir(\theta_k^{tex} | (..., \underbrace{\rho_m^{tex} + \sum_{i=1}^{N} [x_{im}^{tex} * q(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} q(u_{li} > \bar{\nu}_l)]}_{=:\eta_{km}^{tex} \text{ (m-th entry in M1-dim. vector)}}, ...)) \\
&\quad * Dir(\theta_k^{col} | (..., \underbrace{\rho_n^{col} + \sum_{i=1}^{N} [x_{in}^{col} * q(u_{ki} < \bar{\nu}_k) \prod_{l=1}^{k-1} q(u_{li} > \bar{\nu}_l)]}_{:=\eta_{kn}^{col} \text{(n-th entry in M2-dim. vector)}}, ...)) \\
&= \prod_{k=1}^{K} Dir(\theta_k^{tex} | (\eta_{k1}^{tex}, ..., \eta_{kM_1}^{tex})) * Dir(\theta_k^{col} | (\eta_{k1}^{col}, ..., \eta_{kM_2}^{col}))
\end{aligned}
$$

So, as the result of our free form optimization of $q(\theta)$ we again get Dirichlet distributions (because of conjugacy), this time for every k with the datapoints weighted with the normalized specific probability to lay in class k, represented by the normal cumulative distritution functions ϕ

6.3.4 Calculating the distributions $q(\bar{\nu}_k) \prod_{i=1}^{N} q(u_{ki})$

As we cannot perform free form optimization here, we assume $q(\bar{\nu}_k)$ and $q(u_{ki})$ to be Gaussians. We now write the lower bound and then optimize w.r.t. the moments of the variables of interest $\bar{\nu}_k, u_{ki}$:

$$
\begin{aligned}
\mathcal{L}(q) &= E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)}[ln(p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho))] - \int q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)\{\sum_{k=1}^{K-1}[ln(q(\bar{\nu_k})\prod_{i=1}^{N}q(u_{ki})) \\
&\quad +ln(q(\theta_k))]\}d\bar{\nu}_k du_{ki} d\theta dz \\
&= E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)}[ln(p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho))] - \int q(u)q(\bar{\nu})\{\sum_{k=1}^{K-1}[ln(q(\bar{\nu_k})\prod_{i=1}^{N}q(u_{ki}))]\}d\bar{\nu}_k du_{ki} \\
&\quad -\sum_{k=1}^{K}\int q(\theta_k)lnq(\theta_k)d\theta_k \\
&= \underbrace{E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)}[ln(p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho))]}_{\mathcal{L}_1} - \underbrace{\sum_{k=1}^{K-1}\int q(u_k)q(\bar{\nu}_k)\{[ln(q(\bar{\nu_k})\prod_{i=1}^{N}q(u_{ki}))]\}d\bar{\nu}_k du_{ki}}_{\mathcal{L}_2} \\
&\quad \underbrace{-\sum_{k=1}^{K}\int q(\theta_k)lnq(\theta_k)d\theta_k}_{\mathcal{L}_3} \\
&= \mathcal{L}_1 - \mathcal{L}_2 - \mathcal{L}_3
\end{aligned}
$$

The lower bound to be maximized subdivides in intuitive parts: $\mathcal{L}_1$ are negative cross entropies, representing dissimilarity between p and q. A well-known strategy to minimize this gap between the real and the approximate distribution is called *cross-entropy-minimization.* Taking the negative of the cross entropies and maximizing is analogue. $\mathcal{L}_2$ (and $\mathcal{L}_3$) are entropies of the approximate distributions and are to be maximized to maximize the information carried in these distributions. $\mathcal{L}_3$ does not change during optimization and hence we do not regard it.

Lower bound Detailed calculation of the lowerbound can be found in the Appendix. Below are only the results of this calculation to keep the reading uncrowded. The lowerbound becomes:

$$
\begin{aligned}
\mathcal{L}(q) &= \mathcal{L}_{11} + \mathcal{L}_{12} + \mathcal{L}_{13} + \mathcal{L}_{14} - \mathcal{L}_2 \\
&= \underbrace{\sum_{i=1}^{N}\sum_{k=1}^{K}\phi(\frac{v_k-\mu_{ki}}{\sqrt{\delta_k+\lambda_{ki}}})\prod_{l=1}^{k-1}\phi(-\frac{v_l-\mu_{li}}{\sqrt{\delta_l+\lambda_{li}}})\{\sum_{m=1}^{M1}x_{im}^{tex}(\psi(\eta_{km}^{tex})-\psi(\sum_{m=1}^{M_1}\eta_{km}^{tex}))+\sum_{n=1}^{M2}x_{in}^{col}E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]}_{\mathcal{L}_{11}..} \\
&\quad \underbrace{+\sum_{n=1}^{M2}x_{in}^{col}(\psi(\eta_{kn}^{col})-\psi(\sum_{n=1}^{M_2}\eta_{kn}^{col}))+ln(\frac{[\sum_{m=1}^{M_1}x_{im}^{tex}]!}{\prod_{m=1}^{M_1}x_{im}^{tex}!})+ln(\frac{[\sum_{n=1}^{M_2}x_{in}^{col}]!}{\prod_{n=1}^{M_2}x_{in}^{col}!})\}}_{..\mathcal{L}_{11}} \\
&\quad \underbrace{+\sum_{k=1}^{K-1}(-\frac{1}{2}ln[(2\pi e)^N]-\frac{1}{2}(ln(det(\mathcal{K}_k))+TR(\mathcal{K}_k^{-1}\Sigma_{q_k})+(-\mu_{\mathbf{k}})^T\mathcal{K}_k^{-1}(-\mu_{\mathbf{k}})-N))}_{\mathcal{L}_{12}} \\
&\quad \underbrace{+\sum_{k=1}^{K-1}\{ln(\frac{\Gamma(1-\alpha_a+\alpha_b+k\alpha_a)}{\Gamma(1-\alpha_a)\Gamma(\alpha_b+k\alpha_a)})-\alpha_a ln(\phi(\frac{v_k}{\sqrt{1+\delta_k}}))+(\alpha_b+k\alpha_a-1)ln(\phi(-\frac{v_k}{\sqrt{1+\delta_k}}))}_{\mathcal{L}_{13}..} \\
&\quad \underbrace{-\frac{1}{2}ln(2\pi e)-\frac{v_k^2}{2}-\frac{\delta_k}{2}-\frac{1}{2}\}}_{..\mathcal{L}_{13}} \\
&\quad \underbrace{+\sum_{k=1}^{K}\frac{1}{2}ln[(2\pi e)^{N+1}\delta_k\prod_{i=1}^{N}\lambda_{ki}]}_{\mathcal{L}2}
\end{aligned}
$$

Optimization: We wanted to optimize w.r.t. the moments $v_k, \delta_k, \mu_{ki}, \lambda_{ki}$ of the Gaussians $q(\bar{\nu}_k) = N(\bar{\nu}_k|v_k,\delta_k)$ and $q(u_{ki}) = N(u_{ki}|\mu_{ki},\lambda_{ki})$. In doing so $\mathcal{L}_{14}$and $\mathcal{L}_3$will not change in value, as they are independent of the moments. Furthermore in $\mathcal{L}_{12}$ the log of products of λ_{ki} cancel in the first and the second term, leaving both terms to be constant and hence irrelevant for optimization. Cutting away all constant terms $\mathcal{L}_{12}$ becomes:

$$
\mathcal{L}_{12}^* := \sum_{k=1}^{K-1}(-\frac{1}{2}(TR(\mathcal{K}_k^{-1}\Sigma_{q_k})+(-\mu_{\mathbf{k}})^T\mathcal{K}_k^{-1}(-\mu_{\mathbf{k}}))),
$$

where we can simplyfy the trace term, knowing that Σ_{q_k} is diagonal with entries $\Sigma_{q_k}(i,i) = \lambda_{kii}^2$ and $\mathcal{L}_{12}^*$ becomes:

$$
\mathcal{L}_{12}^* = \sum_{k=1}^{K-1}(-\frac{1}{2}(\sum_{i=1}^{N}\mathcal{K}_k^{-1}(i,i)*\lambda_{ki}+(-\mu_{\mathbf{k}})^T\mathcal{K}_k^{-1}(-\mu_{\mathbf{k}})))
$$

In $\mathcal{L}_{13}$ the normalizer, the $\frac{1}{2}ln(\delta_k^2)$ terms and constants as well as constants in $\mathcal{L}_2$ are cancelled for optimization:

$$
\begin{aligned}
\mathcal{L}_{13}^{*} &= \sum_{k=1}^{K-1}\{-\alpha_a ln(\phi(\frac{v_k}{\sqrt{1+\delta_k}})) + (\alpha_b + k\alpha_a - 1)ln(\phi(-\frac{v_k}{\sqrt{1+\delta_k}})) - (\frac{v_k^2}{2} + \frac{\delta_k}{2})\} \\
\mathcal{L}_{2}^{*} &= -\sum_{k=1}^{K-1}\frac{1}{2}ln[\delta_k \prod_{i=1}^{N}\lambda_{ki}]
\end{aligned}
$$

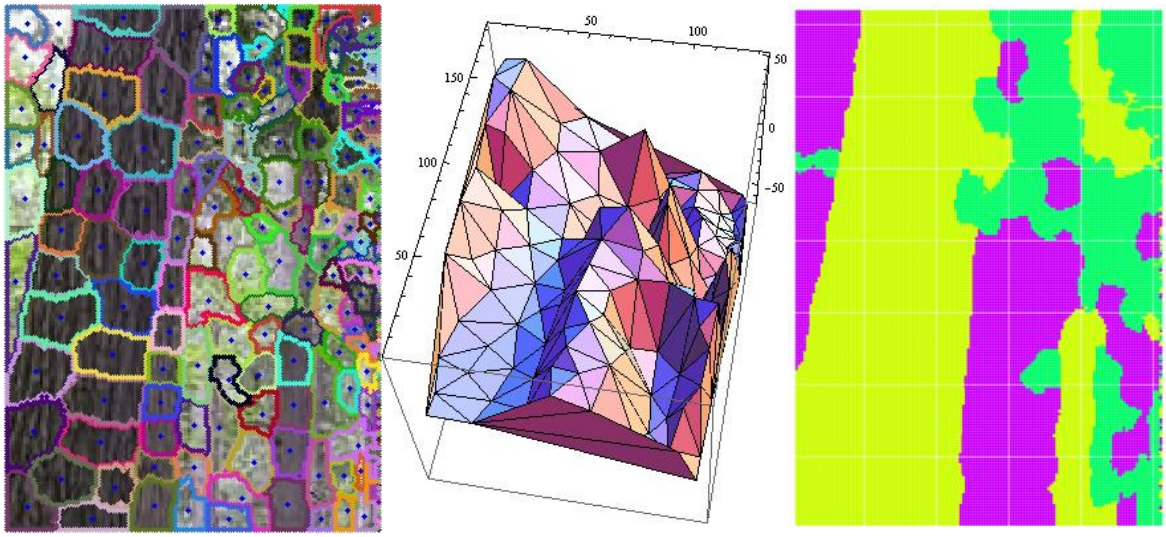
The Lowerbound relevant for optimization contains only terms that change during optimization:

$$
\begin{aligned}
\mathcal{L}^{*}(q) &= \mathcal{L}_{11} + \mathcal{L}_{12}^{*} + \mathcal{L}_{13}^{*} - \mathcal{L}_{2}^{*} \\
&= \underbrace{\sum_{i=1}^{N}\sum_{k=1}^{K}\phi(\frac{v_k - \mu_{ki}}{\sqrt{\delta_k + \lambda_{ki}}})\prod_{l=1}^{k-1}\phi(-\frac{v_l - \mu_{li}}{\sqrt{\delta_l + \lambda_{li}}})\{\sum_{m=1}^{M1} x_{im}^{tex}(\psi(\eta_{km}^{tex}) - \psi(\sum_{m=1}^{M_1}\eta_{km}^{tex}))}_{\mathcal{L}_{11}..} \\
&\quad \underbrace{+\sum_{n=1}^{M2} x_{in}^{col}(\psi(\eta_{kn}^{col}) - \psi(\sum_{n=1}^{M_2}\eta_{kn}^{col})) + ln(\frac{[\sum_{m=1}^{M_1} x_{im}^{tex}]!}{\prod_{m=1}^{M_1} x_{im}^{tex}!}) + ln(\frac{[\sum_{n=1}^{M_2} x_{in}^{col}]!}{\prod_{n=1}^{M_2} x_{in}^{col}!})\}}_{..\mathcal{L}_{11}} \\
&\quad \underbrace{+\sum_{k=1}^{K-1}(-\frac{1}{2}(\sum_{i=1}^{N}\mathcal{K}_k^{-1}(i,i) * \lambda_{ki} + (-\mu_{\mathbf{k}})^T\mathcal{K}_k^{-1}(-\mu_{\mathbf{k}})))}_{\mathcal{L}_{12}^{*}} \\
&\quad \underbrace{+\sum_{k=1}^{K-1}\{-\alpha_a ln(\phi(\frac{v_k}{\sqrt{1+\delta_k}})) + (\alpha_b + k\alpha_a - 1)ln(\phi(-\frac{v_k}{\sqrt{1+\delta_k}})) - \frac{v_k^2}{2} - \frac{\delta_k}{2}\}}_{\mathcal{L}_{13}^{*}} \\
&\quad \underbrace{+\sum_{k=1}^{K-1}\frac{1}{2}ln[\delta_k\prod_{i=1}^{N}\lambda_{ki}]}_{-\mathcal{L}_{2}^{*}}
\end{aligned}
$$

All the terms of the lowerbound $\mathcal{L}_1$ are negative cross entropies. Maximizing negative cross entropies is equivalent to minimizing cross entropy, which is a common strategy to minimize the difference between two distributions, in this case fitting p by q. More in detail, $\mathcal{L}_{11}$represents the difference between the data likelihood and the model. $\mathcal{L}_{12}^{*}$ stands for the difference between a zero mean $GP(0, \mathcal{K})$ and the joint distribution of independent $q(u_{ki})$ over $i = 1...N$, forcing q to behave *gaussian*. $\mathcal{L}_{13}^{*}$is the difference between $p(\bar{\nu})$ and the Gaussian $q(\bar{\nu})$, enshuring q to fit the transformed stick breaking proportions as good as possible. The other relevant term, $-\mathcal{L}_2^{*}$is the entropy of the joint distribution $q(\bar{\nu})q(\mathbf{u})$.

We will use the Lowerbound $\mathcal{L}^{*}(q)$ to optimize the moments using a gradient descent strategy, as calculating gradients for each moment is easily done. We use Mathematica to do a numerical optimization using a Conjugate Gradient optimizer.

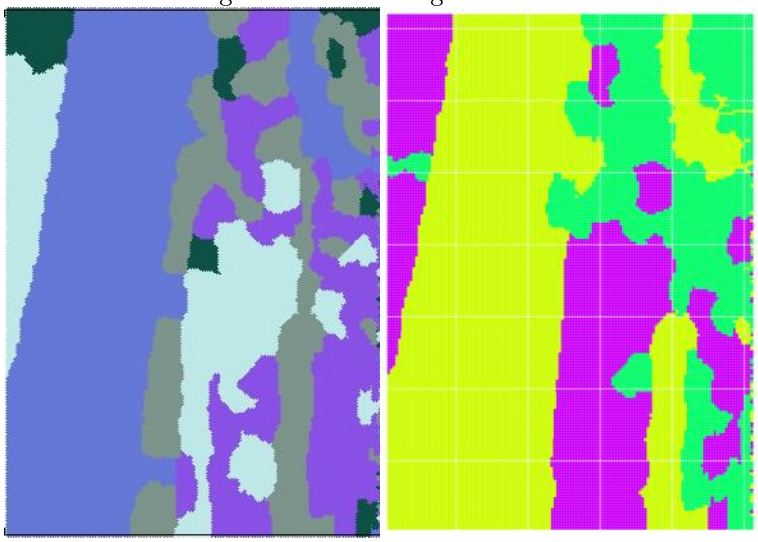
6.4 Results:



The optimization of the lower bound was made in Mathematica and turned out to be time consuming, as even for small image with about $N = 130$ superpixels and and a truncated number of categories of $K = 5$, we have to optimize w.r.t. $N * (K - 1) = 520$ means and variances of the $\mathbf{u}$ (and $K - 1 = 4$ means and variances of the $\bar{\nu}$). The author suggests to subdivide the image in about $N = 1000$ superpixels which would have resulted in several thousands of variables to be optimized.

6.4.1 Comparison to KMeans clustering

To measure the results of the models data fitting, I compared the segmentation of the model with a K Means clustering segmentation, which does not take any spatial relationship between superpixels nor any area distribution of segments in the image into account.



In fact, segmentation looked a lot smoother in the Bayesian model (right image) while at the same time the problem of model selection (e.g. number of segments in the image) was solved implicitely. In contrast the KMeans clustering result for K=5 segments looks a lot more patchy. It is hard to associate the clustering with categories like trunk, leafs of background. This can be done easyly in the right image: The yellow part represents trunks, the green leafs and the pink summerizes the rest as background. Although the Bayesian result merges the very similar trunks on the left, the model is an improvement on the way of intuitive, human-like image understanding. Investigating only one separate image makes it impossible for the algorithm to discover segments which consist of very different textures. Like a tree, which has trunk and leafs, cannot be discovered as one segment. Applying the model to several similar natural scenes should enable the algorithm even to learn this linking between trunk and leafs to one segment. The extension of our model hence is the model discussed in sketches below.

7 Dependent Pitman-Yor Model on multiple images

7.1 Introduction

For the most sophisticated model we take J images, having N_j superpixels each, and jointly segment them, using a shared appearance model θ. The idea is, that a region t in image j is an instance of one of the global categories k. The probability of region t in image j being an instance of category k (Object Label Frequencies) follows a Pitman-Yor Process $GEM(\gamma_a, \gamma_b)$:

$$\begin{aligned}
\varphi &\sim GEM(\gamma_a, \gamma_b), \text{ constructed from stick proportions } w_k \\
\omega_k &\sim Beta(1-\gamma_a, \gamma_b + k\gamma_a) \\
k_{jt} &\propto \varphi = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_K \end{pmatrix} \text{, where } k_{ji} \epsilon \{1....K\} \text{ is the category assignment}
\end{aligned}$$

The probability of superpixel i in image j being assigned to region t is intuitively proportional to the region size. The area distribution in the images follow Pitman-Yor Processes $GEM(\alpha_a, \alpha_b)$. As in the last model we introduce spatial dependency by Gaussian processes for every region in every image:

$$\begin{aligned}
u_{tj} &\propto GP(0, \mathcal{K}_{tj}) \\
t_{ji} &= min(t | u_{jti} < \bar{\nu}_{jt})), \text{ where } t_{ji} \epsilon \{1....T\} \text{ is the region assignment} \\
\bar{\nu}_{jt} &\propto N(\bar{\nu}_{jt}|0,1) * Beta(\phi(\bar{\nu}_{jt})|1-\alpha_a, \alpha_b + t\alpha_a),
\end{aligned}$$

where ϕ is the normal cdf.

7.2 The model

$$p(\mathbf{x}, \mathbf{t}, \mathbf{k}, \bar{\nu}, \mathbf{u}, \theta) = p(\mathbf{x}|\mathbf{t}, \mathbf{k}, \theta) p(\mathbf{t}|\bar{\nu}, \mathbf{u}) p(\bar{\nu}|\alpha) p(\mathbf{k}|\omega) p(\omega|\gamma) p(\mathbf{u}) p(\theta|\rho)$$

$$p(\mathbf{x}|\mathbf{t}, \mathbf{k}, \theta) = \prod_{j=1}^{J} \prod_{i=1}^{N_j} (\prod_{t=1}^{T} \prod_{k=1}^{K} (Mult(x_{ji}^{tex}|\theta_k^{tex}) Mult(x_{ji}^{col}|\theta_k^{col}))^{1_{\{k_{jt}=k\}} 1_{\{t_{ji}=t\}}})$$

$$p(\mathbf{t}|\bar{\nu},\mathbf{u}) = \prod_{j=1}^{J}\prod_{i=1}^{N_j}(\prod_{t=1}^{T}[1_{\{u_{jti}<\bar{\nu}_{jt}\}}]^{1_{\{t_{ji}=t\}}}[1_{\{u_{t_{ji}}>\bar{\nu}_{ji}\}}]^{1_{\{t_{ji}>t\}}})$$

$$p(\bar{\nu}|\alpha) = \prod_{j=1}^{J}\prod_{t=1}^{T} Beta(\phi(\bar{\nu}_{jt})|1-\alpha_a, \alpha_b + t\alpha_a)N(\bar{\nu}_{jt}|0,1)$$

$$p(\mathbf{k}|\omega) = \prod_{j=1}^{J}\prod_{t=1}^{T}(\prod_{k=1}^{K}(\omega_k \prod_{l=1}^{k-1}(1-\omega_l))^{1_{\{k_{jt}=k\}}})$$

$$p(\omega|\gamma) = \prod_{k=1}^{K} Beta(1-\gamma_a, \gamma_b + k\gamma_a)$$

$$p(\mathbf{u}|\mathcal{K}) = \prod_{j=1}^{J}\prod_{t=1}^{T} GP(0, \mathcal{K}_{jt})$$

$$p(\theta|\rho) = \prod_{k=1}^{K} Dir(\theta^{tex}|\underbrace{\rho^{tex}}_{\text{M1-dim.}})Dir(\theta^{col}|\underbrace{\rho^{col}}_{\text{M2-dim.}})$$

The result of this model should build categories over images while at the same time considering spatial continuity of segments.

8 Conclusion

The approach proposed by Sudderth to understand the content of images, which do not contain rigid forms and make template matching fail, is intuitive: The spatial continuity of segments build from superpixels using Gaussian processes and the spatial distribution of the segments following power law distributions governed by Pitman Yor processes describe the transformation of natural observations into powerful mathematical models. It is really astonishing how little foreknowledge about the image is needed to segment it. On the other hand the model is computational highly complex: All the Gaussian processes for every category (number depends on the truncation) are connected and need to be optimized jointly. It shure needs more knowledge in mathematical optimization to implement the optimization of the lower bound more efficiently. In my opinion it is quenstionable if these intuitions described above can't be used in a little more simple model, which does not explode in performance with the number of superpixels as well as the number of categories.

The results of the DPY model compared to those of a simple KMeans clustering of the superpixels in fact showed improvement, as the categories were less patchy and more connected. On the other hand overlapping

similar structures like the two trunks in the example are melted to one cataegory, as they are similar and close to one another, which suggests the Gaussian process, that is is one category. The intuition, that sharp edges in the image (like between the trunks) are often an indicator for a separation line between two categories, is not considered in the model: Where in the preprocessing step, the calculation of superpixels, this was dominant factor to separate superpixels, in the model superpixels are only described by their histograms of texture and color. The information of the separating edges is lost!

The approach described in the paper of Sudderth et. al. showed, that, beside simple pattern matching nowadays implemented in digital cameras or expencive cars, image segmentation of images that consist of non rigid forms like sky, trees and ground make much more sophisticated techniques necessary. Sudderth has shown one convincing way, maybe the spatial connectivity of superpixels could be implemented in a simpler but faster model.

9 Acknowledgements

I would like to thank my parents for supporting me throughout the long studies. I thank Prof. Manfred Opper and my girlfriend Caroline for their support and patience on different fields and finally Philipp Batz for helping me find errors in the jungle of formulas.

10 Appendix

10.0.1 Completing the Square

We want to get the mean and Cov out of a quadratic equation. We therefore consider, that the exponent in a general Gaussian distribution $N(\mathbf{x}|\mu, \mathbf{\Sigma})$ can be written

$-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) = -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu + const$, where *const* denotes everything not dependent of x. This enables us to recover the mean and Cov out of a quadratic equation.

10.1 Factorized distributions

In our models we restrict the family of distributions $q(z)$ the following:

- We partition the latent variables $\mathbf{Z}$ in disjoint groups Z_i, where $i = 1, ..., M$.
- We assume that the distribution q *factorizes* w.r.t. these groups:

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(Z_i)$$

- We make no further assumptions/restrictions about the factors $q_i(Z_i)$ and found the optimal solutions to be in the form

$$\ln q_j^*(Z_j) = E_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})] + const$$

taking the exp and normalizing, we get:

$$q_j^*(Z_j) = \frac{exp(E_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})])}{\int exp(E_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})])dZ_j}$$

The set of equations given by this equation for all $j = 1, ..., M$ represent a set of *consistency conditions* for the maximum of the lower bound $\mathcal{L}(q)$ s.t. the factorization constraint.

However, these equations do *not* represent an *explicit solution,* because the solution for one $q_j$depends on the expectations computed w.r.t. $\{q_{i\neq j}\}$!!!! We therefore seek a solution by first initalizing all q_i *appropriately* and then replace the factors $q_j$in turn with a revised estimate, given the right hand side of the above equation, evaluated using the current estimates for all the $\{q_{i\neq j}\}$! Convergence is garanteed, because bound is convex w.r.t. each q_i.

10.2 Properties of factorized approximations

Consider a bivariate Gaussian

$$p(z) = N(z|\mu, \Lambda^{-1}) \sim exp(-\frac{1}{2}(z-\mu)^T\Lambda(z-\mu)),$$

where

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

$$z = (z_1, z_2)$$

$$\Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}$$

If we apply the above equation to find the optimal $q_1^*(z_1)$, we note that only those terms retain, that have a functional dependence on z_1, the other terms are constants and can be absorbed by the *const* term:

$$\begin{aligned} \ln q_1^*(z_1) &= E_{z_2}[\ln p(z)] + const \\ &= E_{z_2}[-\frac{1}{2}(z_1-\mu_1)^2\Lambda_{11} - (z_1-\mu_1)\Lambda_{12}(z_2-\mu_2)] + const \\ &= E_{z_2}[-\frac{1}{2}(z_1^2 - 2z_1\mu_1 + \mu_1^2)\Lambda_{11} - \Lambda_{12}(z_1z_2 - \mu_1z_2 + \mu_1\mu_2 - z_1\mu_2)] + const \\ &= -\frac{1}{2}z_1^2\Lambda_{11} + z_1\mu_1\Lambda_{11} + z_1\mu_2\Lambda_{12} - E_{z_2}[z_2]\Lambda_{12}z_1 + const \end{aligned}$$

,where the *const* term swallowed all terms no depending on z_1.

We see that this is a quadratic function and so we can identify $q_1^*(z_1)$ as a Gaussian. We didn't make any assumptions about $q_1(z_1)$, but we derived this result from optimization of the KL divergence. From completing the square strategy we see, that the moments of the approximated function become:

$$Cov = \Lambda_{11}$$

$$m_1 = \mu_1 - \mu_2\Lambda_{12}\Lambda_{11}^{-1} - E_{z_2}[z_2]\Lambda_{12}\Lambda_{11}^{-1}$$

So we finally get:

$$q_1^*(z_1) = N(z_1|m_1, \Lambda_{11}^{-1})$$

And by symmetry

$$q_2^*(z_2) = N(z_2|m_2, \Lambda_{22}^{-1}),$$

where

$$m_2 = \mu_2 - \mu_2\Lambda_{21}\Lambda_{22}^{-1} - E_{z_1}[z_1]\Lambda_{21}\Lambda_{22}^{-1}$$

10.2.1 Results:

The problem is simple enough to find a closed form solution: As $E[z_1] = m_1$and $E[z_2] = m_2$ by definition, the two equations are satisfied, if $E[z_1] = \mu_1$and $E[z_2] = \mu_2$. Illustrating this (Bishop, p.468, Fig. 10.2a) we see that the mean of $p(z)$ is nicely captured by $q(z)$ but that the variance is controlled by the smallest variance of p and is significantly underestimated along the orthogonal direction. *In general, the approximations given by the factorized variational model are too compact!!!*

10.2.2 Comparison: reverse Kullback-Leibler divergence: minimize $\mathbf{KL}(p \parallel q)$ instead of $\mathbf{KL}(q \parallel p)$

(also used in *expectation propagation* framework)

Again we consider

$$q(\mathbf{z}) = \prod_{i=1}^{M} q_i(z_i)$$

The reverse KL is defined the following:

$$\mathbf{KL}(p \parallel q) = -\int p(\mathbf{z}) \ln \prod_{i=1}^{M} q_i(z_i) d\mathbf{z} + \int p(\mathbf{z}) \ln p(\mathbf{z}) d\mathbf{z},$$

where the last term is the entropy of $p(\mathbf{z})$ and can be seen as a constant:

$$\mathbf{KL}(p \parallel q) = -\int p(\mathbf{z}) \sum_{i=1}^{M} \{\ln q_i(z_i)\} dz_i + const$$

We now optimize w.r.t. each $q_j(z_j)$ using a Lagrange multiplier and get:

$$q_j^*(z_j) = \int p(\mathbf{z}) \prod_{i \neq j} dz_i = p(z_j)$$

,which is astonishingly a closed form solution, the solution is simply the marginal of p! A visualization (see corresponding Fig. 10.2b in Bishop) shows, that the mean is again captured, but the variance puts too much probability weight on space that has low probability.

▶The difference between the results for the different Kullback-Leibler Divergences can be understood the following:

- large contribution of $KL(\mathrm{q} \parallel \mathrm{p}) = -\int q(z) \ln(\frac{p(z)}{q(z)}) dz$ from regions of z, where both $p(z), q(z)$ are close to zero, but the ln 'overdramatizes' the gap by its nonlinearity. Minimizing $KL(\mathrm{q} \parallel \mathrm{p})$ thus leads to distributions $q(z)$, that *avoid* regions in which $p(z)$ is small ⇒over compact q

- $KL(\mathrm{p} \parallel \mathrm{q}) = -\int p(z) \ln(\frac{q(z)}{p(z)})dz$ is minimized by the q that is nonzero in regions where p is nonzero, because the $ln(0)$is the worst case.

10.3 HPY model (first model): Calculation of the factorized distributions

10.3.1 Calculating $q(\mathbf{t})$

$$\begin{aligned} ln(q*(\mathbf{t})) &= E_{q(\nu,\omega,\theta,k)}[ln(p(x,t,\nu,w,\theta))] \\ &= E_{q(\theta,k)}[ln(p(x|t,k,\theta))] + E_{q(\nu)}[ln(p(t|\nu)] + const \end{aligned}$$

for a single t_{ji} we have:

$$\begin{aligned} E_{q(\nu_j)}[ln(p(t_{ji}|\nu_j))] &= E_{q(\nu_j)}[\sum_{t=1}^{T}((ln(1-\nu_{jt}))^{1_{\{t<t_{ji}\}}}(lnv_{jt})^{1_{\{t=t_{ji}\}}})] \\ &= E_{q(\nu_j)}[\sum_{t=1}^{T}\sum_{s=t+1}^{T} t_{jis}(ln(1-\nu_{jt})) + t_{jit}ln\nu_{jt}] \\ &= \sum_{t=1}^{T} t_{jit}\{E_{q(\nu_j)}[ln(\nu_{jt})] + \sum_{s=1}^{t-1} E_{q(\nu_j)}[ln(1-\nu_{js})]\}, \end{aligned}$$

where $t_{jit} = \begin{cases} 1 & , t_{ji} = t \\ 0 & else \end{cases}$

for a single x_{ji} we have:

$$\begin{aligned} E_{q(\theta,k)}[ln(p(x_{ji}|t,k,\theta))] &= E_{q(\theta,k)}[\sum_{t=1}^{T}\sum_{k=1}^{K} k_{jtk}t_{jit}[ln(Mult(x_{ji}^{tex}|\theta_k^{tex})) + ln(Mult(x_{ji}^{col}|\theta_k^{col}))]] \\ &= E_{q(\theta,k)}[\sum_{t=1}^{T}\sum_{k=1}^{K} k_{jtk}t_{jit}[\sum_{m=1}^{M_1} x_{jim}^{tex} ln\theta_{km}^{tex} + \sum_{n=1}^{M_2} x_{jin}^{col} ln\theta_{kn}^{col}]] \\ &= \sum_{t=1}^{T}\sum_{k=1}^{K} E_{q(k)}[k_{jtk}]t_{jit}[\sum_{m=1}^{M_1} x_{jim}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M_2} x_{jin}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]] \end{aligned}$$

,where $k_{jtk} = \begin{cases} 1 & , k_{jt} = k \\ 0 & else \end{cases}$

Let us put the terms together:

$$\begin{aligned} ln(q*(t)) &\propto E_{q(\theta,k)}[ln(p(x|t,k,\theta))] + E_{q(\nu)}[ln(p(t|\nu)] \\ &\propto \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} t_{jit}\{E_{q(\nu_j)}[ln(\nu_{jt})] + \sum_{s=1}^{t-1} E_{q(\nu_j)}[ln(1-\nu_{js})] \\ &\quad + \sum_{k=1}^{K} E_{q(k)}[k_{jtk}][\sum_{m=1}^{M_1} x_{jim}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M_2} x_{jin}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]]\} \end{aligned}$$

We set everything in {} equal to $ln\rho_{jit}$ and get for $q(t)$:

$$q*(t) = \prod_{j=1}^{J}\prod_{i=1}^{N_j}\prod_{t=1}^{T} r_{jit}^{t_{jit}},$$

where $r_{jit} = \frac{\rho_{jit}}{\sum_{s=1}^{T}\rho_{jis}}$

10.3.2 Calculating $q(\nu)$

$$\begin{aligned} ln(q*(\nu)) &= E_{q(t,k,\omega,\theta)}[ln(p(x,t,\nu,\omega,\theta,k))] \\ &\propto E_{q(t)}[ln(p(t,|\nu))] + ln(p(\nu)) \\ &\propto E_{q(t)}[\sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} t_{jit} ln(\nu_{jt}\prod_{s=1}^{t-1}(1-\nu_{js}))] + ln(p(\nu)) \end{aligned}$$

Because of the Multinomial form of $q(t)$ we have $E_{q(t)}[t_{jit}] = r_{jit}$ and hence:

$$\begin{aligned} ln(q*(\nu)) &\propto \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T-1}\{r_{jit} ln\nu_{jt} + \sum_{s=t+1}^{T-1} r_{jis} ln(1-\nu_{jt})\} + \sum_{j=1}^{J}\sum_{t=1}^{T-1}\{ln\nu_{jt}(-\alpha_a) + ln(1-\nu_{jt})(\alpha_b + k\alpha_a - 1)\} \\ &\propto \sum_{j=1}^{J}\sum_{t=1}^{T-1}\{ln\nu_{jt}[\sum_{i=1}^{N_j} r_{jit} - \alpha_a] + ln(1-\nu_{jt})[\sum_{i=1}^{N_j}\sum_{s=t+1}^{T-1} r_{jis} + \alpha_b + k\alpha_a - 1]\} \end{aligned}$$

We set:

$$\beta_{ajt} := \sum_{i=1}^{N_j} r_{jit} - \alpha_a + 1$$

$$\beta_{bjt} := \sum_{i=1}^{N_j}\sum_{s=t+1}^{T-1} r_{jis} + \alpha_b + k\alpha_a$$

and get:

$$ln(q*(\nu)) \propto \sum_{j=1}^{J}\sum_{t=1}^{T-1}\{ln\nu_{jt}[\beta_{ajt} - 1] + ln(1-\nu_{jt})[\beta_{bjt} - 1]\},$$

which is the log of the unnormalized product of Beta distributions:

$$\begin{aligned} q*(\nu) &= \prod_{j=1}^{J}\prod_{t=1}^{T-1}\frac{\Gamma(\beta_{ajt}+\beta_{bjt})}{\Gamma(\beta_{ajt})\Gamma(\beta_{bjt})}\nu_{jt}^{(\beta_{ajt}-1)}(1-\nu_{jt})^{(\beta_{bjt}-1)} \\ &= \prod_{j=1}^{J}\prod_{t=1}^{T-1} Beta(\nu_{jt}|\beta_{ajt},\beta_{bjt}) \end{aligned}$$

10.3.3 Calculating $q(k)$

$$\begin{aligned} ln(q*(k)) &= E_{q(\nu,t,\omega,\theta)}[ln(p(\mathbf{x},\mathbf{t},\nu,\omega,\theta,\mathbf{k}))] \\ &\propto E_{q(t,\theta)}[ln(p(x|t,k,\theta))] + E_{q(\omega)}[ln(p(k|\omega))] \end{aligned}$$

for a single k_{jt}we have:

$$\begin{aligned} E_{q(\omega)}[ln(p(k_{jt}|\omega))] &= E_{q(\omega)}[\sum_{k=1}^{K-1}\sum_{l=k+1}^{K-1}\{k_{jtk}ln\omega_k + k_{jtl}ln(1-\omega_l)\}] \\ &= \sum_{k=1}^{K-1} k_{jtk}\{E_{q(\omega_k)}[ln\omega_k] + \sum_{l=1}^{k-1} E_{q(\omega_l)}[ln(1-\omega_l)]\} \end{aligned}$$

for a single x_{ji}we have:

$$E_{q(t_{ji},\theta)}[ln(p(x_{ji}|t_{ji},\theta))] \propto \sum_{t=1}^{T}\sum_{k=1}^{K-1} E_{q(t_{ji})}[t_{jit}]k_{jtk}\{\sum_{m=1}^{M_1} x_{jim}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M_2} x_{jin}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]\}$$

Putting the terms together one gets:

$$\begin{aligned} ln(q*(k)) \propto & \sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{k=1}^{K} k_{jtk}\{E_{q(\omega_k)}[ln\omega_k] + \sum_{l=1}^{k-1} E_{q(\omega_l)}[ln(1-\omega_l)] \\ & + \sum_{i=1}^{N_j} r_{jit}[\sum_{m=1}^{M_1} x_{jim}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M_2} x_{jin}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}]]\} \end{aligned}$$

We denote everything in $\{\}$ as $ln\delta_{jtk}$and finally get:

$$q*(k) = \prod_{j=1}^{J}\prod_{t=1}^{T}\prod_{k=1}^{K} d_{jtk}^{k_{jtk}},$$

where

$$d_{jtk} = \frac{\delta_{jtk}}{\sum_{l=1}^{K}\delta_{jtl}}$$

10.3.4 Calculating $q(\omega)$

$$\begin{aligned} ln(q*(\omega)) &= E_{q(t,\nu,k,\theta)}[ln(p(x,t,\nu,k,\omega,\theta))] \\ &\propto E_{q(k)}[ln(p(k|\omega))] + ln(p(\omega)) \\ &\propto E_{q(k)}[\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{k=1}^{K-1} k_{jtk}ln(\omega_k\prod_{l=1}^{k-1}(1-\omega_l))] + ln(p(\omega)) \end{aligned}$$

Because of the Multinomial form of $q(k)$, we have $E_{q(k_{jkt})}[k_{jtk}] = d_{jtk}$ and hence:

$$\begin{aligned} ln(q*(\omega)) &\propto \sum_{j=1}^{J}\sum_{t=1}^{T}\{\sum_{k=1}^{K-1} d_{jtk} ln\omega_k + \sum_{l=k+1}^{K-1} d_{jtl} ln(1-\omega_k))\} \\ &\quad + \sum\{\sum_{k=1}^{K-1}(\gamma_a) ln\omega_k + (\gamma_b + k\gamma_a - 1) ln(1-\omega_k)\} \\ &\propto \sum_{k=1}^{K-1}\{ln\omega_k[\sum_{j=1}^{J}\sum_{t=1}^{T} d_{jtk} - \gamma_a] + ln(1-\omega_k)[\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{l=1}^{K-1} d_{jtl} + \gamma_b + k\gamma_a - 1]\} \end{aligned}$$

We set:

$$\begin{aligned} \varepsilon_{ak} &:= \sum_{j=1}^{J}\sum_{t=1}^{T} d_{jtk} - \gamma_a + 1 \\ \varepsilon_{bk} &:= \sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{l=1}^{K-1} d_{jtl} + \gamma_b + k\gamma_a \end{aligned}$$

and finally get $q(\omega)$ in the form of Beta distributions:

$$q*(\omega) = \prod_{k=1}^{K} Beta(\omega_k|\varepsilon_{ak}, \varepsilon_{bk})$$

10.3.5 Calculating $q(\theta)$

$$\begin{aligned} ln(q*(\theta)) &= E_{q(t,\nu,k,\omega)}[ln(p(x,t,\nu,k,\omega,\theta))] \\ &\propto E_{q(t,k)}[p(x|t,k,\theta)] + ln(p(\theta)) \\ &\propto \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T}\sum_{k=1}^{K} E_{q(t)}[t_{jit}] E_{q(k)}[k_{jtk}]\{\sum_{m=1}^{M_1} x_{jim}^{tex} ln\theta_{km}^{tex} + \sum_{n=1}^{M_2} x_{jin}^{col} ln\theta_{kn}^{col}\} \\ &\quad + \sum_{k=1}^{K}\{\sum_{m=1}^{M_1}(\rho^{tex}-1) ln\theta_{km}^{tex} + \sum_{n=1}^{M_2}(\rho^{col}-1) ln\theta_{kn}^{col}\} \\ &\propto \sum_{k=1}^{K}\sum_{m=1}^{M_1} ln\theta_{km}^{tex}\{\rho^{tex} - 1 + \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} r_{jit} d_{jtk} x_{jim}^{tex}\} \\ &\quad + \sum_{k=1}^{K}\sum_{n=1}^{M_2} ln\theta_{kn}^{col}\{\rho^{col} - 1 + \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} r_{jit} d_{jtk} x_{jin}^{col}\} \end{aligned}$$

which again takes the form of a Dirichlet distribution:

$$q*(\theta) = \prod_{k=1}^{K} Dir(\theta_k^{tex}|(...., \underbrace{\rho^{tex} + \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} r_{jit} d_{jtk} x_{jim}^{tex}}_{\text{m-th entry in M1-dim parameter vector}} , ...)) * Dir(\theta_k^{col}|(...., \underbrace{\rho^{col} + \sum_{j=1}^{J}\sum_{i=1}^{N_j}\sum_{t=1}^{T} r_{jit} d_{jtk} x_{jin}^{col}}_{\text{n-th entry in M2-dim parameter vector}} , ...))$$

10.4 DPY model (second model): Calculation of Lowerbound

Calculating $\mathcal{L}_2$: $\mathcal{L}_2$ is the negative entropy of $q(\bar{\nu_k})\prod_{i=1}^{N} q(u_{ki})$. Because of the given gaussian form of the distributions, we have:

$$\begin{aligned}
q(\bar{\nu}_k)\prod_{i=1}^{N} q(u_{ki}) &= N(\bar{\nu}_k|v_k,\delta_k)\prod_{i=1}^{N} N(u_{ki}|\mu_{ki},\lambda_{ki}) \\
&= \frac{1}{\sqrt{2\pi\delta_k}} exp(-\frac{1}{2}\frac{(\bar{\nu}_k - v_k)^2}{\delta_k})\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\lambda_{ki}}} exp(-\frac{1}{2}\frac{(u_{ki}-\mu_{ki})^2}{\lambda_{ki}}) \\
&= \frac{1}{\delta_k\prod_{i=1}^{N}\lambda_{ki}(2\pi)^{\frac{N+1}{2}}} exp(-\frac{1}{2}(\frac{(\bar{\nu}_k - v_k)^2}{\delta_k} + \sum_{i=1}^{N}\frac{(u_{ki}-\mu_{ki})^2}{\lambda_{ki}})) \\
&= \frac{1}{Norm} exp(-\frac{1}{2}(\left(\begin{pmatrix} \bar{\nu}_k \\ u_{k1} \\ \vdots \\ u_{kN} \end{pmatrix} - \underbrace{\begin{pmatrix} v_k \\ \mu_{k1} \\ \vdots \\ \mu_{kN} \end{pmatrix}}_{\vec{m}} \right)^T \underbrace{\begin{bmatrix} \frac{1}{\delta_k} & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda_{k1}} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\lambda_{kN}} \end{bmatrix}}_{\Sigma^{-1}} \left(\begin{pmatrix} \bar{\nu}_k \\ u_{k1} \\ \vdots \\ u_{kN} \end{pmatrix} - \underbrace{\begin{pmatrix} v_k \\ \mu_{k1} \\ \vdots \\ \mu_{kN} \end{pmatrix}}_{\vec{m}} \right))) \\
q(\bar{\nu}_k)\prod_{i=1}^{N} q(u_{ki}) &= N(\begin{pmatrix} \bar{\nu}_k \\ u_{k1} \\ \vdots \\ u_{kN} \end{pmatrix} | \vec{m}, \Sigma)
\end{aligned}$$

The negative entropy of this $N+1$-dimensional Gaussian is:

$$\begin{aligned}
\mathcal{L}_2 &= \sum_{k=1}^{K-1} -H(\begin{pmatrix} \bar{\nu}_k \\ u_{k1} \\ \vdots \\ u_{kN} \end{pmatrix}) = \\
&= -\sum_{k=1}^{K-1}\frac{1}{2} ln[(2\pi e)^{N+1} det(\Sigma)] \\
&= -\sum_{k=1}^{K-1}\frac{1}{2} ln[(2\pi e)^{N+1}\delta_k\prod_{i=1}^{N}\lambda_{ki}]
\end{aligned}$$

Calculating $\mathcal{L}_1$ The more complicated part of the lower bound $\mathcal{L}(q)$ is the first part $\mathcal{L}_1$.

$$\begin{aligned}
\mathcal{L}_1 &= E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)}[ln(p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{u})p(\bar{\nu}|\alpha)p(\theta|\rho))] \\
&= E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)}\{[ln(p(\mathbf{x}|\mathbf{z},\theta)+ln(p(\mathbf{u}))+ln(p(\bar{\nu}|\alpha))+ln(p(\theta|\rho))]\} \\
&= \underbrace{E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)q(\theta)}\{ln(p(\mathbf{x}|\mathbf{z},\theta)\}}_{\mathcal{L}_{11}} + \underbrace{E_{q(u)}\{ln(p(\mathbf{u}))\}}_{\mathcal{L}_{12}} + \underbrace{E_{q(\bar{\nu})}\{ln(p(\bar{\nu}|\alpha))\}}_{\mathcal{L}_{13}} + \underbrace{E_{q(\theta)}\{ln(p(\theta|\rho))\}}_{\mathcal{L}_{14}}
\end{aligned}$$

for $\mathcal{L}_{11}$ we get:

$$\begin{aligned}
\mathcal{L}_{11} &= \sum_{i=1}^{N}\sum_{k=1}^{K}[E_{q(u)q(\bar{\nu})q(z|\bar{\nu},u)}[z_{ik}]\{\sum_{m=1}^{M1} x_{im}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M2} x_{in}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}] + ln(\frac{[\sum_{m=1}^{M_1} x_{im}^{tex}]!}{\prod_{m=1}^{M_1} x_{im}^{tex}!}) + ln(\frac{[\sum_{n=1}^{M_2} x_{in}^{col}]!}{\prod_{n=1}^{M_2} x_{in}^{col}!})\}] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\underbrace{\phi(\frac{v_k-\mu_{ki}}{\sqrt{\delta_k+\lambda_{ki}}})\prod_{l=1}^{k-1}\phi(-\frac{v_l-\mu_{li}}{\sqrt{\delta_l+\lambda_{li}}})}_{\text{see above..}}\{\sum_{m=1}^{M1} x_{im}^{tex} E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] + \sum_{n=1}^{M2} x_{in}^{col} E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}] + ln(\frac{[\sum_{m=1}^{M_1} x_{im}^{tex}]!}{\prod_{m=1}^{M_1} x_{im}^{tex}!}) + ln(\frac{[\sum}{\prod}
\end{aligned}$$

The expectation w.r.t. $ln(\theta_{km}^{tex})$ and $ln(\theta_{kn}^{col})$ are:

$$\begin{aligned}
E_{q(\theta_{km}^{tex})}[ln\theta_{km}^{tex}] &= \psi(\eta_{km}^{tex}) - \psi(\sum_{m=1}^{M_1}\eta_{km}^{tex}) \\
E_{q(\theta_{kn}^{col})}[ln\theta_{kn}^{col}] &= \psi(\eta_{kn}^{col}) - \psi(\sum_{n=1}^{M_2}\eta_{kn}^{col})
\end{aligned}$$

so $\mathcal{L}_{11}$ becomes:

$$\begin{aligned}
\mathcal{L}_{11} &= \sum_{i=1}^{N}\sum_{k=1}^{K}\phi(\frac{v_k-\mu_{ki}}{\sqrt{\delta_k+\lambda_{ki}}})\prod_{l=1}^{k-1}\phi(-\frac{v_l-\mu_{li}}{\sqrt{\delta_l+\lambda_{li}}})\{\sum_{m=1}^{M1} x_{im}^{tex}(\psi(\eta_{km}^{tex}) - \psi(\sum_{m=1}^{M_1}\eta_{km}^{tex})) \\
&\quad + \sum_{n=1}^{M2} x_{in}^{col}(\psi(\eta_{kn}^{col}) - \psi(\sum_{n=1}^{M_2}\eta_{kn}^{col})) + ln(\frac{[\sum_{m=1}^{M_1} x_{im}^{tex}]!}{\prod_{m=1}^{M_1} x_{im}^{tex}!}) + ln(\frac{[\sum_{n=1}^{M_2} x_{in}^{col}]!}{\prod_{n=1}^{M_2} x_{in}^{col}!})\}
\end{aligned}$$

Let us analyse this term: The Digamma terms $\psi(\eta_{km}^{tex}) - \psi(\sum_{m=1}^{M_1}\eta_{km}^{tex}))$ for a category k weight those entries in x_{im}^{tex} the more, the less likely they are produced by category k (with a negative sign), while at the same time the product of CDFs, which actually is a category probability, weight those error terms with the probability that this datapoint lies in this category k. Hence, as this negative term wishes to have a small value to maximize the lower bound, the term encourages low category probability for datapoints unlikely to be produced by category k. Let us investigate this category probability a bit closer: It consists of a product of CDFs, who have big values for positive arguments. More in detail, the kth. category probability is big, if $\mu_{ki} < v_k$ and $\mu_{li} > v_l$ for $l = 1....(k-1)$. But how do we prevent this term simply to be zero for all datapoints? This should be governed by the prior distributions on $\mathbf{u}$ and $\bar{\nu}$.

for $\mathcal{L}_{12}$ we get:

$$\begin{aligned}
\mathcal{L}_{12} &= \int q(\mathbf{u})\{ln(p(\mathbf{u}))\}d\mathbf{u} \\
&= \int \prod_{k=1}^{K-1} q(\mathbf{u}_k)\{lnp(\mathbf{u}_1) + \ldots + lnp(\mathbf{u}_k) + \ldots lnp(\mathbf{u}_{K-1})\} \prod_{k=1}^{K-1} d\mathbf{u}_k \\
&= \sum_{l=1}^{K-1} \int \prod_{k=1}^{K-1} q(\mathbf{u}_k) lnp(\mathbf{u}_l) d\mathbf{u}_k \\
&= \sum_{l=1}^{K-1} \int q(\mathbf{u}_l)\{\underbrace{\int (\prod_{k\neq l}^{K-1} q(\mathbf{u}_k)\{lnp(\mathbf{u}_l)\} \prod_{k\neq l}^{K-1} d\mathbf{u}_k\}}_{=lnp(\mathbf{u}_l)\underbrace{\int \prod_{k\neq l}^{K-1} q(\mathbf{u}_k)d\mathbf{u}_k}_{=1}} d\mathbf{u}_l \\
&= \sum_{l=1}^{K-1} \int q(\mathbf{u}_l) lnp(\mathbf{u}_l) d\mathbf{u}_l
\end{aligned}$$

As $\mathbf{u}_k$ is independent for two different k in both distributions p and q, this is a sum of negative cross entropies $\mathcal{H}(q,p)$ with

$$q(\mathbf{u}_k) = N(\begin{pmatrix} u_{k1} \\ \vdots \\ u_{kN} \end{pmatrix} | \begin{pmatrix} \mu_{k1} \\ \vdots \\ \mu_{kN} \end{pmatrix}, \underbrace{\begin{bmatrix} \lambda_{k1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{kN} \end{bmatrix}}_{\Sigma_{q_k}})$$

$$p(\mathbf{u}_k) = N(\begin{pmatrix} u_{k1} \\ \vdots \\ u_{kN} \end{pmatrix} | \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \mathcal{K}_k)$$

The negative cross entropy is defined as:

$$-\mathcal{H}(q(\mathbf{u}_k), p(\mathbf{u}_k)) = \int q(\mathbf{u}_k) ln(p(\mathbf{u}_k)) d\mathbf{u}_k$$

and also as:

$$-\mathcal{H}(q(\mathbf{u}_k), p(\mathbf{u}_k)) = -\mathcal{H}(q(\mathbf{u}_k)) - D_{KL}(q(\mathbf{u}_k)||p(\mathbf{u}_k)),$$

where $\mathcal{H}(q(\mathbf{u}_k))$ is the entropy of $q(\mathbf{u}_k)$ and $D_{KL}(q||p)$ the Kullback-Leibler-Divergence between $p(\mathbf{u}_k)$ and $q(\mathbf{u}_k)$. But for multivariate Gaussians of dimension N we know that:

$$\begin{aligned}\mathcal{H}(q(\mathbf{u}_k)) &= \frac{1}{2}ln[(2\pi e)^N det(\Sigma_{q_k})] \\ &= \frac{1}{2}ln[(2\pi e)^N \prod_{i=1}^{N} \lambda_{ki}]\end{aligned}$$

$$D_{KL}(q(\mathbf{u}_k)||p(\mathbf{u}_k)) = \frac{1}{2}(ln(\frac{det(\mathcal{K}_k)}{det(\Sigma_{q_k})}) + TR(\mathcal{K}_k^{-1}\Sigma_{q_k}) + (-\mu_\mathbf{k})^T\mathcal{K}_k^{-1}(-\mu_\mathbf{k}) - N)$$

So $\mathcal{L}_{12}$ becomes:

$$\begin{aligned}\mathcal{L}_{12} &= \sum_{k=1}^{K-1} -\mathcal{H}(q(\mathbf{u}_k), p(\mathbf{u}_k)) \\ &= \sum_{k=1}^{K-1} (-\mathcal{H}(q(\mathbf{u}_k)) - D_{KL}(q(\mathbf{u}_k)||p(\mathbf{u}_k))) \\ &= \sum_{k=1}^{K-1} (-\frac{1}{2}ln[(2\pi e)^N \prod_{i=1}^{N} \lambda_{ki}] - \frac{1}{2}(ln(\frac{det(\mathcal{K}_k)}{det(\Sigma_{q_k})}) + TR(\mathcal{K}_k^{-1}\Sigma_{q_k}) + (-\mu_\mathbf{k})^T\mathcal{K}_k^{-1}(-\mu_\mathbf{k}) - N)) \\ &= \sum_{k=1}^{K-1} (-\frac{1}{2}ln[(2\pi e)^N \prod_{i=1}^{N} \lambda_{ki}] - \frac{1}{2}(ln(\frac{det(\mathcal{K}_k)}{\prod_{i=1}^{N} \lambda_{ki}}) + TR(\mathcal{K}_k^{-1}\Sigma_{q_k}) + (-\mu_\mathbf{k})^T\mathcal{K}_k^{-1}(-\mu_\mathbf{k}) - N)) \\ &= \sum_{k=1}^{K-1} (-\frac{1}{2}ln[(2\pi e)^N] - \frac{1}{2}[ln(det(\mathcal{K}_k)) + TR(\mathcal{K}_k^{-1}\Sigma_{q_k}) + (-\mu_\mathbf{k})^T\mathcal{K}_k^{-1}(-\mu_\mathbf{k}) - N]) \\ &= \sum_{k=1}^{K-1} (-\frac{1}{2}ln[(2\pi e)^N] - \frac{1}{2}[ln(det(\mathcal{K}_k)) + \sum_{i=1}^{N} \mathcal{K}_k^{-1}(i,i) * \lambda_{ki} + (-\mu_\mathbf{k})^T\mathcal{K}_k^{-1}(-\mu_\mathbf{k}) - N])\end{aligned}$$

Again we analyze the result: The first two terms are constant during optimization, we start with term 3: The diagonal elements $\mathcal{K}_k^{-1}(i,i)$ are positive like the λ_{ki}. The negative multiplier $(-\frac{1}{2})$ encourages the variances λ_{ki} to be small. The next (and last term of interest) is very similar to the log of a zero-mean multivariate normal distribution of $\mu_k$with covariance matrix $\mathcal{K}_k$. Such a term is maximal, if the vector μ_k is close to the zero vector, which is what we desire above.

For $\mathcal{L}_{13}$ we have:

$$\begin{aligned}
\mathcal{L}_{13} &= E_{q(\bar{\nu})}\{ln(p(\bar{\nu}|\alpha))\} \\
&= E_{q(\bar{\nu})}\{\sum_{k=1}^{K-1} ln(p(\bar{\nu}_k|\alpha))\} \\
&= \sum_{k=1}^{K-1} E_{q(\bar{\nu}_k)}\{ln(p(\bar{\nu}_k|\alpha))\} \\
&= \sum_{k=1}^{K-1} E_{q(\bar{\nu}_k)}\{ln(Beta(\phi(\bar{\nu}_k)|1-\alpha_a, \alpha_b + k\alpha_a)) + ln(N(\bar{\nu}_k|0,1))\} \\
&= \sum_{k=1}^{K-1} \{ln(\frac{\Gamma(1-\alpha_a+\alpha_b+k\alpha_a)}{\Gamma(1-\alpha_a)\Gamma(\alpha_b+k\alpha_a)}) - \alpha_a E_{q(\bar{\nu}_k)}\{ln(\phi(\bar{\nu}_k))\} + (\alpha_b + k\alpha_a - 1)E_{q(\bar{\nu}_k)}\{ln(1-\phi(\bar{\nu}_k))\} \\
&\quad + \int q(\bar{\nu}_k) ln(N(\bar{\nu}_k|0,1)) d\bar{\nu}_k\} \\
&\approx \sum_{k=1}^{K-1} \{ln(\frac{\Gamma(1-\alpha_a+\alpha_b+k\alpha_a)}{\Gamma(1-\alpha_a)\Gamma(\alpha_b+k\alpha_a)}) - \alpha_a ln(E_{q(\bar{\nu}_k)}\{\phi(\bar{\nu}_k)\}) + (\alpha_b + k\alpha_a - 1)\{ln(E_{q(\bar{\nu}_k)}\{\phi(-\bar{\nu}_k)\}) \\
&\quad + \int q(\bar{\nu}_k) ln(N(\bar{\nu}_k|0,1)) d\bar{\nu}_k\} \\
&\approx \sum_{k=1}^{K-1} \{ln(\frac{\Gamma(1-\alpha_a+\alpha_b+k\alpha_a)}{\Gamma(1-\alpha_a)\Gamma(\alpha_b+k\alpha_a)}) - \alpha_a ln(\phi(\frac{v_k}{\sqrt{1+\delta_k}})) + (\alpha_b + k\alpha_a - 1)\{ln(\phi(-\frac{v_k}{\sqrt{1+\delta_k}})\}) \\
&\quad + \int q(\bar{\nu}_k) ln(N(\bar{\nu}_k|0,1)) d\bar{\nu}_k\}
\end{aligned}$$

We made use of Jensen's Inequality to approximate the intractable term $E_{q(\bar{\nu}_k)}\{ln(\phi(\bar{\nu}_k))\}$, as proposed in the paper. The last term is again a negative cross entropy, this time between univariate gaussians $q(\bar{\nu}_k)$ and $N(\bar{\nu}_k|0,1)$:

$$\begin{aligned}
\int q(\bar{\nu}_k) ln(N(\bar{\nu}_k|0,1)) d\bar{\nu}_k &= (-\mathcal{H}(q(\bar{\nu}_k)) - D_{KL}(q(\bar{\nu}_k)||N(\bar{\nu}_k|0,1))) \\
&= -\frac{1}{2}ln(2\pi e\delta_k) - (\frac{v_k^2}{2} + \frac{1}{2}(\delta_k - 1 - ln\delta_k)) \\
&= \frac{1}{2}ln(2\pi e) - \frac{v_k^2}{2} - \frac{\delta_k}{2} + \frac{1}{2}
\end{aligned}$$

Putting $\mathcal{L}_{13}$ together we get:

$$\begin{aligned}
\mathcal{L}_{13} \approx \sum_{k=1}^{K-1} \{&ln(\frac{\Gamma(1-\alpha_a+\alpha_b+k\alpha_a)}{\Gamma(1-\alpha_a)\Gamma(\alpha_b+k\alpha_a)}) - \alpha_a ln(\phi(\frac{v_k}{\sqrt{1+\delta_k}})) + (\alpha_b + k\alpha_a - 1)ln(\phi(-\frac{v_k}{\sqrt{1+\delta_k}})) \\
&\frac{1}{2}ln(2\pi e) - \frac{v_k^2}{2} - \frac{\delta_k}{2} + \frac{1}{2}\}
\end{aligned}$$

$\mathcal{L}_{14}$ is constant in our optimization and is no regarded.

11 References

- Erik B. Sudderth and Michael I. Jordan. Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes (Proceedings of NIPS 2008)
- Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation (IEEE Conf. on Comp. Vision and Pattern Recognition, San Juan, Puerto Rico, 1997)
- David R. Martin, Charless C. Fowlkes and Jitendra Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues (IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No.5, 2004)
- Ulrike von Luxburg. A Tutorial on Spectral Clustering (Technical Report No. TR-149, 2006)
- Charless Fowlkes, David Martin, Jitendra Malik. Learning Affinity Functions for Image Segmentation: Combining Patch-based and Gradient-based Approaches(2003)
- Jitendra Malik, Serge Belongie, Thomas Leung, Jinabo Shi. Contour and Texture Analysis for Image Segmentation(2001)
- Yossi Rubner, Jan Puzicha, Carlo Tomasi, Joachim M. Buhmann. Empirical Evaluation of Dissimilarity Measures for Color and Texture(2001)
- Khalid El-Arini. Dirichlet Processes - A gentle tutorial (SELECT Lab, 2008)
- Michael I. Jordan. Dirichlet Processes, Chinese Restaurant Processes and All That (University of California, Berkeley)
- Daniel J. Navarro, Thomas L. Griffiths et al. Modeling Individual Differences Using Dirichlet Processes (Journal of Mathematical Psychology 2005)
- Erik B. Sudderth. Graphical Models for Visual Object Recognition and Tracking (Phd thesis 2006)
- Erik B. Sudderth and Michael I. Jordan. Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes- NIPS 2008 Presentation (NIPS 2008)
- Philipp Batz. Variational Inference(May 2010)
- Song-Chun Zhu, Cheng-en Guo, Yizhou Wang and Zijian Xu. What are Textons? (International Journal of Computer Vision)
- Li Fei-Fei and Pietro Perona. A Bayesian Hierarchichal Model for Learning Natural Scene Categories (2005)
- Jonathan Richard Shewchuk. An Introduction to the Conjugate Gradient Method without hte Agonizing Pain(1994)
- Mark W. Woolrich et al. Variational Bayes Inference of Spatial Mixture Models for Segmentation (IEEE Transactions on Medical Imaging, Vol. 25, No. 10, 2006)
- Peter Orbanz and Joachim M. Buhmann. Smooth Image Segmentation by Nonparametric Bayesian Inference (2006)
- Jason A. Duan. Generalized Spatial Dirichlet Process Models (Duke University, Durham)