VARIANTS OF NONNEGATIVE MATRIX FACTORIZATION

P. Ebert C. (s153758), O. Majidi (s163502), M. Hatting P. (s144234), N. Refsgaard (s154317)

Technical University of Denmark

ABSTRACT

In this paper we have investigated two different Non-Negative Matrix Factorization methods to recover the underlying spectrum from a Raman spectroscopy. During our experiments we discovered that the Non-Negative Matrix Factorization with a Gaussian process prior did a greater job in recovering the underlying spectrum than the other methods we investigated. We found out that all our methods is really sensitive to initial hyperparameter choices and is thereby not so stable to minor changes in those. Even though the Non-Negative Matrix Factorization with a Gaussian process prior we also discovered that the Variational-Bayes Non-Negative Matrix Factorization also did a great job in recovering the spectrum but it was a bit more noisy.

Index Terms— NMF, VB, GPP, Raman spectroscopy

1. INTRODUCTION

The problem we are going to solve is to learn the true underlying variables that constitute the signals in raman spectroscopy data.

This raman spectroscopy data is first given as a raman map, that is a surface with small dots indicating the presence of molecules, but we are interested in a spectrum of the molecules that we can extract from this map.
The spectrum we extract, which is presented with a lot of peaks coupled with their location can be used to determine what types of molecules are present of the surface, if any.

The raman map is not presented as small hot-spots on a measured surface, but instead as stripes. This is because we stack the data as vectors which is why the data look a little weird. The reason why we are not modelling the spectrum directly is because we cannot measure it directly. It is only through modelling the raman map at first and then decompose it into a spectra and loadings matrix that we get what we want. Thus our problem is an unsupervised problem, where the NMF models come in.

One might ask what other models could be used for an unsupervised problem, and here PCA comes to mind. When deciding between using NMF models, which can be constructed several ways, and PCA one should note that in the case of very low signal to noise ratios, methods like PCA fail to recover meaningful and interpret-able spectral components [1].

The spectres one might find can sometimes have negative components. Thus the NMF models have a clear advantage in successfully extracting the target spectrum as these spectres are always non-negative [2].

The non-negative matrix factorisation (NMF) model can be stated as $\mathbf{X} = \mathbf{DH} + \mathbf{E}$, where X is the data matrix that contain the entire Raman map. X is factorized into two matrices, the spectra matrix D and the loadings, that is a matrix that contain their amplitudes for a given observation spectrum, $\mathbf{H}$. Both matrices contains only non negative real elements.

In this case we are interested in the matrix $\mathbf{D}$, which will to contain a spectra that are easily interpret-able. In this project we will investigate two methods to uncover the true underlying spectrum. We will try NFM with a Gaussian process prior and a Variational Bayes NMF.

2. MATERIALS AND METHODS

2.1. Data

The data used in the experiments comes from a simulation script, which will generate a noisy Raman map, as well as the underlying Raman map, as well as a spectra and loadings matrix. The underlying spectrum will be based on voigt shaped profiles, which are commonly used in spectroscopy, which looks a lot like Gaussian distributions.

2.2. Models

The NMF model is learned using two methods; a method that includes prior knowledge in a nonnegative matrix factorization based on Gaussian process priors, called GPP (inspired by [3]) and a Bayesian inference schema, here called VB (inspired by [4]).

2.2.1. GPP

The GPP model is based on the assumption that the underlying variable vectors $\mathbf{d}, \mathbf{h}$ each follows a Gaussian process specified by covariance matrices $\mathbf{\Sigma}_d$ and $\mathbf{\Sigma}_h$ respectively.

To get from the Gaussian process into a desired distribution over the nonnegative reals we use a link function. The only requirements for the link functions is that its a strictly increasing function that maps the reals into the nonnegative reals. A proper choice of such link function is to choose a function f_h such that f_h^{-1} maps the marginal distribution of $\mathbf{h}$ into a chosen marginal distribution of $\mathbf{H}$ the same goes for $\mathbf{d}$ of course.

We want to find a suitable loss that minimizes the reconstruction error and enables us to use priors on $\mathbf{D}$ and $\mathbf{H}$ such that we use the fact that the underlying spectrum is a smooth function. The least squares loss can be used as it minimizes the reconstruction error:

$$\mathcal{L}^{\text{LS}}_{X|D,H}(\mathbf{D}, \mathbf{H}) \propto \frac{1}{2\sigma_N^2}||\mathbf{X} - \mathbf{DH}||_F^2$$

By using this loss we then assume that our noise is iid. Gaussian. To include the priors we use a Maximum a Posteriori (MAP) estimate:

$$\mathcal{L}_{D,H|X}(\mathbf{D}, \mathbf{H}) \propto \mathcal{L}^{\text{LS}}_{X|D,H}(\mathbf{D}, \mathbf{H}) + \mathcal{L}_{D,H}(\mathbf{D}, \mathbf{H})$$

Where we assume that $\mathcal{L}_{D,H}(\mathbf{D}, \mathbf{H}) = \mathcal{L}_D(\mathbf{D}) + \mathcal{L}_H(\mathbf{H})$. So we can set a prior on both $\mathbf{D}$ and $\mathbf{H}$.

We have chosen two different link functions to map from the Gaussian process into a chosen distribution. The exponential-to-Gaussian $f_h(\mathbf{h}_i)$ where the inverse is given by:

$$f_h^{-1}(\mathbf{h}_i) = -\frac{1}{\lambda}\log\left(\frac{1}{2} - \frac{1}{2}\Phi\left(\frac{\mathbf{h}_i}{\sqrt{2}\sigma_i}\right)\right)$$

and the rectified-Gaussian-to-Gaussian $f_h(\mathbf{h}_i)$ where the inverse is given by:

$$f_h^{-1}(\mathbf{h}_i) = \sqrt{2}s\Phi^{-1}\left(\frac{1}{2} - \frac{1}{2}\Phi\left(\frac{\mathbf{h}_i}{\sqrt{2}\sigma_i}\right)\right)$$

In this way we have a "link" between $\mathbf{h}$ and $\mathbf{H}$ and also for $\mathbf{d}$ and $\mathbf{D}$ which is given as:

$$\mathbf{H} = \text{vec}^{-1}(f_h^{-1}(\mathbf{h}))$$

Then we just need to choose a covariance function to the Gaussian process for both $\mathbf{D}$ and $\mathbf{H}$. The covariance function describes the correlations in the underlying factors $\mathbf{d}$ and $\mathbf{h}$. In our experiments we have tried both the spatial radial basis function and Laplacian kernel. The RBF kernel is defined as:

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

And the Laplacian is defined as:

$$k(x, y) = \exp\left(-\frac{||x - y||}{\sigma}\right)$$

We tried using both kernels to see what gives the better result and we also tried different combinations of the link functions. In our implementation we have used some change of variables for computational convenience but these are not shown here as the overall algorithm is the same (for further details we refer to the appendix).

2.2.2. VB

In our Bayesian treatment we want to calculate the posterior distribution $p(\mathbf{D}, \mathbf{H}|\mathbf{X}, \Theta)$ of the matrix D and H given data X and hyper parameters $\Theta = \Theta^D, \Theta^H = (A^D, B^D), (A^H, B^H)$ and select the most likely estimate of D and H.

This is done by using the optimizing the lower bound $\mathcal{B}_{\text{VB}}[q]$ of the marginal log likelihood $\mathcal{L}_X(\Theta)$ by minimizing the kl divergence with an instrumental distribution q to get a lower bound that is close to the likelihood.

$$\mathcal{L}_{\mathbf{X}}(\Theta) = \ln(p(\mathbf{X}|\Theta)) \geq$$
$$\langle \ln(p(\mathbf{X}, \mathbf{S}, \mathbf{D}, \mathbf{H}))\rangle_q + H[q] = \mathcal{B}_{\text{VB}}[q]$$

Here we assume that the distribution q can be factorized into 3 terms, $q = q(S, D, H) = q(S)q(D)q(H)$, as it is otherwise a complex function to work with.

With this factorization of q it turns out that we can use the following fixed point iteration equation to get a local optimum of q;

$$q_\alpha^{n+1} \propto \exp(\langle \ln[p(\mathbf{X}, \mathbf{S}, \mathbf{D}, \mathbf{H}|\Theta)]\rangle_{q^{\neg\alpha^{(n)}}}) \qquad (1)$$

This iteration monotonically improves the individual factors of the distribution q, which helps the algorithm converge. The α implies that we first optimize q with respect to $\mathbf{S}$, $\mathbf{D}$ or $\mathbf{H}$ and also the order of which variable we choose to optimize doesn't matter.

In order to calculate an iteration of q we need to expand the expression $p(\mathbf{X}, \mathbf{S}, \mathbf{D}, \mathbf{H}|\Theta)$. This can be rewritten by using the product rules of probability;

$$p(\mathbf{X}, \mathbf{S}, \mathbf{D}, \mathbf{H}|\Theta) = p(X|S)p(S|D, H)p(D|\Theta^D)p(H|\Theta^H)$$

We will also need to make some assumptions about the prior distributions on the matrix D and H, which in this case means that the elements of matrix D and H are gamma distributed and that our data X is govern by some latent sources S, in form of a sum of latent variables, which follows a Poisson

distribution;

$$d_{k,m} \sim \mathcal{G}\left(A^D, \frac{B^D}{A^D}\right), h_{m,l} \sim \mathcal{G}\left(A^H, \frac{B^H}{A^H}\right)$$
$$s_{k,m,l} \sim \mathcal{PO}(s_{k,m,l}, d_{k,m}h_{m,l}) \quad x_{k,l} = \sum_m s_{k,m,l}$$

In order to get new estimates for the elements in **D** and **H** we use the above equations for d and h as initial values for an iterative update scheme;

$$\langle d^0_{k,m} \rangle \sim \mathcal{G}(\cdot, \Theta^D) \qquad \langle h^0_{m,l} \rangle \sim \mathcal{G}(\cdot, \Theta^H)$$

Then we update them by finding the expectation of d and h as $q(D)$ and $q(H)$ follow as gamma distribution ($\mathcal{G}(d_{k,m}, \alpha_d, \beta_d)$) and ($\mathcal{G}(h_{m,l}, \alpha_h, \beta_h)$) and their mean values are given by $d_{k,m}$ and $h_{m,l}$ and the α and β parameters can be expressed in terms of the parameters for the priors on D and H as well as their elements from the previous iteration;

$$\langle d^n_{k,m} \rangle = \alpha^n_d \beta^n_d \qquad \langle h^n_{m,l} \rangle = \alpha^n_h, \beta^n_h$$

The term $q(S)$ follows a multinomial distribution and a more specific expression of the above update equation can be found in the appendix.

3. RESULTS

Applying the two methods on the following noisy data, will yield a reconstruction of the data on the left side of the figure and a spectrum on the right side, which should be compared to the following spectrum on the right;

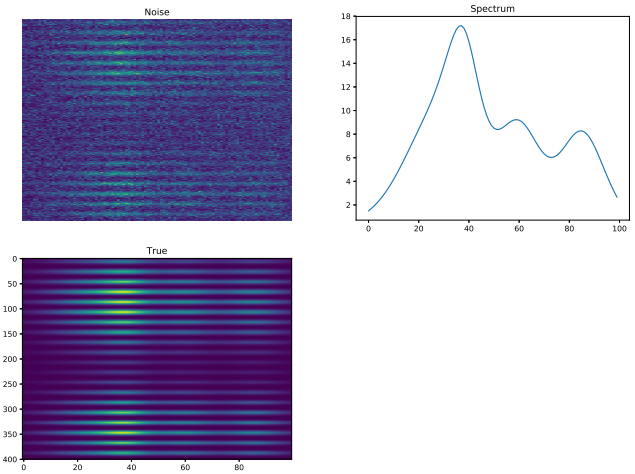

Fig. 1. The noisy data matrix (left) and the true ground spectrum (right). The true underlying data matrix is displayed below the noisy data matrix.

By the given choices of covariance functions (in this case the laplace kernel) and link functions (exponential to gaussian for D and rectified gaussian for H) we observed that the GPP (Fig. 2.) was better to reconstruct the spectrum than traditional NMF (Fig.4.) and VB (Fig. 3.), when using the correct covariance function.

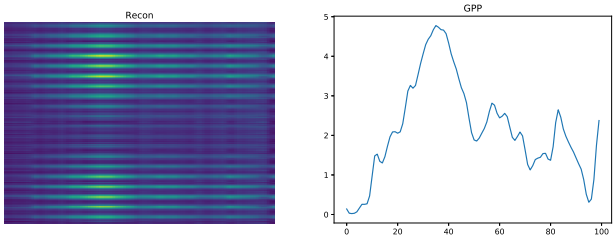

Fig. 2. GPP reconstruction of **X** (left) and reconstruction of spectrum (right) with Laplacian covariance function.

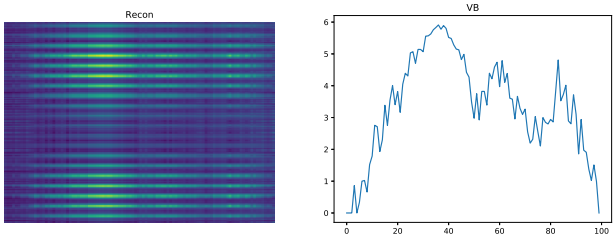

Fig. 3. VB reconstruction of **X** (left) and reconstruction of spectrum (right).

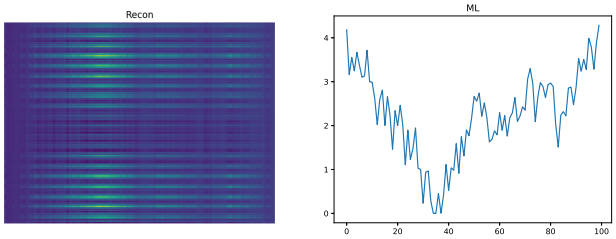

Fig. 4. Reconstruction of **X** (left) and the spectrum (right) by means of Maximum Likelihood (traditional NMF).

However when choosing a wrong covariance (Fig.5.) function (such as the gaussian kernel) as well as wrong initial parameters (Fig.6.) for this function (a width parameter for the rbf that is too large) the VB (Fig.3.) scheme performed better in general.

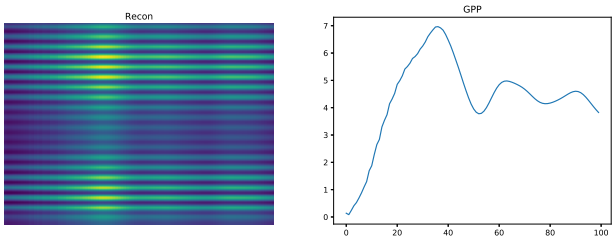

Fig. 5. GPP reconstruction of **X** (left) and reconstruction of spectrum (right).

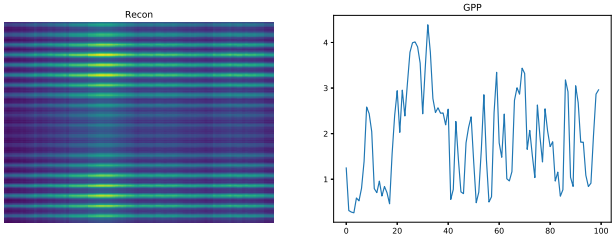

Fig. 6. GPP reconstruction of **X** (left) and reconstruction of spectrum (right) with altered parameters.

Experiments on the generated data showed that both methods where highly sensitive to changes in initial values as well as parameter settings, when reconstructing the spectrum.

Both methods where very good at reconstructing the data X however, as well as much less sensitive to changes in parameters and initial values.

In general the VB method where a lot faster than the GPP to find a solution, while it also didn't require specification of any covariance matrices.

4. DISCUSSION

In our experiments we have investigated the two methods mentioned earlier namely GPP-NMF and VB-NMF. We have tried with different settings for the two methods and discovered that the methods are not very robust to changes in initial settings and therefore for future work one might look into hyperparameter settings using random-search or grid-search.

The priors we have chosen to work with can seem kinda arbitrary but they are chosen very carefully such that for the VB-scheme the priors are conjugate such that we can compute the instrumental distribution q. So therefore we use the gamma distribution and a Poisson distribution. Another reason for using the gamma distribution is due to this is often used to model physical systems that only take non-negative values. The Poisson distribution is mainly used when we are counting some events that happen infrequently, independent of each-other and when they occur with a known rate.

We are using a general method for choosing a prior to improve the quality of the NMF and the reason why we did not choose another prior is because by having priors that are not conjugate, the expectation of the distribution becomes intractable.

5. CONCLUSION

We have used two methods for extracting the spectrum from a raman map, both using prior knowledge based on either Poisson & gamma priors or Gaussian process priors linked to the factors D and H by a link function. The experiments on the generated data showed that the GPP method can give better results in terms of estimating the true underlying spectrum, when comparing with traditional NMF and VB.

6. ACKNOWLEDGEMENTS

We would like to thank Tommy S. Alstrøm and Mikkel N. Schmidt for supervision, providing us a data generator and always answering our questions both during meetings and at other times.

7. REFERENCES

[1] Tommy S. Alstrøm, Kasper B. Frohling, Jan Larsen, Mikkel N. Schmidt, Michael Bache, Michael S. Schmidt, Mogens H. Jakobsen, and Anja Boisen, "Improving the robustness of Surface Enhanced Raman Spectroscopy based sensors by Bayesian Non-negative Matrix Factorization," *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2014.

[2] Daniel D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788, 1999.

[3] Mikkel N. Schmidt and Hans Laurberg, "Nonnegative matrix factorization with Gaussian process priors," *Computational Intelligence and Neuroscience*, 2008.

[4] Ali T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Computational Intelligence and Neuroscience*, 2009.