Using probabilistic methods for hierarchical visualization of single-cell RNA-seq data

Tobias Beers

Universiteit van Amsterdam

July 22nd, 2020

Layout

1. Introduction
    - scRNA-seq data and dimensionality reduction
    - Theory behind probabilistic hierarchical visualization
    - Probabilistic programming: Stan
2. Methods
    - Data
    - Experiment set-up
3. Results
4. Conclusions

Gene expression and scRNA-seq

- Gene expression: DNA → mRNA → gene product
- High expression of a gene means more mRNA
- Single-cell RNA sequencing (scRNA-seq) measures relative gene expression by quantifying mRNA transcripts
- ScRNA-seq data may contain thousands of dimensions (genes), but visualization is easier in two dimensions

Dimensionality reduction

- linear techniques
  - PCA
  - PPCA
- non-linear techniques
  - t-SNE
  - UMAP
  - Hierarchical Mixture of PPCAs (HmPPCAs)

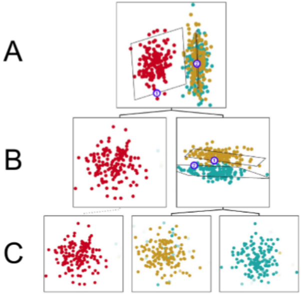


Figure has been copied from Bishop & Tipping (1998) [1]

Innovation and relevance

- HmPPCAs has been used on scRNA-seq data before [2]
- So far, HmPPCAs tree has been built interactively
  - Automatic Clustering
- HmPPCAs is solved through expectation-maximization (EM) algorithm
  - Probabilistic programming

PPCA

- Full data-set $\boldsymbol{X}$, latent data-set $\boldsymbol{Z}$
- Latent data (m dimensions) is transformed into full data-set (d dimensions): $\boldsymbol{x}_i = \boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu} + \epsilon$
  - $\boldsymbol{W}$: factor loadings, $\boldsymbol{\mu}$: added means, ϵ: noise
- Therefore, $\boldsymbol{x}|\boldsymbol{z}$ follows the distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$

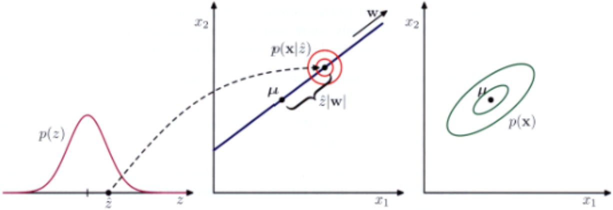


Figure has been copied from Bishop (2006) [3]. $m = 1, d = 2$

Mixture of PPCAs (MoPPCAs)

- Suppose our data-set is the result of K latent variable models
- Now, $p(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}, \boldsymbol{W}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}_k \boldsymbol{z}_k + \boldsymbol{\mu}_k, \sigma_k^2 \boldsymbol{I})$
    - Where the mixture coefficient π_k denotes which proportion of the data comes from mixture component k

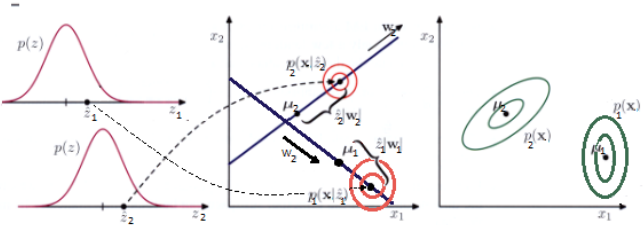


Figure has been copied and modified from Bishop (2006) [3].
$m = 1, d = 2, K = 2$

Hierarchical Mixture of PPCAs (HmPPCAs)

- We can add more levels: suppose mixture component k consists of multiple sub-components m
- As many levels can be added as necessary!

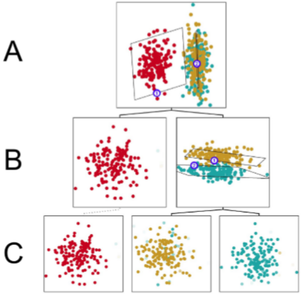
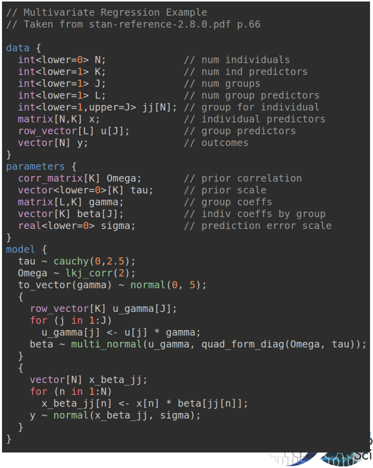


Figure has been copied from Bishop & Tipping (1998) [1]

Stan

- Probabilistic Programming Language
- Specify a model, input data and Stan finds posterior distribution of parameters given observed data
- Easy add changes to model
- Two methods of inference:
  - NUTS
  - ADVI/VB

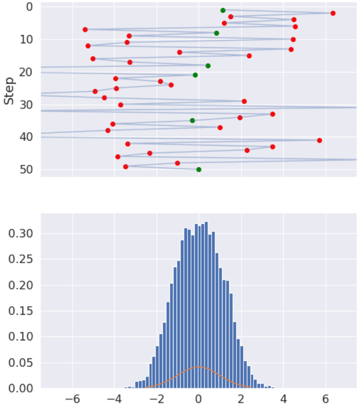


```
// Multivariate Regression Example
// Taken from stan-reference-2.8.0.pdf p.66

data {
  int<lower=0> N;              // num individuals
  int<lower=1> K;              // num ind predictors
  int<lower=1> J;              // num groups
  int<lower=1> L;              // num group predictors
  int<lower=1,upper=J> jj[N];  // group for individual
  matrix[N,K] x;               // individual predictors
  row_vector[L] u[J];          // group predictors
  vector[N] y;                 // outcomes
}
parameters {
  corr_matrix[K] Omega;        // prior correlation
  vector<lower=0>[K] tau;      // prior scale
  matrix[L,K] gamma;           // group coeffs
  vector[K] beta[J];           // indiv coeffs by group
  real<lower=0> sigma;         // prediction error scale
}
model {
  tau ~ cauchy(0,2.5);
  Omega ~ lkj_corr(2);
  to_vector(gamma) ~ normal(0, 5);
  {
    row_vector[K] u_gamma[J];
    for (j in 1:J)
      u_gamma[j] <- u[j] * gamma;
    beta ~ multi_normal(u_gamma, quad_form_diag(Omega, tau));
  }
  {
    vector[N] x_beta_jj;
    for (n in 1:N)
      x_beta_jj[n] <- x[n] * beta[jj[n]];
    y ~ normal(x_beta_jj, sigma);
  }
}
```

NUTS

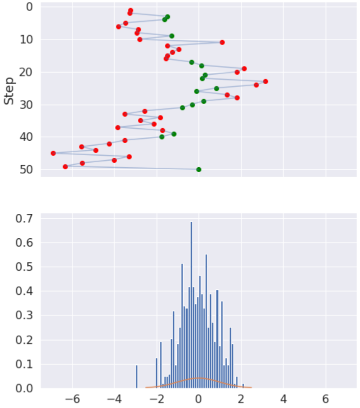
- **Metropolis-Hastings**
  - Convergence takes long due to inefficient pathways
- Hamiltonian Monte Carlo
- NUTS



ıtional
ce

NUTS

- Metropolis-Hastings
- **Hamiltonian Monte Carlo**
  - Faster convergence
  - Need to pick values for path-length L and step-size ϵ
- NUTS



ıtional
ce

NUTS

- Metropolis-Hastings
- Hamiltonian Monte Carlo
- **NUTS**
  - Automatically tunes path-length L and step-size ϵ

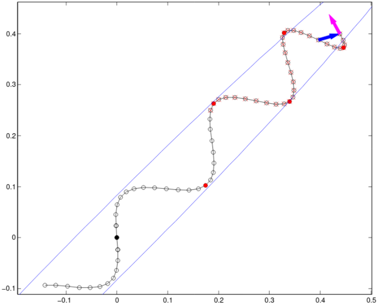
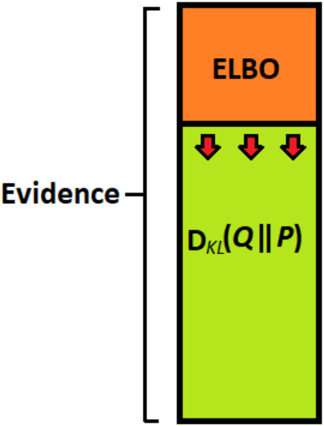


Figure has been copied from Hoffman & Gelman (2014) [4]

ADVI

- Variational inference
  - Approach $P(\theta|\boldsymbol{X})$ by initializing $Q(\zeta)$ and minimize $D_{KL}(Q||P)$
  - $D_{KL}(Q||P) = \text{evidence} - \text{ELBO}$
  - Evidence is independent of Q
- ADVI automatizes this process

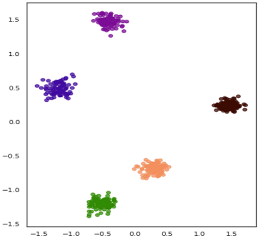


Data

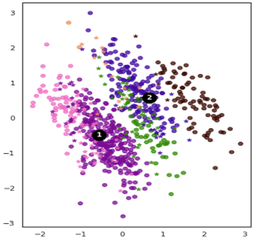
- Simulated
  - 10 Splatter data-sets varying in complexity and number of genes
  - 5 - 250 genes
- Experimental
  - Darmanis *et al.* [5]
  - Nestorowa *et al.* [6]
  - Both were filtered to include only 500 genes with largest variance

Experiment set-up

- HmPPCAs was compared with PPCA, t-SNE and UMAP
- Visualization performance was scored on cell type separability
  - Multinomial logistic regression on latent data-sets using 5-fold cross-validation

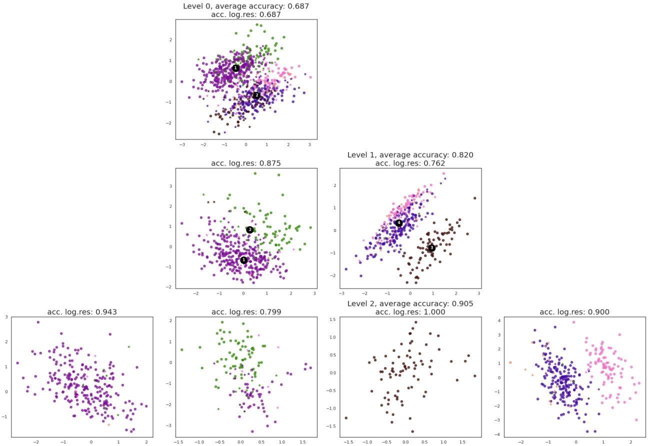


(a) Well separated, easy to predict cell type of new data-points



(b) Badly separated, difficult to predict cell type of new data-points

Example of HmPPCAs on a Splatter data-set



Level 0, average accuracy: 0.687
acc. log.res: 0.687

acc. log.res: 0.875

Level 1, average accuracy: 0.820
acc. log.res: 0.762

acc. log.res: 0.943

acc. log.res: 0.799

Level 2, average accuracy: 0.905
acc. log.res: 1.000

acc. log.res: 0.900

Results

- HmPPCAs almost always scored higher than PPCA
- t-SNE and UMAP consistently outperformed HmPPCAs

Conclusions

- HmPPCAs not as accurate as UMAP or t-SNE
- Adding hierarchy did improve on a standard PPCA
- HmPPCAs outperfomed UMAP and t-SNE in earlier literature
    - Errors in automatic clustering
    - Incorrect initialization MoPPCAs
- Easy to incorporate more elements in the model due to probabilistic programming (e.g. zero-inflation)

Thank you!

References:

[1] Christopher M Bishop and Michael E Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.

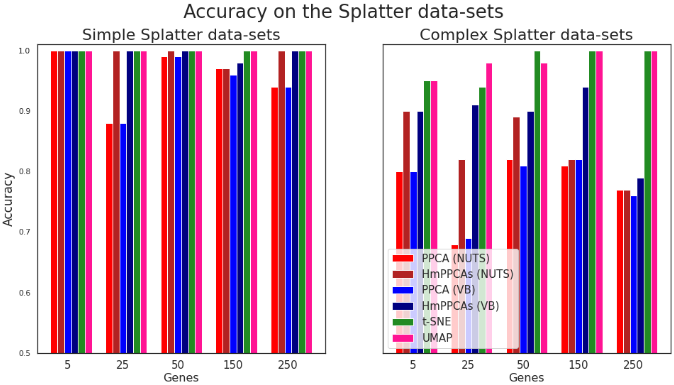
[2] Philip van Kuiken. Hierarchical visualization of single cell RNA-seq data. Master's thesis, Vrije Universiteit Amsterdam, the Netherlands, 2017.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

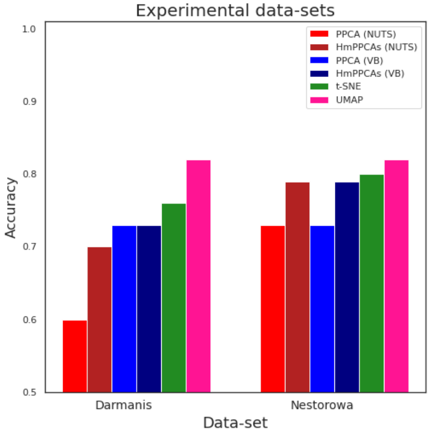
[4] Matthew D Hoffman and Andrew Gelman. The no-U-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[5] Spyros Darmanis, Steven A Sloan, Ye Zhang, Martin Enge, Christine Caneda, Lawrence M Shuer, Melanie G Hayden Gephart, Ben A Barres, and Stephen R Quake. A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences*, 112(23):7285–7290, 2015.

[6] Sonia Nestorowa, Fiona K Hamey, Blanca Pijuan Sala, Evangelia Diamanti, Mairi Shepherd, Elisa Laurenti, Nicola K Wilson, David G Kent, and Berthold Göttgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood, The Journal of the American Society of Hematology*, 128(8):e20–e31, 2016.


Accuracy on the Splatter data-sets
Simple Splatter data-sets
Complex Splatter data-sets
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
5
25
50
150
250
Genes
PPCA (NUTS)
HmPPCAs (NUTS)
PPCA (VB)
HmPPCAs (VB)
t-SNE
UMAP

Results - accuracy



Results - accuracy

Table: **Accuracy of multinomial logistic regressions on the latent data-sets found by each model in a 5-fold cross-validation scheme**

	Splatter simple					Splatter complex					Darmanis	Nestorowa
genes	5	25	50	150	250	5	25	50	150	250	500	500
PPCA (NUTS)	**1.00**	0.88	0.99	0.97	0.94	0.80	0.68	0.82	0.81	0.77	0.60	0.73
HmPPCAs (NUTS)	**1.00**	**1.00**	**1.00**	0.97	**1.00**	0.90	0.82	0.89	0.82	0.69	0.70	0.79
PPCA (VB)	**1.00**	0.88	0.99	0.96	0.94	0.80	0.69	0.81	0.82	0.76	0.73	0.73
HmPPCAs (VB)	**1.00**	**1.00**	**1.00**	0.98	**1.00**	0.90	0.91	0.90	0.94	0.79	0.73	0.79
UMAP	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	**0.98**	0.98	**1.00**	**1.00**	**0.82**	**0.82**
t-SNE	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.95**	0.94	**1.00**	**1.00**	**1.00**	0.76	0.80

Automatic clustering

- Fit GMM models on latent data, while varying the number of clusters
- Compute BIC: $BIC = k \ln n - 2 \ln \mathcal{L}$,
  - n: number of data-points, k: number of clusters, $\mathcal{L}$: likelihood of model
- Choose model with lowest BIC for the intialization of MoPPCAs