Comp790-166: Computational Biology

Lecture 7

February 9, 2021

Announcements

- Homework 1 is online! (due Feb 18)
- Project Proposal due in 1 month (March 9)
- Wellness day next Tuesday Feb 16, no class!

Today

- Geometry-Based Data Generation with SUGAR
- Linking Single Cells to Clinical Outcomes with MELD (start MELD)

Rare + Sparse Cell Populations

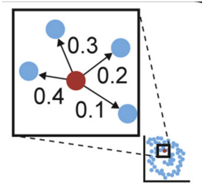


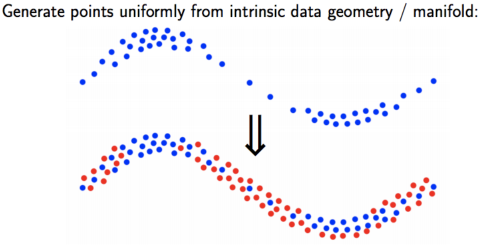
Figure: from MAGIC paper

- **From Last Time:** ‘Dense‘ cell-populations vs ‘Sparse‘ cell-populations.
- Low-frequency cell-populations and data artifacts can fail to be under-represented and not accurately reflect the underlying biology.
- Are members of a particular cell-type not there or are they just very infrequent?

General Problems with Downstream Tasks from Sparse Data

- Imbalanced classes can affect classification accuracy
- In clustering, imbalanced ‘ground-truth‘ clusters can cause distortion or mis-representation of the clusters in the data
- Too few samples/observations can cause mis-representation of the dependencies or correlations between features.

Welcome SUGAR



Generate points uniformly from intrinsic data geometry / manifold:

Figure: from Lindenbaum *et al.* NeurIPS. 2018

- Most generative modeling approaches seek to learn and replicate the data density
- SUGAR is a data-generation approach that uses the underlying geometry of the data (e.g. a random walk based approach)

Common Data Generation Approaches

- Older Techniques
  - **Parametric Models :** Specify a model and optimize the parameters through maximum likelihood optimization.
  - Use the learned model to generate new data
  - Use a histogram or kernel to estimate generating distributions
- Newer techniques to generate additional points from complicated data dsitributions
  - GAN (generative adversarial nets) → `https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`
  - VAE (variational autoencoders) → `https://arxiv.org/abs/1606.05908`

Remembering our Good Friend Gaussian Kernel

For a pair of nodes, i and j, the strength of their connection can be computed as $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, or

$$K_{ij} = \exp(-\frac{||\mathbf{x}_i - x_j||^2}{2\sigma^2}) \tag{1}$$

We also remember our row-stochastic markov matrix, **P** computed from **K**.

Measure Based Gaussian Correlation

- Given your data, $\mathbf{X}$, define a set of reference points, $\mathbf{r}$ with $r \in \mathbf{X}$.
- Define $\mu(\mathbf{r})$ as some measure over the reference points

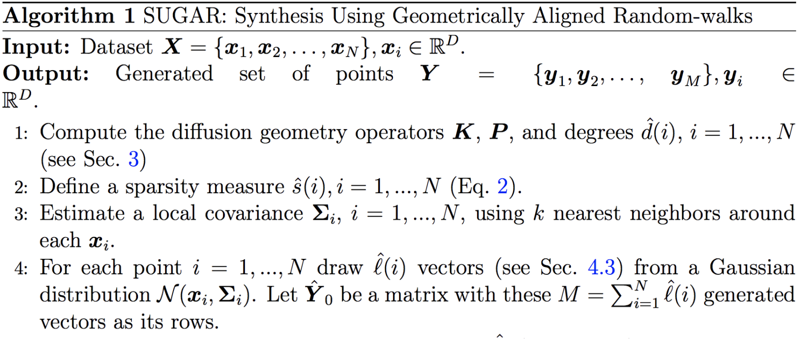
Then define a new kernel, $\hat{\mathcal{K}}\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ as,

$$\hat{\mathcal{K}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\mathbf{r} \in \mathbf{X}} \mathcal{K}(\mathbf{x}_i, \mathbf{r})(\mathbf{x}_j, \mathbf{r})\mu(\mathbf{r}) \tag{2}$$

Problem Formulation for Data Generation Problem

- Let $\mathcal{M}$ be a d-dimensional manifold such that $\mathcal{M} \in \mathbb{R}^d$
- Let $\mathbf{X} \subset \mathcal{M}$ be a subset of points samples from $\mathcal{M}$ with $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\} \in \mathbf{X}$
- Assuming that instances, $\mathbf{X}$ are unevenly sampled from $\mathcal{M}$, we seek a set of new points, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_M\}$ that also lie of $\mathcal{M}$ and that $\mathbf{X} \cup \mathbf{Y}$ is uniform.

Synthesis Using Geometrically Aligned Random Walks



```
Algorithm 1 SUGAR: Synthesis Using Geometrically Aligned Random-walks
Input: Dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}, \boldsymbol{x}_i \in \mathbb{R}^D$.
Output: Generated set of points $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_M\}, \boldsymbol{y}_i \in \mathbb{R}^D$.
1: Compute the diffusion geometry operators $\boldsymbol{K}$, $\boldsymbol{P}$, and degrees $\hat{d}(i)$, $i = 1, ..., N$ (see Sec. 3)
2: Define a sparsity measure $\hat{s}(i), i = 1, ..., N$ (Eq. 2).
3: Estimate a local covariance $\boldsymbol{\Sigma}_i$, $i = 1, ..., N$, using $k$ nearest neighbors around each $\boldsymbol{x}_i$.
4: For each point $i = 1, ..., N$ draw $\hat{\ell}(i)$ vectors (see Sec. 4.3) from a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_i, \boldsymbol{\Sigma}_i)$. Let $\hat{\boldsymbol{Y}}_0$ be a matrix with these $M = \sum_{i=1}^{N} \hat{\ell}(i)$ generated vectors as its rows.
```

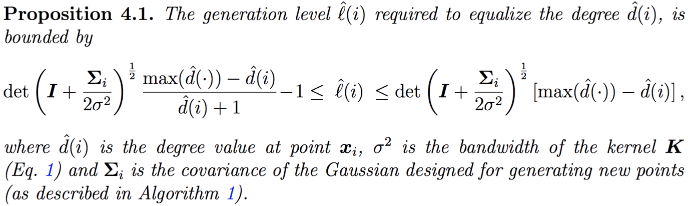
Figure: from Lindenbaum *et al.* NeurIPS. 2018. Let's walk through steps 1-4 for now.

Synthesis Using Geometrically Aligned Random Walks

- **Specify Kernel:** Initialized by forming a Gaussian Kernel over the input data, $\mathbf{X}$, $\mathbf{G_x}$.
    - Use $\mathbf{G_x}$ to estimate the degree of each node i, $d(i)$.
    - The sparsity of each point, $s(i)$ is defined as the inverse degree of node i or $s(i) = 1/d(i)$.
- **Sample According to Each Point** Next, sample $\ell(i)$ points, $\mathbf{h}_j \in \mathbf{H}_i$ around each $\mathbf{x}_i \in \mathbf{X}$ from a localized gaussian distribution.
    - $G_i = \mathcal{N}(\mathbf{x}_i, \Sigma_i)$

Choosing $\ell(i)$

- $\ell(i)$ determines how many points should be sampled around point i to comprise the $\ell_i \times d$ matrix, $\mathbf{H}_i$.
- ‘Too long don’t read‘ but the authors proved bounds for the number of points, $\ell(i)$ to sample for each i to get uniform representation, or similarly degree for each i, across $\mathcal{M}$.



Proposition 4.1. *The generation level $\hat{\ell}(i)$ required to equalize the degree $\hat{d}(i)$, is bounded by*

$$\det\left(\boldsymbol{I}+\frac{\boldsymbol{\Sigma}_i}{2\sigma^2}\right)^{\frac{1}{2}}\frac{\max(\hat{d}(\cdot))-\hat{d}(i)}{\hat{d}(i)+1}-1\le \hat{\ell}(i)\le\det\left(\boldsymbol{I}+\frac{\boldsymbol{\Sigma}_i}{2\sigma^2}\right)^{\frac{1}{2}}\left[\max(\hat{d}(\cdot))-\hat{d}(i)\right],$$

where $\hat{d}(i)$ is the degree value at point $\boldsymbol{x}_i$, σ^2 is the bandwidth of the kernel $\boldsymbol{K}$ (Eq. 1) and $\boldsymbol{\Sigma}_i$ is the covariance of the Gaussian designed for generating new points (as described in Algorithm 1).

Figure: from Lindenbaum *et al.* NeurIPS. 2018.

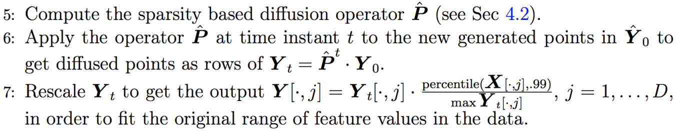
Back to SUGAR

- Let $\mathbf{Y}_0$ be the set of all new $M = \sum_i \ell(i)$ points generated around each i, with $\mathbf{Y}_0 = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$
- **MGC Kernel:** Now use affinities between points in $\mathbf{X}\ \mathbf{Y}_0$. Here, points in $\mathbf{X}$ are used as reference.

$$\hat{\mathcal{K}}(\mathbf{y}_i, \mathbf{y}_j) = \sum_r \mathcal{K}(\mathbf{y}_i, \mathbf{x}_r)\mathcal{K}(\mathbf{x}_r, \mathbf{y}_j)s(\mathbf{r}) \tag{3}$$

Pulling $\mathbf{Y}_0$ towards sparser regions of $\mathcal{M}$

Pasted psuedo code for the second part of SUGAR



```
Compute the sparsity based diffusion operator $\hat{\boldsymbol{P}}$ (see Sec 4.2).
Apply the operator $\hat{\boldsymbol{P}}$ at time instant $t$ to the new generated points in $\hat{\boldsymbol{Y}}_0$ to
    get diffused points as rows of $\boldsymbol{Y}_t = \hat{\boldsymbol{P}}^t \cdot \boldsymbol{Y}_0$.
Rescale $\boldsymbol{Y}_t$ to get the output $\boldsymbol{Y}[\cdot, j] = \boldsymbol{Y}_t[\cdot, j] \cdot \frac{\text{percentile}(\boldsymbol{X}[\cdot,j],.99)}{\max \boldsymbol{Y}_t[\cdot,j]}$, $j = 1, \ldots, D$,
    in order to fit the original range of feature values in the data.
```

Figure: from Lindenbaum *et al.* NeurIPS. 2018.

Diffusion Operator Again

- Take affinities from $\hat{\mathcal{K}}$ and convert them to $\mathbf{P}$, the row-normalized Markov matrix.
- This will allow us to correct points in $\mathbf{Y}_0$ according to neighborhood regions

As you will all recognize, powering $\mathbf{P}$ as $\mathbf{P}^t$ estimates the probability of successfully traveling between nodes with t steps. The transformed matrix, $\mathbf{Y}_t$ is computed as

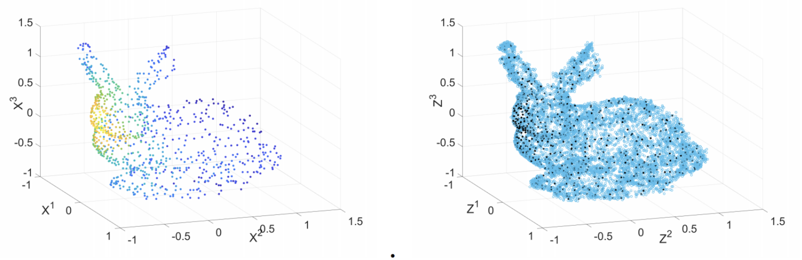
$$\mathbf{Y}_t = \mathbf{P}^t \cdot \mathbf{Y}_o \tag{4}$$

Same Story Regarding t

Remember in PHATE, t was chosen according to the knee point of the Von-Neumann entropy of the normalized eigenvalues of $\mathbf{P}^t$.

Experiments : Synthetic and Biological

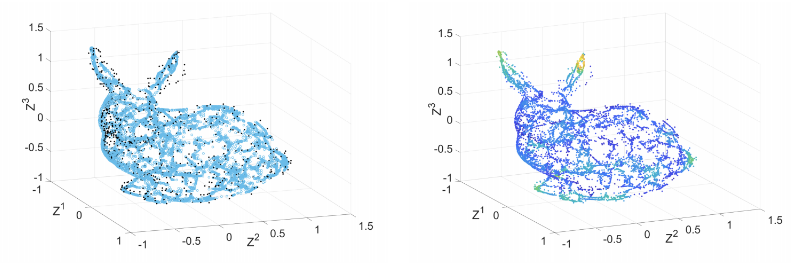
Stanford Bunny



(a) (b)

Figure: from Lindenbaum *et al.* NeurIPS. 2018. (**a**). The original set, **X** of points, colored by node degree. (**b**) $\mathbf{Y}_0$ are generated points (blue) and original points, **X** (black).

Stanford Bunny Part II



(c)

(d)

Figure: from Lindenbaum *et al.* NeurIPS. 2018. (**c**). Original and generated points, before MGC diffusion. (**d**) Generated points (**Y**) after MGC diffusion. Points are colored by degree.

Application : Augmented Clustering

SUGAR was used to generate additional data points to improve the quality of clusters identified with *k*-means.

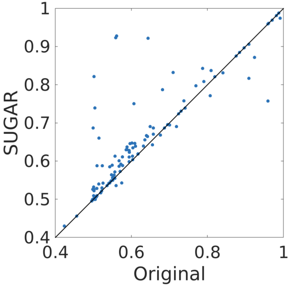


Figure: from Lindenbaum *et al.* NeurIPS. 2018. In 119 datasets, the data were augmented with additional datapoints using sugar. Adjusted Rand Index was computed for the original data vs data + SUGAR.

SUGAR on Single-Cell

SUGAR was using to augment cells in a single-cell RNAseq dataset.

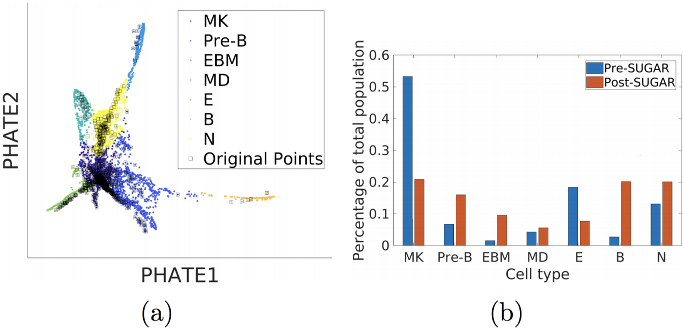
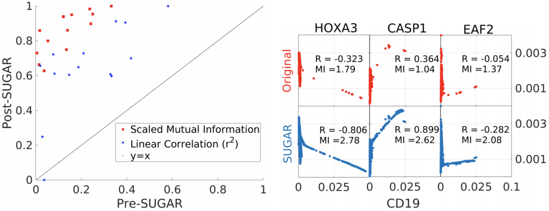


Figure: from Lindenbaum *et al.* NeurIPS. 2018

Maintaining Intra-Module Marker Co-Expression

Among cells assigned to the same module or cluster, after SUGAR lead to higher intra-module between-marker correlation.



(c) (d)

Figure: from Lindenbaum *et al.* NeurIPS. 2018

Linking Single Cell Data to External Information

MELD

The idea of MELD is to model how experimental perturbation alters cell states.

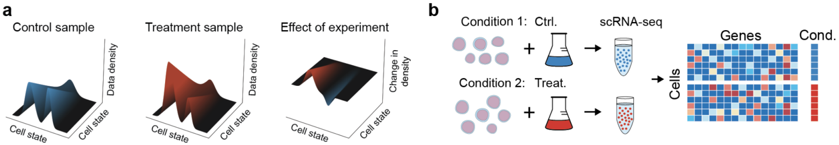
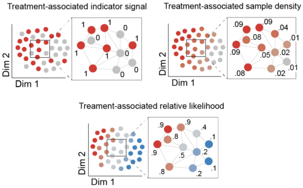


Figure: Burkhardt *et al.*, Nature Biotechnology. 2021. How more or less likely are we to observe a cell in a particular state in a control vs treatment sample?

Treatment as a Signal on a Graph

After creating a graph of cells, an indicator of treatment or control can be viewed as the signal on the graph. Interpretations of ‘signal‘ in relation to graph structure should help to inform treatment associated relative likelihood.



Treatment-associated indicator signal

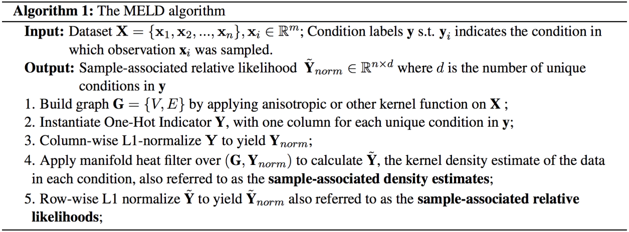
Treatment-associated sample density

Treament-associated relative likelihood

Figure: Burkhardt *et al.*, Nature Biotechnology. 2021.

MELD Overview

Ultimately, the goal is to estimate the density of signals over a graph, where the nodes are cells and edges represent affinity between cells.



```
Algorithm 1: The MELD algorithm
  Input: Dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^m$; Condition labels $\mathbf{y}$ s.t. $\mathbf{y}_i$ indicates the condition in
         which observation $\mathbf{x}_i$ was sampled.
  Output: Sample-associated relative likelihood $\tilde{\mathbf{Y}}_{norm} \in \mathbb{R}^{n \times d}$ where $d$ is the number of unique
          conditions in $\mathbf{y}$
  1. Build graph $\mathbf{G} = \{V, E\}$ by applying anisotropic or other kernel function on $\mathbf{X}$ ;
  2. Instantiate One-Hot Indicator $\mathbf{Y}$, with one column for each unique condition in $\mathbf{y}$;
  3. Column-wise L1-normalize $\mathbf{Y}$ to yield $\mathbf{Y}_{norm}$;
  4. Apply manifold heat filter over $(\mathbf{G}, \mathbf{Y}_{norm})$ to calculate $\tilde{\mathbf{Y}}$, the kernel density estimate of the data
    in each condition, also referred to as the sample-associated density estimates;
  5. Row-wise L1 normalize $\tilde{\mathbf{Y}}$ to yield $\tilde{\mathbf{Y}}_{norm}$ also referred to as the sample-associated relative
    likelihoods;
```

Figure: Burkhardt *et al.*, Nature Biotechnology. 2021.

What on Earth is a Cell-State

We have seen cell states already!

- Frequency → how many of a particular cell-type are there in a sample?
- Function → Which proteins are activated in a particular cell-type?

Defining Kernel Density Estimation (KDE)

- The objective is to estimate the density of each point in some high-dimensional space. Assuming $\mathbf{X}$ represents the data (e.g. cells $\times$ features)

A general kernel density estimator $f(\mathbf{x}, t)$ with bandwidth $t > 0$ and kernel function $K(\mathbf{x}, t)$ can be written as,

$$\hat{f}(\mathbf{x}, t) = \frac{1}{N} \sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i, t), x \in X \tag{5}$$

KDE, continued

A general kernel density estimator $f(x,t)$ with bandwidth $t > 0$ and kernel function $K(x,y,t)$ can be written as,

$$\hat{f}(\mathbf{x},t) = \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{x},\mathbf{x}_i,t), x \in X \tag{6}$$

with K as follows, assuming $\mathbf{X} \in \mathbb{R}^d$

$$K(\mathbf{x},\mathbf{y},t) = \frac{1}{(4\pi t)^{d/2}} \exp^{-||\mathbf{x}-\mathbf{y}||_2^2/4t} \tag{7}$$

A Between-Cell Graph as a Discrete Analog

- Assume that the data lie over a discrete manifold. Distances on a manifold are only locally Euclidean, so we will use a discrete analog.
- Also assume that as the number of samples, $N \to \infty$, the discrete approximation would match the continuous KDE
- From a practical viewpoint, we will derive the Graph Fourier Transform (GFT), which has nice associated theory for studying graph signals.

Graph Laplacian as a Solution to the Diffusion Equation

When we derived the Graph Laplacian, we noted its relationship to the Diffusion Equation (see Lecture 2).

- Properties of the underlying data manifold can be approximated by the eigenvectors of the discrete Laplacian. $\mathcal{L}$.

So,

$$\hat{K}_{\mathcal{L}}(x,t) = \exp^{-t\mathcal{L}} x \tag{8}$$

Here, x is some empirical density estimated from the data.

Graph Fourier Transform

- Extract frequency content of spatio-temporal signals.
- Encodes a notion of smoothness. A signal is smooth if neighboring points have similar values.

Deriving Graph Fourier Transform

Assuming we have a signal on each node, $\mathbf{f} \in \mathbb{R}^N$. Starting the Graph Laplacian, we can compute $\mathcal{L}\mathbf{f}$ or the local variation of a signal around node i as,

$$\begin{aligned}
(\mathcal{L}\mathbf{f})(i) &= ([\mathbf{D} - \mathbf{A}]\mathbf{f})(i) &\quad (9)\\
&= d(i)\mathbf{f}(i) - \sum_j A_{ij}\mathbf{f}(j) &\quad (10)\\
&= \sum_j A_{ij}(\mathbf{f}(i) - \mathbf{f}(j)) &\quad (11)
\end{aligned}$$

Total Variation of a Signal

The total variation of a signal on a graph is defined as follows and is also known as the Laplacian Quadratic Form

$$
\begin{aligned}
TV(\mathbf{f}) &= \sum_{i,j} A_{ij}(\mathbf{f}(i) - \mathbf{f}(j))^2 \qquad (12)\\
&= \mathbf{f}^T \mathcal{L} \mathbf{f} \qquad (13)
\end{aligned}
$$

Getting to Graph Fourier Basis

- We can look at eigenvectors, $\mathbf{\Psi} = [\psi_1, \psi_2, \ldots, \psi_N]$
- and eigenvalues, $\mathbf{\Lambda} = [0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}]$ of $\mathcal{L}$

The Graph Fourier Transform of a Signal

The graph fourier transform ($\hat{\mathbf{f}}$) of a signal, $\mathbf{f}$ can be written as,

$$\hat{\mathbf{f}} = \psi^T \mathbf{f} \tag{14}$$

Therefore, the inverse to get $\mathbf{f}$ back is defined as,

$$\mathbf{f} = \psi^T \hat{\mathbf{f}} \tag{15}$$

Where we Are Going → Filtering

- The goal is to estimate the change in sample density along a manifold that is represented by a graph
- **Mathematical Filter:** Take in a signal and attenuate it according to a frequency response function
- In the simplest case, a signal can be filtered according to spectral properties of $\mathcal{L}$