COM SCI C121 Week 8

Aidan Jan

May 28, 2024

Matrices

- A matrix can be used to store data
- Consider a matrix of dimension $m \times n$
- The rows can be a high-dimensional sample of dimension n
- That means that each column is *one* dimension across *all* samples, m

The goal is to summarize the data in the matrix to see relationship across samples.

- The matrix itself is described by $m \times n$ data points, which on its own is unwieldy
- Naturally, we would want a way to summarize n data points into an "intuitive" representation that we can make sense of
- One way to think about this is creating a "faithful" representation of $m \times n$ data points into, say, $n \times 2$ data points while preserving structure in the data.
  - Taking this to the extreme, we will look at linear algebra for inspiration
- In an ideal case, samples that are more "similar" will "cluster" together. For example, if I put tumor samples and normal samples, one would expect the tumor samples to be more similar to each other and the normal samples to be more similar to each other.

What is a Matrix in Linear Algebra?

In short, a matrix is code for a linear function (to transform a set of vectors to another). For example,

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \to \begin{cases} s = x + y \\ t = 0x + y \end{cases}$$

where x and y are the x and y of the original vector, and s and t are the components of the new vector.

Singular Value Decomposition (SVD)

What is singular value decomposition about?

- Linear transformations, and their corresponding matrices (which are rectnagular tables filled with numbers), are seemingly complicated and arbitrary.
- The singular value decomposition (SVD) says that every matrix is essentially diagonal, i.e., "nice", provided the "right" bases are used for the domain and range spaces.
- By finding the "right" bases, the SVD provides fundamental insights into linear transformations and their accompanying matrix representations.

SVD Algorithm

- **Input:** an $m \times n$ matrix
- **Output:** a set of numbers called *singular values* and a two collection of vectors: a set of *right singular vectors* and another set of *left singular vectors*.

$$\begin{bmatrix} \\ M \\ \\ \end{bmatrix} = \begin{bmatrix} \\ U \\ \\ \end{bmatrix} \times \begin{bmatrix} \\ \Sigma \\ \\ \end{bmatrix} \times \begin{bmatrix} \\ V^* \\ \\ \end{bmatrix}$$

. . . where:

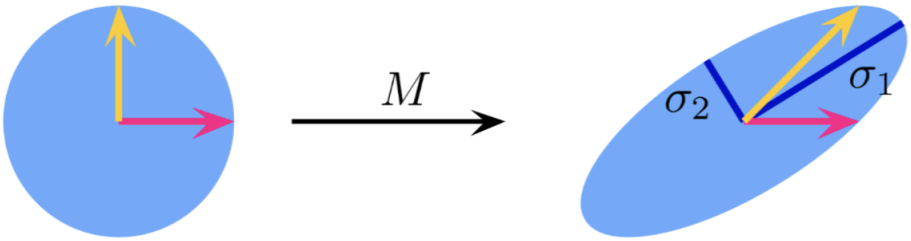
- M is a $m \times n$ matrix
- U is a $m \times m$ matrix
- Σ is a $m \times n$ matrix
- V^* is a $n \times n$ matrix

additionally

- U and V^* have the property that their transposes equal their inverses.

The meaning of *singular values*

- As a linear map, an $m \times n$ matrix M can be thought of as mapping a vector x from R^n to R^m.
- A unit sphere in R^n is mapped to an ellipsoid in R^m
- The non-zero *singular values* of M are the lengths of the *semi-axes* of the ellipsoid.



Measuring directions of distortion

- The maximal singular value can therefore be understood to be the size of the vector that points in the direction in which the linear transformation corresponding to M has the largesst effect.
- Formally, the maximal singular value, which is usually denoted as σ_1 can be understood to be

$$\sigma_1 = \max_{x:\|x\|=1} \|Mx\|$$

- Similarly, the smallest non-zero singular value is the size of the smallest semi-axis of the ellipsoid that is the image of the unit sphere under M.

Calculating SVD

- Let $M = U\Sigma V^T$ and set $M_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. The matrix M_k is a good low-rank matrix approximation of M. Specifically,
$$\min_{rank(X)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$
- This theorem makes precise the intuition that the top singular values, which measure the sizes of the largest of the semi-axes of the ellipsoid resulting from mapping of a sphere by the linear transformation corresponding to a matrix, capture "most" of the transformation.

A Centered Matrix can be Transformed to Summarize the Covariance of the Data

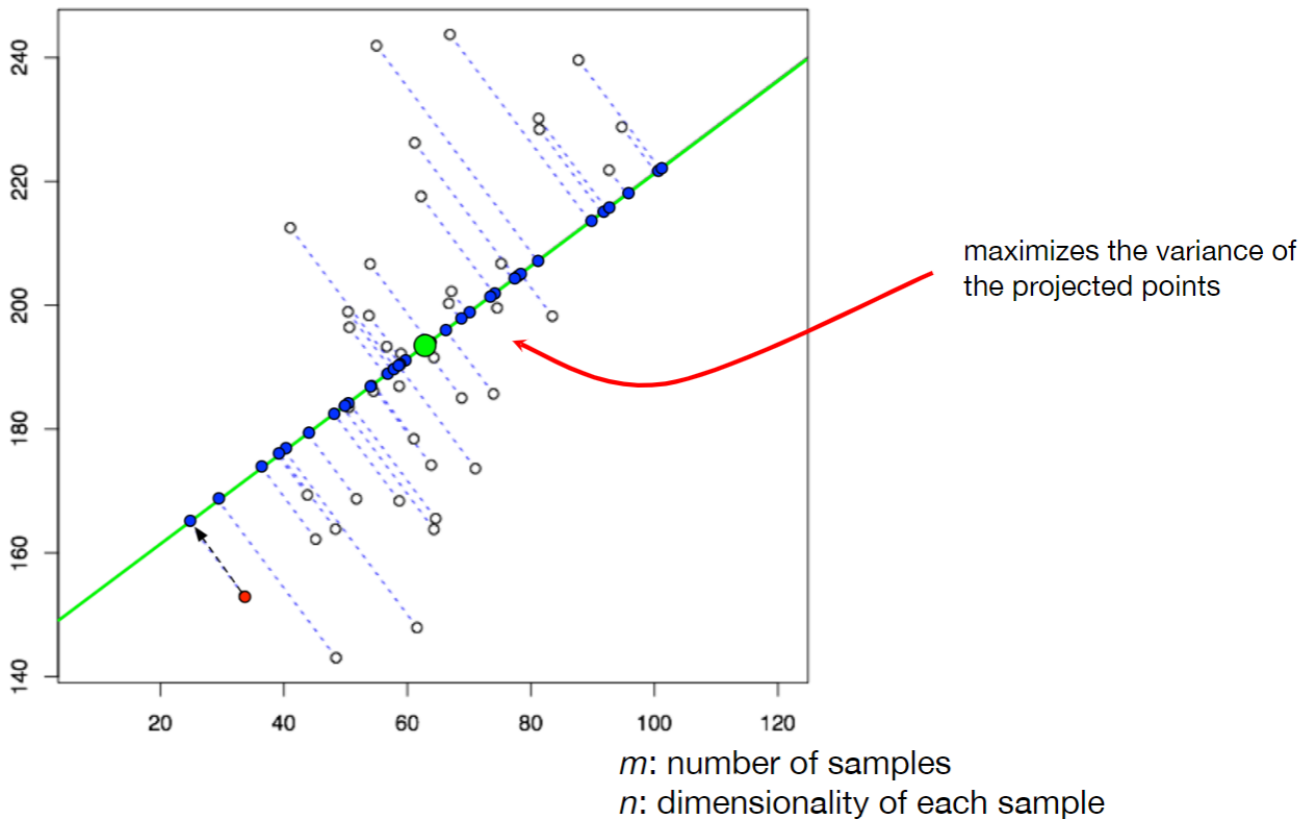
- Take a matrix A with $m \times n$ data points. Create M by subtracting the mean of each column from that corersponding column
  - $M_{ij} = A_{ij} - \frac{1}{n}\sum_{k=1}^{n} a_{kj}$
  - M is now a matrix that is *centered*
- $M^T M$ is a *covariance matrix* of A with the property:
  - $(M^T M)_{ij} = \text{Cov}(A_i, A_j)$
  - In words, $M_i j$ represents how much samples i and j covary with each other
    - If $(M^T M)_{ij} > 0$, samples i and j are positively related
    - If $(M^T M)_{ij} < 0$, samples i and j are negatively related
    - If $(M^T M)_{ij} \approx 0$, samples i and j are unrelated.

PCA with SVD

- The SVD of a (centered) M, given by $M = U\Sigma V^T$, yields a decomposition of $M^T M$ ad $M^T M = V\Sigma^2 V^{-1}$, i.e., eigendecomposition of the covariance matrix $M^T M$ can be performed by SVD of M.Set V_k to be the first k columns of V, i.e., $V_k = [v_1, v_2, \ldots v_k]$. Then the projection of the points in M by V_k, i.e., PCA(k) = MV_k has numerous useful properties.

An Example of a PCA Projection

- Each dot represents a sample
- Remember, M is $m \times n$, V is $n \times n$.
- If I take the first two columns of V:
  - $PCA(2) = MV_2$ results in a $m \times 2$ matrix.

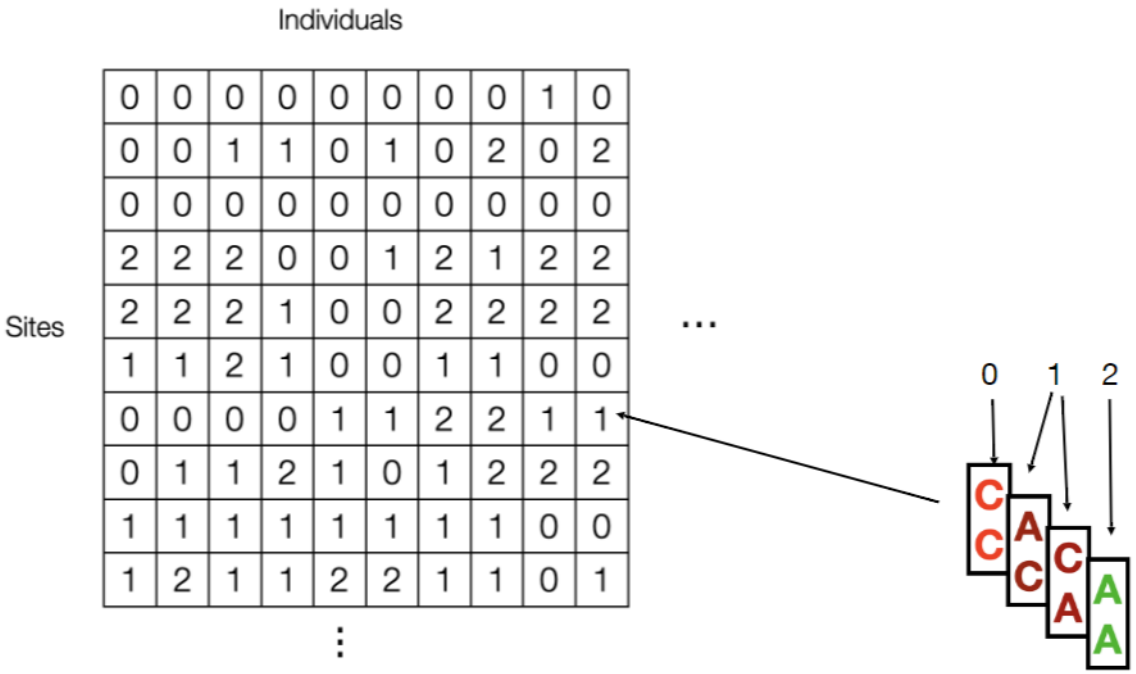


PCA Steps:

- Start with a data matrix A.
- Center A to get M.
- M has a singular value decomposition that is derived from viewing M as a linear transformation. $M = U\Sigma V^T$.
- The matrix V consists of the eigenvectors which diagonalize the covariance matrix $M^T M$.
- Compute V from M using the SVD.
- Let V_k be the truncation of V to its first k columns. We know from linear algebra that this is a meaningful restriction because $M_k = U_k \Sigma V_k^T$ is a good low-rank approximation to M.
- Project the data matrix M with V_k to obtain a new set of points: MV_k.
- The projection has the property that it will maximize the variance of the projected points.

An Application of PCA: the Human Genotype Matrix

- Differences between any pair of human genomes are largely in the same sites, and consist of single nucleotide polymorphisms (SNPs).
- Most human SNPs are biallelic.



"Genes Mirror Geography Within Europe"

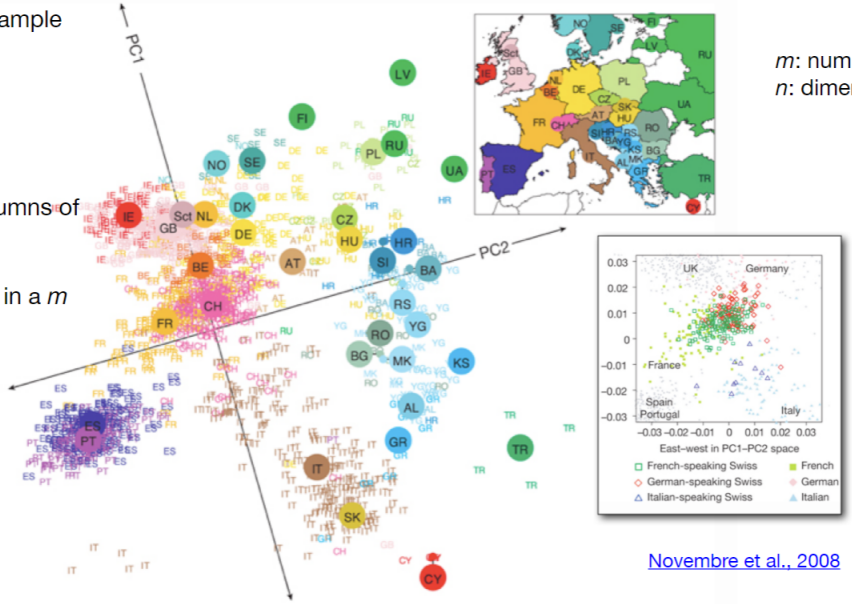
Each dot represents a sample (person) with *n* SNPs

M is *m x n*

V is *n x n*

If I take the first two columns of *V*:

$PCA(2) = MV_2$ results in a *m x 2* matrix



m: number of samples
n: dimensionality of each sample

Novembre et al., 2008

Some Properties of the PCA

- Each subsequent dimension explains less variance than the previous
  - i.e., there are *diminishing returns* by including additional PCs
- The singular values, σ_i are related to how much variance each dimension explains
  - Proportion of variance explained: $\sigma_p^2 l \sum_{j=1}^{n} \sigma_j^2$

Choosing k in K-means Clustering

As a review, K-means minimizes the following loss function:

$$L(\mu, \alpha) = \sum_{k=1}^{K} \sum_{i=1}^{n} \|x_i - \mu_k\|_2^2 \mathbb{1}\{\alpha_i = k\}$$

where

- x_i is the data
- μ_k is the cluster center (mean of the data)
- $\mathbb{1}$ is a function that returns 1 if the condition is true. In this case, 1 if the data belongs to cluster k, 0 otherwise.

What happens if k increases and approaches n?

More thoughts

- Conceptually, you want a clustering that satisfies the *good clustering principle:*
  - "Every pair of points from the same cluster should be close to each other than any pair of points from different clusters."
- Does setting $k = n$ satisfy this principle?

In Reality

Technically yes. But then the data would not be useful.

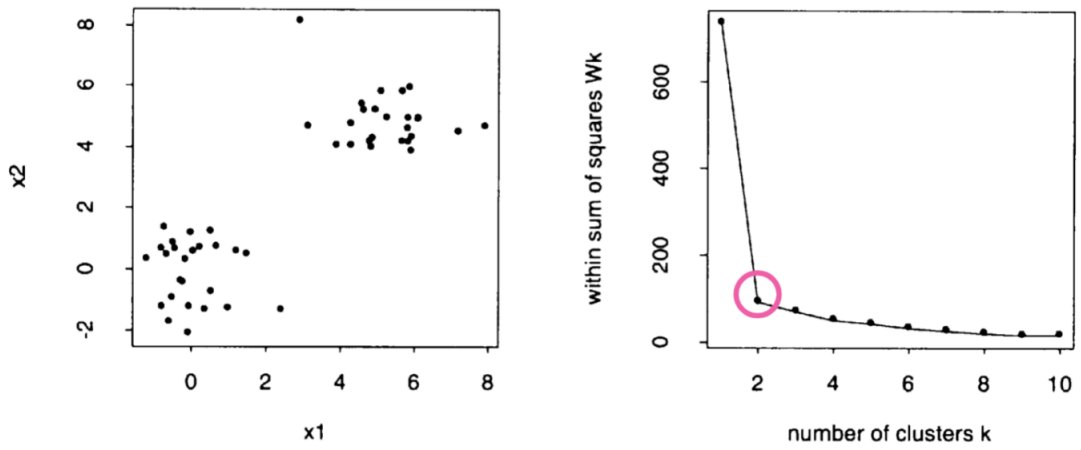
- We want to satisfy the good clustering principle with the *simplest* model that sufficiently captures the complexity of the data

The Good Clustering Principle Mathematically

- Let $d_{ii'} = \|x_i - x_{i'}\|_2^2 = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$, the pairwise distance between two points.
- Let $D_r = \sum_{i,i' \in C_r} d_{ii'}$ be the pairwise distance between all points in a cluster C_r.
- Then

$$W_k = \sum_{r=1}^{K} \frac{1}{2n_r} D_r$$

  - where n_r is the number of points in cluster r.
  - What is k-means with respect to W? K-means minimizes W. (W is the same objective function as the loss function above.)



Graphically, we are looking for the value of k at the "elbow". Unfortunately, a lot of the times, the graph isn't that clean.

Gap Statistic

A more principled approach: the Gap statistic is the difference in the expected W under a *reasonable null distribution* and the observed W.

$$\text{Gap}_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

To see how to find an appropriate reference distribution, consider for a moment the population version corresponding to the gap statistic in the case of K-means clustering:

$$g(k) = \log\left\{\frac{\text{MSE}_{X^*}(k)}{\text{MSE}_{X^*}(1)}\right\} - \log\left\{\frac{\text{MSE}_X(k)}{\text{MSE}_X(1)}\right\}$$

Gap Statistic in Practice

We consider two choices for the reference distribution:

(a) generate each reference feature uniformly over the range of the observed values for that feature

(b) generate the reference features from a uniform distribution over a box aligned with the principal components of the data. In detail, if X is our $n \times p$ data matrix, assume that the columns have mean 0 and compute the singular value decomposition $X = UDV^T$. We transform via $X' = XV$ and then draw uniform features Z' over the ranges of the columns of X', as in method (a) above. Finally we back-transform via $Z = Z'V^T$ to give reference data Z.

Method (a) has the advantage of simplicity. Method (b) takes into account the shape of the data distribution and makes the procedure rotationally invariant, as long as the clustering method itself is invariant.

Estimation of the Gap

Steps:

1. cluster the observed data, varying the total number of clusters from $k = 1, 2, \ldots . K$, giving within-dispersion measures $W_k, 1, 2, \ldots, K$.

2. generate B reference data sets, using the uniform prescription (a) or (b) above, and cluster each one giving within-dispersion measures W_{kb}^*, $b = 1, 2, \ldots, B, k = 1, 2, \ldots, K$. Compute the (estimated) gap statistic

$$\text{Gap}(k) = \frac{1}{B}\sum_b \log(W_{kb}^*) - \log(W_k)$$

3. let $\bar{l} = \frac{1}{B}\sum_b \log(W^*_{kb})$, compute the standard deviation

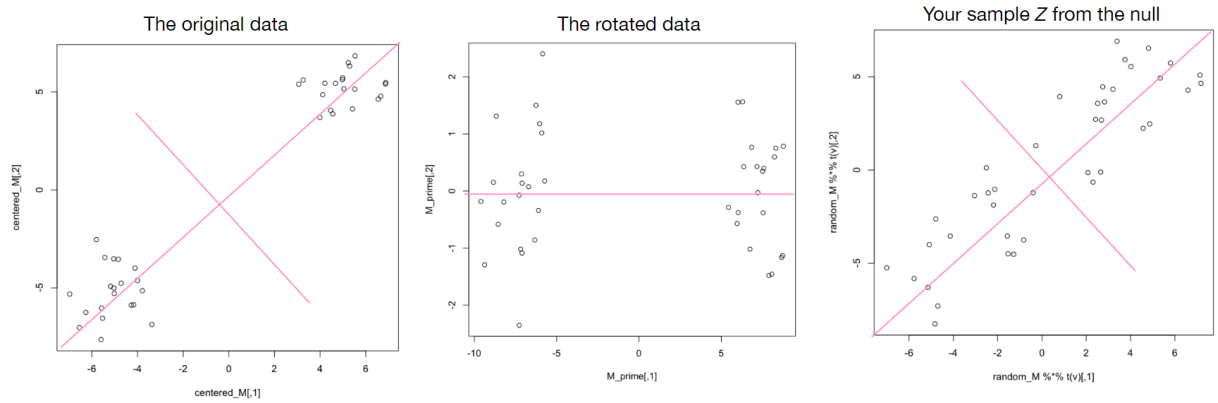
$$\mathrm{sd}_k = \left[\left(\frac{1}{B}\right)\sum_b \{\log(W^*_{kb}) - \bar{l}\}^2\right]^{\frac{1}{2}}$$

and define $s_k = \mathrm{sd}_k\sqrt{(1+\frac{1}{B})}$. Finally, choose the number of clusters via

$$\hat{k} = \text{smallest } k \text{ such that } \mathrm{Gap}(k) \geq \mathrm{Gap}(k+1) - s_{k+1}$$

The Gap null distribution

We transform the singular value decomposition $X = UDV^T$ via $X' = XV$ so that the uniform features Z' over the ranges of the columns of X'. Finally, we back-transform via $Z = Z'V^T$ to give reference data Z.

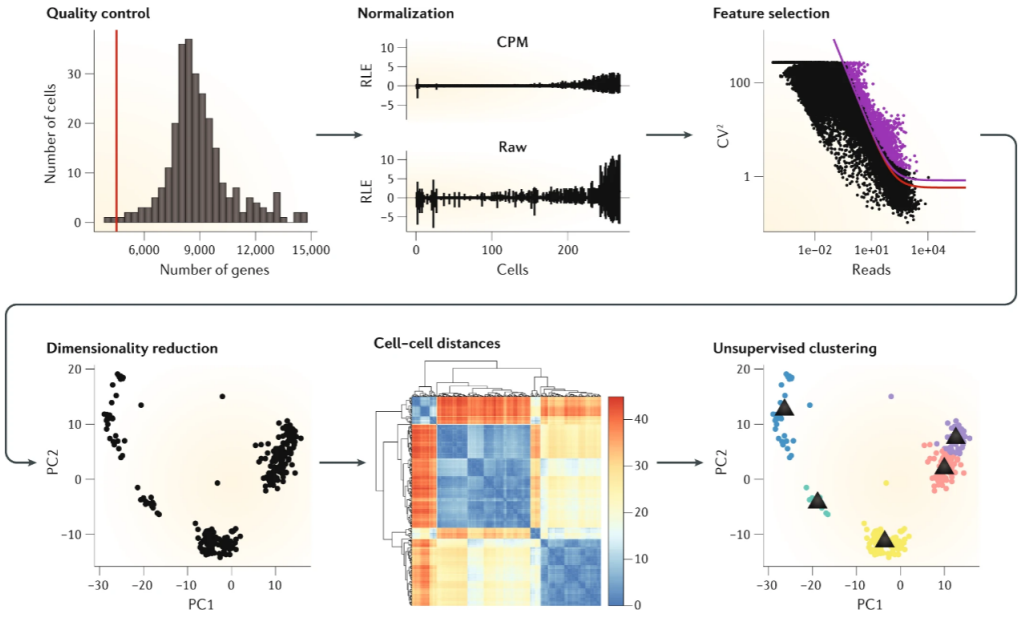


Nonlinear Dimension Reduction Techniques

Broad goals from scRNA-seq:

- Imagine you have a set of cells that are heterogenous, can you tell me interesting things about the cell-types and their characteristics?
- What makes a *cell-type*?

The Data Processing Pipeline



Filtering and "Feature Selection"

Feature selection is the process of choosing which genes you're going to include versus not.
Why is this important?

- Practically speaking, zeroes screw up the denominator.

$$\hat{s}_{jg} = \frac{X_{jg}}{\left(\prod_{k=1}^{N_{\text{samples}}} X_{kg}\right)^{1/N_{\text{samples}}}}$$

- What if my true floor was 1 and not 0?

Lowly expressed genes have unreliable variance and thus should be filtered.

Goals of Dimension Reduction

In general, you want a faithful representation of your *high-dimensional* data in *low-dimensions*
In scRNA-seq, we usually think about a cell being a sample in high-dimensional gene space: $\mathbb{R}^G \to \mathbb{R}^D$ where $D << G$.

$$f\left(\begin{bmatrix} X_{11} & X_{12} & \dots & X_{1C} \\ X_{21} & X_{22} & \dots & X_{2C} \\ \dots & & & \\ X_{G1} & X_{G2} & \dots & X_{GC} \end{bmatrix}\right) \to \begin{bmatrix} X'_{11} & X'_{12} & \dots & X'_{1C} \\ X'_{21} & X'_{22} & \dots & X'_{2C} \\ \dots & & & \\ X'_{D1} & X'_{D2} & \dots & X'_{DC} \end{bmatrix}$$

Data is transformed into a Low Dimensional Representation. $1 \dots C$ (rows in both matrices) represent the cells, and the columns represent genes (G) and mappings (D).

Faithful Representation?

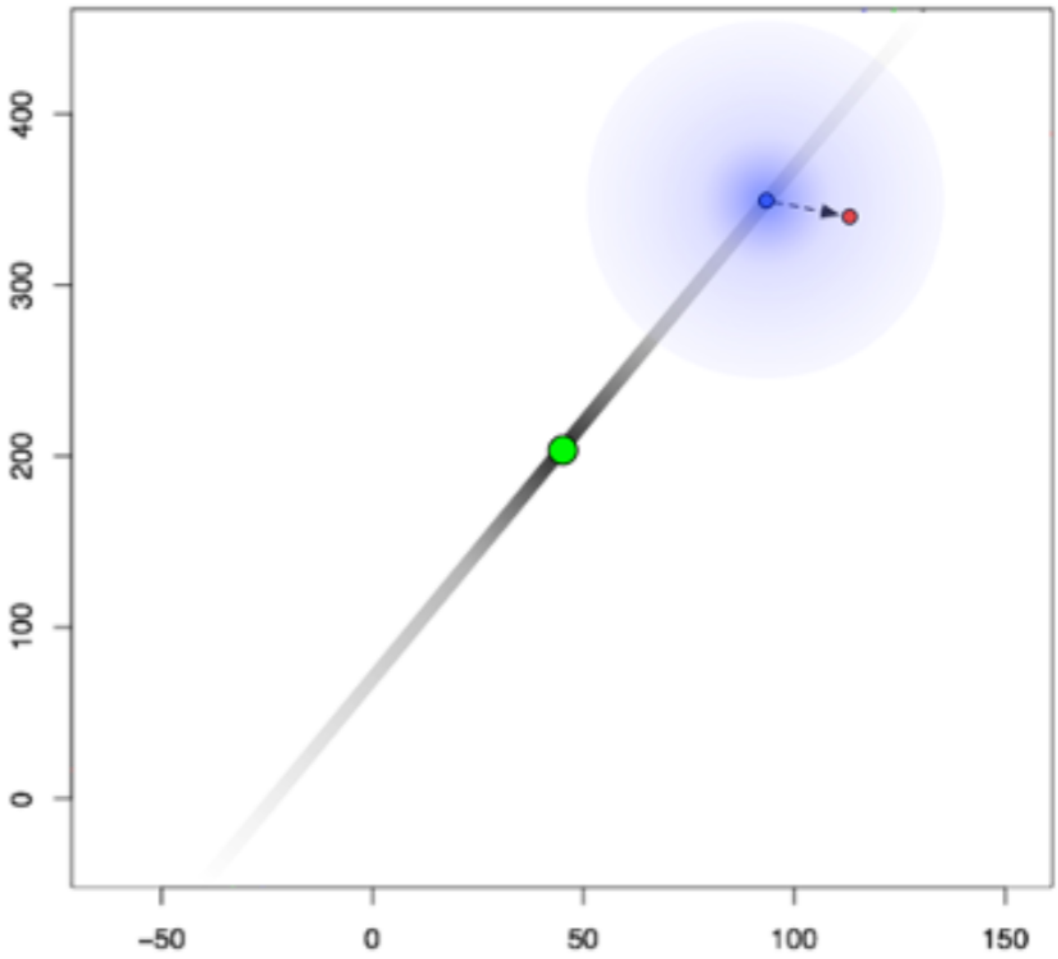
Recall: PCA projects the data such that the maximum variation is explained in low dimenison.
However, it is somewhat subjective.

- What distances do you want to preserve? Overall? Local?

- Actually, what is the distance?
- Different distant functions behave differently
- t-SNE takes an opinionated approach (you'll see on next few slides)

Another View of PCA: Probabilistic PCA

- Assume that points are generated from Gaussian distributions along a line (in this case, the black line, centered around the green point).
- The points are randomly perturbed (Gaussian noise, equal variance), in both directions (up and down, left and right)
- The goal is the find the black line using only the perturbed points as input
- Note: this problem formulation is also known as 'total least squares'.



Steps:

- Draw $x \sim N_D(0, I)$ in low dimensional space
- Transform it into higher dimensional space by the principle axes
  - $t|x \sim N(Wx + \mu, \sigma^2 I)$
  - I represents the identity matrix, and W is a linear transformation

t-distributed Stochastic Neighbor Embedding

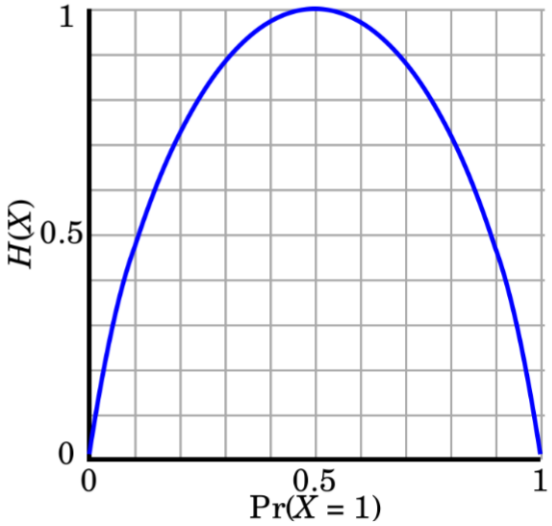
- Goal: find a *distribution* in low-dimensions (often 2 or 3) that represents a high-dimension distribution reasonably.
- Approach:
  - Let the "true" high-dimensional distribution (your data) by represented by distribution P.
  - Approximate the high-dimensional distribution with a low-dimensional one, Q.
  - Minimize the Kullback-Leibler divergence of P given Q.

$$KL(P\|Q) = \sum_{i=1}^{N} \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

  - *"KLD is the expected excess surprise from using Q if the true distribution is P"*

Short aside: Entropy is a Way to Summarize a Distribution

- Entropy is defined as:
  - $H(X) = -\sum_{i=1}^{N} p(x_i) \log(p(x_i)) = -E[\log(p(X))]$
- "How surprising an observation is on average"
- Why is this "surprisal"?



Some Useful Properties of the KLD

- $KL(P\|Q) \geq 0$, always.
- It is *not* symmetric $KL(P\|Q) \neq KL(Q\|P)$.
- The "further away" Q is from P, the larger $KL(P\|Q)$ is.

$$KL(P\|Q) = \sum_{x} p(x) \log(\frac{p(x)}{q(x)})$$

- Given the points above, KLD is useful as a *distance*, but is **not a distance metric** in the strict sense.

t-distributed Stochastic Neighbor Embedding

- Approach:
  - Let the "true high-dimensional distribution (your data) be represented by distribution P.
  - Approximate the high-dimensional distribution with a low-dimensional one, Q.
  - Minimize the Kullback-Leibler divergence of P given Q: $KL(P\|Q) = \sum_{i=1}^{N} \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$
- TODOs:
  - Define p on data (reference distribution)
  - Define q on data (generating distribution)
  - Minimuze KLD?

Modeling the "probability" in high-dimensions

- Note, this notion of "conditional probability" is a bit... strange? But it's an assumption one can make
- First, define our data as samples $x_k \in R^G$
- $p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2//2\sigma_i^2)}$, where $p_{i|i} = 0$
  - x_k = our data
  - G = the observed dimension of our data
- Make it symmetric, so define $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$
- Very similar to soft k-means distance

We have defined P, now define Q, our "reference" distribution in lower dimension

- Recall our goal: Minimize the Kullback-Leibler divergence of P given Q.

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- Q is a low-dimensional representation of P that minimizes the KL divergence.

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

  where y are literally made up points in some lower dimensional space
- t-distribution has much heavier tails and thus alleviates "crowding"
- Notice the numerator is essentially a Student t-distribution.

The Perplexity Parameter

- Defined as $Perp(P_i) = 2^{H(P_i)}$, or the Shannon entropy.
  - You, the oracle, chooses a fixed value, say, 5.
  - Find σ_k^2 such that your perplexity is basically what you said it should be.

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

- Intuition: probability mass has to be shared (sum to 1), so setting larger bandwidth makes more "neighbors" and smaller fewer neighbors.
- Desired behavior: dense regions get smaller σ_i, sparse regions get larger σ_i.
  - "Small" values of perplexity favor local variation
  - "Large" values of perplexity favor global variation
  - Made up rule: perplexity between 5-50, because that is what *they* say.

P and Q are defined, what's the intuition?

- Compute "probabilities" (distances) in high-dimensions using the Gaussian kernel:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

- Choose the appropriate values of σ_i in order to get a reasonable set of "neighbors". Dense regions get smaller bandwidth, sparse get larger
- Finally, find the t-distribution Q in low dimension such that you minimize the KL-divergence:

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$