Recitation 6

3D Chromatin Structure & Dimensionality Reduction

Ge (Saber) Liu, Corban Swain


2020-03-12 / 2020-03-13
• DeepSEA
• Linear algebra basics
• Principle component analysis (PCA)
• t-SNE and parametric t-SNE
• Auto-encoder
• U-MAP
deep learning-based sequence analyzer (DeepSEA)
Linear algebra basics

Eigenvector: An eigenvector or characteristic vector of a linear transformation $T$ is a non-zero vector that changes by only a scalar factor when that linear transformation is applied to it.

$$T(v) = \lambda v \quad \text{or} \quad Av = \lambda v$$

if the transformation can be represented as a square matrix $A$ where $\lambda$ is a scalar, known as the eigenvalue, characteristic value, or characteristic root associated with the eigenvector $v$. An $N \times N$ matrix has at most $N$ linearly independent eigenvectors.

Eigen-decomposition: Factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigenvalues and eigenvectors.

$$A = Q\Lambda Q^{-1}$$

where $Q$ is the square $N \times N$ matrix whose $i$-th column is the eigenvector $v_i$ of $A$, and $\Lambda$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\Lambda_{ii} = \lambda_i$. Note that $A$ has to have $N$ linearly independent eigenvectors (Only diagonalizable matrices can be factorized in this way).

Singular value decomposition (SVD): factorization of a real or complex matrix $M_{m \times n}$ into $M = UV^*$ where $U_{m \times m}$ and $V_{n \times n}$ are unitary matrix with orthonormal eigenvectors of $MM^*$ and $M^*M$, and $m \times n$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal. $X^*$ means the conjugate transpose of $X$. 
Linear algebra basics

Special matrices

Real symmetric matrices: Matrix $A$ is symmetric if $A = A^T$

Theorem

Any symmetric matrix:
- has only real eigenvalues
- is always diagonalizable
- has orthogonal eigenvectors

Corollary: If matrix $A$ is symmetric then there exists $Q^T Q = I$ such that $A = Q^T \Lambda Q$.

Positive definite matrices: A symmetric matrix $A$ is positive definite/semi-definite if all its eigenvalues are positive/non-negative.

Theorem

$A$ is positive definite if and only if $x^T A x > 0, \forall x \neq 0$.

Diagonalizable $\supset$ Symmetric $\supset$ Positive semi-definite $\supset$ Positive definite
**Principal component analysis**: Orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components. This transformation is defined such that the first principal component has the **largest possible variance**, and each succeeding component in turn has the highest variance possible under the constraint that it is **orthogonal** to the preceding components. The resulting vectors are **linear combination** of the variables and form an **orthogonal basis set**.
Principal component analysis: Consider a data matrix $X_{m \times n}$ with $m$ examples of $n$ dimensional features (each dimension has been z-centered such that the mean is zero). Try to find a set of $n$-dimensional vectors of weights or coefficients $w(k) = (w_1, \ldots, w_n)$ that projects each row vector $x(i)$ of $X$ to a new vector of principal component scores $t(i) = (t_1, \ldots, t_n)_{(i)}$, given by $t_{k(i)} = x(i) \cdot w(k)$.

$$
\begin{bmatrix}
- x(1) \\
\vdots \\
- x(m)
\end{bmatrix}_{m \times n} \times \begin{bmatrix}
| & | & | \\
| w(1) & \ldots & w(n) |
\end{bmatrix}_{n \times n} = \begin{bmatrix}
- t(1) \\
\vdots \\
- t(m)
\end{bmatrix}_{m \times n} = \begin{bmatrix}
| & | & | \\
| PC(1) & \ldots & PC(n) |
\end{bmatrix}_{m \times n}
$$

$T = XW$

The weights are constrained to be a unit vector such that $w_{(i)}^T w_{(i)} = 1$. 

PCA
To solve for projection with maximized variance, the first weight vector has to satisfy:

\[ w_1 = \arg \max_{w^T w = 1} \sum_i (x_i \cdot w_1)^2 = \arg \max_{w^T w = 1} w^T X^T X w = \arg \max \frac{w^T X^T X w}{w^T w} \]

A standard solution to this optimization for a positive semidefinite matrix such as \( X^T X \) is the largest eigenvalue of the matrix, which occurs when \( w \) is the corresponding eigenvector. The rest of the component can be given as:

\[ w_k = \arg \max \frac{w^T X^T X_k w}{w^T w} \quad \text{where} \quad X_k = X - \sum_{s=1}^{k-1} X w_s w_s^T \]

It turns out that this gives the remaining eigenvectors of \( X^T X \), with the maximum values equal to the corresponding eigenvalues. Thus, solving weight vectors for PCA is equivalent to finding the eigenvectors of \( X^T X \) and sorting by its corresponding eigenvalues.

The SVD of \( X = U W^T \), so \( T = X W = U W^T W = U \). Each column of \( T \) is given by one of the left singular vectors of \( X \) multiplied by the corresponding singular value.
**Stochastic neighbor embedding (SNE):** An unsupervised nonlinear dimensionality reduction technique where the goal is to find a low-dimensional (2-dimensional) representation of the original inputs such that pairwise similarity are best preserved and the inherent clustering structure can be visualized.

**Similarity in original input space:** the similarity of datapoint $x_i$ to datapoint $x_j$ is defined as the conditional probability, $p(x_j|x_i)$, that $x_i$ would pick $x_j$ as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at $x_i$. Given an input matrix $X$, in which each row is a sample $x_i$ and each column represent a feature dimension, the pairwise similarity is defined as:

$$P_{i,j} = p(x_j|x_i) = \frac{\exp(-||x_i - x_j||^2/2\sigma^2_i)}{\sum_{k\neq i} \exp(-||x_i - x_k||^2/2\sigma^2_i)}$$

We define $\beta_i = 1/\sigma_i$ which is equivalent to the precision of a multivariant Gaussian. We further set constraints on the perplexity of the conditional distribution where perplexity is defined using Shannon entropy of $P_i$:

$$\text{Perplexity}(P_i) = 2^{H(P_i)} \quad \text{and} \quad H(P_i) = -\sum_j p_{x_j|x_i} \log_2 p_{x_j|x_i}$$

In order to guarantee the perplexity constraint for each $P_i$, we need to find the corresponding $\beta_i$ such that the resulting distribution has the desired perplexity. Given that $\text{Perplexity}(P_i)$ is a monotonically decreasing function of $\beta_i$, we could use binary search to estimate the solution for $\beta_i$. 
**t-distribution Stochastic neighbor embedding (t-SNE)**

There are several modifications we need to make:

- **Use symmetric similarity matrix instead:** Since the gradients for the conditional distribution is hard to compute, people use the joint distribution as an alternative that is "just as good" and this gives symmetric distribution matrix:

  \[
P_{ij}^{\text{symmetric}} = p(x_i, x_j) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_l - x_k\|^2/2\sigma^2)} = \frac{p(x_j|x_i) + p(x_i|x_j)}{2N}
\]

- **Use Student t-distribution for similarity in embedded space:** We look for a 2D representation of \(X\) which is \(Y\) (\(N \times 2\) matrix), such that the pairwise similarity on \(Y\):

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

  is similar to \(P_{ij}\). Which is is equivalent to minimizing the **KL-divergence** between \(P\) and \(Q\):

  \[
  C = KL(P\|Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}
  \]

  The gradients for conducting gradient descent are:

  \[
  \frac{\partial C}{\partial y_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij})(1 + \|y_i - y_j\|^2)^{-1}(y_i - y_j)
  \]
Evaluating distance using the $t$-statistic allows for pairs of points separated by large distances to maintain significant probability mass compared to that for a gaussian distance. This is because the $t$-distribution has a “heavier tail” than the gaussian distribution. We want to maintain the probability mass for these distances so during the gradient update of embedded locations there is a “force” to bring locally clustered points in high-D space together in low-D space, even if they are initialized as far apart in the t-SNE scheme.
The non-parametric t-SNE has several drawbacks for being non-parametric.

- You can’t embed new points that weren’t used in the training phase without running the algorithm from scratch again and without preserving the previous embedding results.
- It is not scalable because the more points you have the larger memory you need to store $D, P$ and $Q$. 

![Diagram of t-SNE algorithm]

**Diagram:**
- **Data (Nxd matrix)**
  - Compute matrix $P$ (2)
  - Matrix $P$
  - Compute matrix $Q$ (3)
- **Initialize $Y$**
  - $Y$
  - Optimize w.r.t. $Y$ using SGD
  - Matrix $Q$
- **Compute cost (4)**
  - Cost
We could think of a parametric approach instead, in which we want to build a model $y = f(x, \Theta)$ that maps any given input $x_i$ to low-dimensional output $y_i$. A useful family of model we would consider is of course the **neural networks**. The good thing about this is that we can calculate $P$ on a smaller batch of $X$ and train model using batched data. Also once the model is trained, we will have a deterministic embedding that can be calculated within linear time.

**parametric t-SNE**
Practical Considerations for Dimensionality Reduction

- It may be necessary to first process raw data using principal component analysis then use the top $k$ PCA loadings as a proxy for the raw data to perform t-SNE visualization. This could prevent issues and provide computational speed-up when trying to down-map from very high dimensional data to very low dimensional spaces.
**Auto-encoder**: An autoencoder learns to compress data from the input layer into a low dimensional representation, and then uncompress that representation into something that closely matches the original data.

\[ z = g_\phi(x) \quad x' = f_\Theta(z) = f_\Theta(g_\phi(x)) \quad \mathcal{L}(x, x') = \|x - x'\|^2 \]
**Uniform Manifold Approximation and Projection (UMAP):** a general dimension reduction technique based on manifold learning techniques and topological data analysis. The algorithm is founded on three assumptions about the data:

- The data is uniformly distributed on Riemannian manifold
- The Riemannian metric is locally constant (or can be approximated as such)
- The manifold is locally connected

The embedding is found by searching for a low dimensional projection of the data that has the closest possible equivalent fuzzy topological structure.
Dimension reduction algorithms

![Dimension reduction algorithms diagram](https://via.placeholder.com/150)

- **cancer**
- **PCA**
- **PCA + t-SNE**
- **t-SNE**
- **UMAP**

Class:
- benign
- malignant