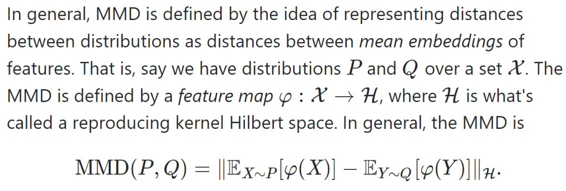
- autoencoder model with custom regularization at each layer to learn different properties of the data
- As SAUCIE reduces input dimensionality, regularizations on different layers reveal differ70 ent representations of the data: for visualization, batch correction, clustering, and denoising.

71 In order to achieve these representations we use customized regularizations in each layer. We
72 use the architectural choice of having a two-dimensional bottleneck layer to provide a visual73 ization of the data. We develop a novel batch-level maximal mean discrepancy (MMD)-based
74 penalty constraint to remove batch effects in the embedding layer. A customized sparse encod75 ing layer featuring our novel information-dimension (ID) regularization provides an automated
76 clustering of the data with no parametric assumptions on the shape or number of clusters. All
77 regularizations balance against reconstruction accuracy, which is the basic penalty in an au78 toencoder that steers the network convergence away from trivial solutions. Furthermore, this
79 penalty ensures that the final layer of the network provides reconstructed measurements that
80 are denoised; in the case of single-cell RNA sequencing data, this layer also naturally imputes
81 missing values

Maximum Mean Discrepancy

In general, MMD is defined by the idea of representing distances between distributions as distances between mean embeddings of features.

- https://stats.stackexchange.com/questions/276497/maximum-mean-discrepancy-distance-distribution



In general, MMD is defined by the idea of representing distances between distributions as distances between *mean embeddings* of features. That is, say we have distributions P and Q over a set $\mathcal{X}$. The MMD is defined by a *feature map* $\varphi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is what's called a reproducing kernel Hilbert space. In general, the MMD is

$$\mathrm{MMD}(P, Q) = \|\mathbb{E}_{X\sim P}[\varphi(X)] - \mathbb{E}_{Y\sim Q}[\varphi(Y)]\|_{\mathcal{H}}.$$

Kernel methods

- instance based learners -- instead of learning some fied set of parameters corresponding to the features of their inputs, they instead 'remember' the ith training example $x_i,\ y_i$ and learn a corresponding weight w_i for it

- prediction for unlabeled inputs i.e. a data point not in the training set is treated by application of a similarity function k called a kernel, between the unlabelled x' and each of the training inputs x_i

116 Specifically, we seek representations in hidden layers that are useful for performing the various
117 analysis tasks associated with single cell data. Here, we introduce several design decisions and
118 novel regularizations to our autoencoder architecture (Figure 1) in order to constrain the learned
119 representations for four key tasks:
120 1. visualization and dimensionality reduction,
121 2. batch correction,
122 3. clustering, and
123 4. denoising and imputation.

https://web.stanford.edu/class/cs168/l/l11.pdf

Spectral graph theory
$v^t L v \ = \ \Sigma_{i>j}(v_i \ - \ v_j)^2$
- Laplacian = degee on diagonal, -1 if (i, j) is an edge
    - Eigendecomposition of Laplacian
        - number of zero eigenvalues = number of connected components
        - low eigenvealues correspond to eigenvectors that will minimize sum of sq distance -- so neighbours will be assigned similar scores in v - visualization
        - high eigenvalues will seek to max sum of square distance - graph colouring
    -