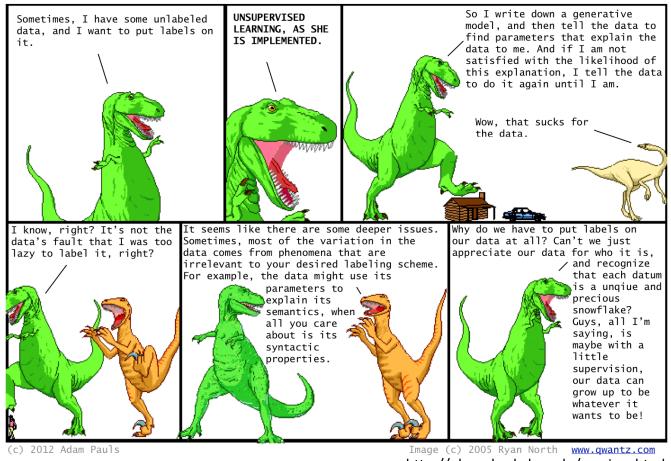
Unsupervised Learning: Summary

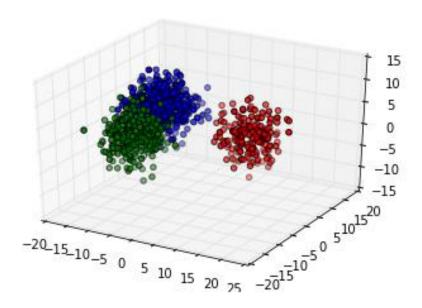


http://nlp.cs.berkeley.edu/comics.shtml

Slides from Hugo Larochelle, Geoffrey Hinton, and Yoshua Bengio CSC411: Machine Learning and Data Mining, Winter 2017

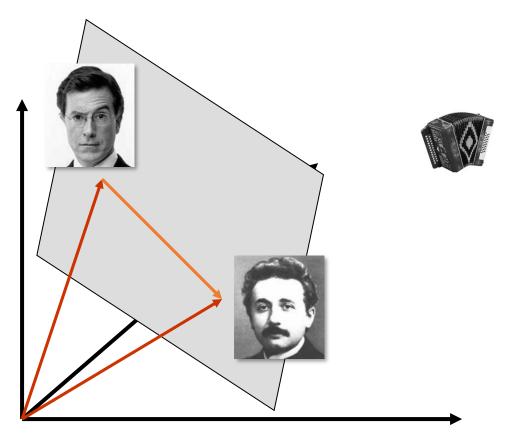
Mixture of Gaussians/k-Means

- Data: n-dimensional points in space
- Idea: the data data are organized into clouds
 - Find the centres (μ_i) and sizes (Σ_i) of the clouds!



PCA

- Data: n-dimensional points in space, centred around some point μ
- Idea: the centred x_i 's (i.e., $(x_i \mu)$'s) form a subspace: any x_i can be approximately reconstructured using $\widehat{x_i} \approx \mu + \alpha_1^i v_1 + \ldots + \alpha_k^i v_k$ for a small k
 - The points form a cloud that's not n-dimensional
 - Find a basis v_1, \dots, v_k (for a set k) s.t. $\sum_i (\hat{x}_i x_i)^2$ is minimized
 - The $\alpha_1 \dots \alpha_k$ encode most of the information about x
 - That's what lets us get a good reconstruction



- The set of faces is a "subspace" of the set of images
 - Suppose it is K dimensional
 - We can find the best subspace using PCA
 - This is like fitting a "hyper-plane" to the set of faces
 - spanned by vectors $\mathbf{v_1}$, $\mathbf{v_2}$, ..., $\mathbf{v_K}$
 - any face $\mathbf{x} \approx \overline{\mathbf{x}} + a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 + \ldots + a_k \mathbf{v}_k$

MoG+PCA

- First, find the clouds of points
- Then, apply PCA to each cloud separately

RNN, Word2Vec, etc.

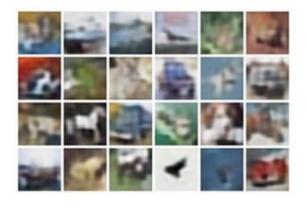
- Find good ways to represent the data by learning to predict the n-th data point from the (n-1)-st data point
 - Use supervised learning techniques even though we are technically doing unsupervised learning!

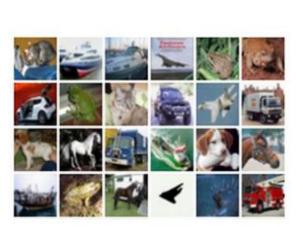
Autoencoders/"Diabolo networks"

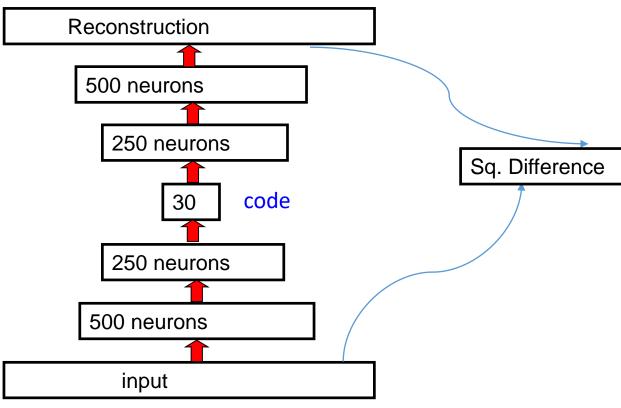


Goal

- Want to obtain good feautres of the training set
- Good features should allow us to be able to generate the training set







- Find the weights that produce as small a difference as possible between the input and the reconstruction
- Train using Backprop
- The code layer is a summary of the input
 - Somewhat similar to the alphas in PCA

Uses

- Can use to compress data
- Can use the encoder as a feature extractor
 - E.g., train autoencoder on unlabelled data, and then use it to extract features from labelled data to train classifiers