Unsupervised Learning: Summary

Sometimes, I have some unlabeled data, and I want to put labels on it.

UNSUPERVISED LEARNING, AS SHE IS IMPLEMENTED.

So I write down a generative model, and then tell the data to find parameters that explain the data to me. And if I am not satisfied with the likelihood of this explanation, I tell the data to do it again until I am.

Wow, that sucks for the data.

I know, right? It's not the data's fault that I was too lazy to label it, right?

It seems like there are some deeper issues. Sometimes, most of the variation in the data comes from phenomena that are irrelevant to your desired labeling scheme. For example, the data might use its parameters to explain its semantics, when all you care about is its syntactic properties.

Why do we have to put labels on our data at all? Can't we just appreciate our data for who it is, and recognize that each datum is a unique and precious snowflake? Guys, all I'm saying, is maybe with a little supervision, our data can grow up to be whatever it wants to be!

(c) 2012 Adam Pauls

Image (c) 2005 Ryan North  
www.qwantz.com
http://nlp.cs.berkeley.edu/comics.shtml

Slides from Hugo Larochelle, Geoffrey Hinton, and Yoshua Bengio

CSC411: Machine Learning and Data Mining, Winter 2017

Michael Guerzhoy
Mixture of Gaussians/k-Means

• Data: n-dimensional points in space
• Idea: the data are organized into clouds
  • Find the centres ($\mu_j$) and sizes ($\Sigma_j$) of the clouds!
PCA

- Data: n-dimensional points in space, centred around some point $\mu$
- Idea: the centred $x_i$’s (i.e., $(x_i - \mu)$’s) form a subspace: any $x_i$ can be approximately reconstructed using $\hat{x}_i \approx \mu + \alpha_1^i v_1 + \ldots + \alpha_k^i v_k$ for a small $k$

  - The points form a cloud that’s not n-dimensional
  - Find a basis $v_1, \ldots, v_k$ (for a set $k$) s.t. $\sum_i (\hat{x}_i - x_i)^2$ is minimized
  - The $\alpha_1 \ldots \alpha_k$ 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 $v_1, v_2, \ldots, v_K$
    • any face $x \approx \bar{x} + a_1 v_1 + a_2 v_2 + \ldots + a_k v_k$
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 features 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