Lecture 1(Densities & Sampling)

Why dimensionality reduction?

1. Local models perform poorly in higher dimensional spaces
2. High-dimensional methods are non-local along some direction
3. Clustering methods perform poorly in higher dimensional spaces
   Common operations on PDFs include the following:

- fitting a distribution to the data
- sampling from a distribution to generate new data points that follow the distribution
- factorizing a distribution to reduce complexity

How to sample from any PDF?

Great guide on sampling
There are several approaches for sampling from a distribution:

- inverse transform method
- rejection sampling
- importance sampling
- MCMC(Monte Carlo Markov Chain)
  In lecture 1, we only talk about inverse transform method, which is not applicable in high dimensional spaces.

Inverse Transform Method

Assumption: we are able to generate random variables with uniform distribution $U([0,1])$

Assume random variable we want has a cdf F(x) strictly increasing. Here is how it works:

1. Generate $X \sim U([0,1])$
2. Set $Y = F^{-1}(x)$
   then Y has cdf $F(x)$
   Example: suppose we want to generate a Bernoulli random variable X, (**discrete)** which case $P(x=0) = 1-p$ and $P(X=1) = p$ for some $p \in (0,1)$. Then the discrete inverse-transform yields:
3. Generate $U \sim unif(0,1)$
4. Set $X = 0$ if $U < 1-p$ and $X = 1$ if $U = 1-p$

More from the lecture notes: it is said that we have to discretize the domain of the pdf $P(X)$ and linearize $P(X)$ if it is multi-dimensional.

Density Estimation

Great summary on density estimation

Non-parametric Density Estimation

Typically, no distributional assumptions. The number of parameters scale with the number of training data. Examples:

- Histograms(# of bins grow exponentially with data's dimensionality + discontinuities at boundaries)
- Decision Trees
- KNN Classifier
- Kernel Regression
  4 things that make a nonparametric/memory/instance-based/lazy learner:

1. A distance metric : Euclidean, Manhattan, etc.
2. How many nearby neighbors/radius to look at?
3. A weighting function : W based on kernel K
4. How to fit with the local points? Average, Majority, Weighted vote

There are two methods that improve on histogram approach, yielding slightly better results. Both Parzen window estimator and KNN estimator are "non-parametric", but they are not free of parameters. We still have kernel scaling h and the # of neighbors k as hyperparameters. In addition, both estimators need to store all samples, which is why they are also called **memory methods**.

Kernel-based method(Parzen Window)

It is called Parzen Window Estimator. It fixes V and varies K

KNN method

This shares the same mathematical framework as above kernel-based method but it fixes K and varies V.

Parametric Density Estimation

Assume some model of the data in the form PDFs: Bernoulli, Multinomial, or **Mixtures** (of Laplacians, Gaussians). Then, estimate the parameters $(\mu, \sigma^2, \theta, \beta)$ using MLE/MAP and plug in
Pro: need few data points to learn parameters
Cons: strong distributional assumptions, not satisfied in practice

Unsupervised Learning

What can we learn from unlabeled data?

- Density Estimation
- Groups or clusters in the data
- Dimensionality reduction

Model Selection Problem

Hyperparameters must be optimized on a held-out dataset (validation set). When training data is limited, **cross-validation** is a better option. Do k training/evaluation runs using $n-1$ folds of a dataset as training data, while keeping the nth fold as validation set. Finally, select the best performing hyperparameters based on the runs.

Cross validation for unsupervised learning requires additional tricks. The trick is to optimize the DE hyperparameters by using the prediction of held-out samples as objective function. Basically, doing an MLE estimate on validation dataset.