CS5785 - Applied Machine Learning - Lecture 13

Prof. Nathan Kallus, Cornell Tech
Scribes: Karim Arem, Aobei Cheng

October 11, 2018

1 Recap - Unsupervised learning/Dimensionality Reduction

We can use dimensionality reduction when we have X $\in \mathbb{R}^{n*p}$ that is high dimensional data and want to transform it to a matrix Z $\in \mathbb{R}^{n*q}$ where $q < p$ that preserves the most information.

We can use PCA:
$z = e(x)$
$\hat{x} = d(z)$
$e(x) = Ax$
$d(Z) = A^T Z$
$A^T A = I$

The best A is $A = Vq$: first q columns from V in SVD of X

2 Multidimensional Scaling

We have the following dissimilarity measures and their usual uses in practice:

Dissimilarity measures	Usual Uses
Euclidean Distance	Vectors
Chi-squared Distance	Distributions
Cosine Similarity	Distance between bag of words, patterns
Edit Distance	Distance between strings

Multidimensional Scaling process: we start with the distance/dissimilarity matrix D $\in \mathbb{R}^{n*n}$.
What vectors D $\in \mathbb{R}^{n*p}$ recover D in the closest way via $D_{ij} \approx ||X_i - X_j||_2^2$ i.e. X is an embedding of the n abstract data points in p dimensions that maintains the dissimilarities as vector distances.

If indeed $D_{ij} = ||X_i - X_j||_2^2$ then

$D_{ij} = ||X_i||_2^2 + ||X_j||_2^2 - 2X_i^T X_j$ $D_{ij} = T_{ii} + T_{jj} - 2T_{ij}$
such that $T = XX^T \in \mathbb{R}^{n*n}$

$T = \frac{-1}{2}(D - \frac{Dee^T}{n} - \frac{ee^T D}{n} + \frac{e^T De}{n^2})$

In MDS, we eigendecompose T to get X such that $T = XX^T$, $T = UVU^T$ and $X = U_p V_p^{1/2}$, a reconstruction up to rotation and translation.

Fact: if D comes from euclidean distances then MDS is just PCA.

3 Clustering

:

The goal here is to assign data points to finitely many clusters. One natural perspective would be to find clusters in data. Another perspective is dimensionality reduction where $e(x) = 1, ..., k$

Encoder = cluster membership

Decoder = mean of that cluster =¿ $e(x) \in 1, ..., k$ and $d(z) = \hat{\mu_z}$

3.1 K-means Algorithm

The goal with k-means clustering is to assign data points to clusters such that the within cluster distance is small:

Let $c(i) = 1, ..., K$ indicate the assignment of point i.
The quality of c is defined in terms of the within-cluster distance:

$W(c) = \sum_{j=1}^{k} \sum_{i:c(i)=j} ||X_i - \mu_j||_2^2$ where:

$\mu_j = \frac{1}{n_j} \sum_{i:c(i)=j} X_i$
$n_j = \sum_{i:c(i)=j} 1$

$W(c) = \frac{1}{2} \sum_{j=1}^{k} \frac{1}{n_k} \sum_{i:c(i)=j} \sum_{i':c(i')=j} ||X_i - X_{i'}||_2^2$

What is the best C (min W(c))? How many C's are there? The number of ways of assigning n things to k buckets is far too big for us to be able to compute so we use a greedy algorithm (k-means):

1) Start with some C_0
2) For $t = 1, 2, ...$:

1. Compute the cluster means for C_{t-1}

$\mu_j^{(t-1)} = \frac{1}{n_j^{(i-1)}} \sum_{i:C_{t-1}(i)=j} X_i$ for every $j = 1, ..., k$

2. Reassign n data points to the closest mean:

$C_t(i) = argmin||x_i - \mu_j^{(i-1)}||_2^2$ for every $j = 1, ..., k$

Observation 1: different initialization lead to different solutions. Try different (random) starting points. Pick the final clustering with smallest W(c).

Observation 2: k-means will always converge in finite number of steps.

- Might not be to global option.
- Might take a long time so often we terminate after a fixed number of steps.
- Do not choose k by cross-validation.
- Increase k to decrease $\min_c W(c)$

- There do exist information theoretical ways to choose k, and there is no "right" or "wrong" way.

4 Soft Clustering

Hard clustering: k-means assigns each point to exactly 1 cluster

Soft clustering: sometimes it's not clear that there's a single cluster that a point belongs to, instead we want to partially assign points to clusters.

Suppose we have a binomial distribution as follows: $X_0 \sim N(\mu_0, \sigma^2)$

$Y \sim Bernoulli(\pi)$
$X = (1-Y)X_0 + YX_1 = \begin{cases} X_0 & Y = 0 \\ X_1 & Y = 1 \end{cases}$ $(X_1 Y = y) \sim N(\mu_y, \sigma_y^2)$
$Approach: find\ \theta = (\pi, \mu_0, \sigma_o^2, \mu_1, \sigma_1^2)\ that\ maximize\ the\ likelihood\ of\ the\ data\ (MLE)$