Unsupervised Learning

1 Clustering

Given a dataset without labels, or where all data-points are of the same class.

Two methods are clustering and dimensionality reduction.

The most basic algorithm for clustering is k-means.

K-means assigns cluster centers then all data points closer to it are associated with it. Then the total quadratic distance from the assigned center to the data points is minimized. The process of assigning, then optimizing the clusters is repeated until the optimization each iteration falls beneath a given threshold or meets some other terminating constraint.

The initial location of the cluster centers strongly affects the resulting split especially if the data is more uniform.

Sklearn k-means algorithm requires choosing the number of clusters, iterations, and the initializations (how many times does the algorithm cluster).

The final parameter initializations is useful for data that may be more sensitive to initial cluster assignment.

K-means is susceptible from local-minimum created by poor initialization of the cluster centers.

Local minimum are solutions that work but do not necessarily represent the optimal solution for the given dataspace.

2 More Clustering

Because there is no definition for clustering, it is very algorithm driven.

2.1 Single Linkage Clustering

Also known as SLC and it a hierarchical agglomerative clustering algorithm. A

We begin by considering each data-point a cluster.

We then define the intercluster distance as the distance between the closest two points in the two clusters.

Merge the two closest clusters and then repeat the algorithm $n - k$ times where k is the number of clusters we want.

Can represent the linking process with a hierarchical tree structure.

Using the tree structure can be useful in switching between clustering granularity.

There are multiple intercluster distance metrics that can be used such as mean, furthest and median distances.

Defining intercluster distance can be considered an aspect of domain knowledge.

The general running time of SLC is $O(n^3)$.

SLC is apt to find local minima especially if the clustering has seemingly isolated groupings.

2.2 Soft Clustering

Given a cluster thats equidistant from two other clusters, that center cluster will sometimes be assigned to one and sometimes assigned to others. It depends on how ties are broken.

Soft clustering is the approach of sharing points between clusters with different granularities.

Probability theory is the basis of soft cluster. That is data-points are considered probabilistically from different clusters.

It assumes that the data generated by the following process:

1. Select one of K Gaussians (with a known fixed variance α^2) uniformly.
2. Sample x_i from that Gaussian.
3. Repeat the process n times to sample n data-points.

Points can be sampled from the same Gaussian. The object is to try to find a hypothesis $h =< \mu_1, \ldots, \mu_k >$ that maximizes the probability of the data (a maximum likelihood hypothesis).

The Maximum Likelihood mean of the Gaussian is the mean of the data.

Use hidden variables to generate k clusters. That is given given k clusters each datapoint x_i has additional vector data added to it. Where $x =< x_1, \ldots, x_n >$ initially then several more features are added, where $z_j \in (0, 1)$ refers to membership to the j^{th} cluster. We then have $x =< x_1, \ldots, x_n, z_1, \ldots, z_k >$.

Expectation maximization is aesthetically similar to k-means. Two probabilistic calculations are called expectation where we do soft clustering and maximization where means are calculated.

Expectation is done using the equation $E[z_{ij}] = \frac{P(x=x_i|\mu=\mu_i)}{\sum_{i=1}^{k} P(x=x_i|\mu=\mu_i)}$

This uses Bayes Rule to determine the likelihood of x_i given μ_i and don't require the prior due to the maximum likelihood nature of the problem. The division is a normalization factor.

We calculate the mean of the cluster μ_i as $\mu_i = \frac{\sum_i E[z_{ij} x_i}{\sum_i E[z_{ij}]}$. The weighted average is required due to the soft assigning.

K-means can be derived from expected maximization using a hidden argmax.

EM is not forced to choose a cluster, however, most points even that should be clearly in one cluster will have a nonzero chance of being generated by another cluster.

Generally, the likelihood of each iteration is monotonically non-decreasing. That is with each iteration, we get either no likelihood gain or some.

However, this does not imply convergence, it simply will not diverge.

Since the initial location affects the placement of the initial means, the final implied mean of the Gaussian may accidentally find a local optima for the maximum likelihood of the Gaussian. This is fixed by randomly reassigning the initial means.

The algorithm is not focused on Gaussians, it works with any distribution. It simply requires determining the estimation and maximization equations for that distribution.

Usually the estimation part of the algorithm is the more expensive aspect. Small matter of mathematical algorithm derivation to determine the E and M steps.

2.3 Clustering Properties

There are three major properties of clustering algorithms.

- Richness: For any assignment of objects to clusters, there is some distance matrix D such that P_D (the clustering algorithm) returns that clustering $\forall c, \exists D$ such that $P_D = C$. This states that certain algorithms cannot generate certain clusters and is a measure of how rich the possible generation of clusters is and if there are so called 'blind spots' for the algorithm.

- Scale-invariance: Scaling distances by a positive value does not change the clustering $\forall D \forall K > D P_D = P_{KD}$. This implies that if a scaling factor occurs, it should not affect the clustering and the clusters produced by the P_D.
- Consistency: Shrinking intra-clustering distances and expanding intercluster distances does not change the clusterings $P_D = P_{D'}$. If the points in the clusters are drawn together and the clusters are moved apart, it should not change the clustering.

The impossibility theorem (by Kleinberg). No clustering scheme can achieve all three properties. Has the 'pick two' affect. The quality and specificity of each property can be tuned to retrieve a less precise algorithm.

3 Feature Scaling

How features are compared and analyzed is affected by their scale, having a common scale prevents any specific metric will not suffer from scaling issues.

Feature scaling reduces the scale between $[0, 1]$ and it prevents certain features from dominating over others.

The scaling formula is $x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$ where x is the pre-scaled feature and x' is the scaled feature.

Reliable outputs (between known bounds). However, outliers can skew the scale.

Sklearn has a MinMaxScaler.

SVMS and k-means clustering are affected by feature scaling. Algorithms stratified by two dimensions are affected when the boundary applies to both dimensions equally.

4 Feature Selection

Feature selection is useful for interpretability as well as dealing with the curse of dimensionality.

Feature selection is the use of an algorithm to determine the useful features for a given model and to isolate them.

The process of feature selection is exponential.

Feature selection is NP hard.

Two methods of handling the problem which is filtering and wrapping.

4.1 Filtering and Wrapping

Filtering works by pushing the feature set through an algorithm and the algorithm returns the set selected features, this is passed to a learning algorithm for training. The criterion is analyzed inside the search engine.

Wrapping instead uses the output of the learning algorithm in the search process for determining the best feature subset.

With filtering, there is no feedback between the learning algorithm and the filter.

The pros and cons of filtering are:

- Pro: Approximation and other techniques can be used to increase speed.
- Con: Using speed causes features to be analyzed in isolation.
- Con: Ignores the learning problem.

The pros and cons of wrapping are:

- Pro: Takes into account the model bias.
- Pro: Worried about the learning process itself.
- Con: Very slow since it has to run the learning algorithm on each iteration.

Decision trees are a kind of filtering algorithm.

Filtering algorithms are not allowed to directly interact with the learning algorithm. However they are able to understand the features in relation to the labels.

It is possible to use decision trees to find subsets of features, the features the tree uses are then passed as the subset to be used by the learner.

Determining the filtering criterion is extremely important. Several different criteria classes area common:

- Information gain
- Entropy (variance), Gini index, (doesn't depend on the label)
- Useful features selected by learning algorithms as having the most weight in predicting labels
- Independent/Non-redundant features

Reducing the feature set is useful until there are not enough features to classify/predict properly.

The scoring criterion for wrapping is a little different as the results of the learning algorithm come in to play, allowing for different range of scoring and evaluation methods.

- Hill climbing, gradient search/descent over reduction of error
- Randomized optimization (genetic algorithms, etc)
- A*
- Forward sequential selection
- Backwards elimination

Forward search begins the analysis with all the features in isolation. It then ranks the best feature and then combines that with the remaining features through an iterative process of testing, ranking and combining until either the error is minimized, all features are used or some other threshold criteria has been met.

Forward search's terminating criteria for error minimization can be tooled between algorithm speed (how many iterations) and its precision by analyzing the *gain* from each iteration and if that falls beneath a specific threshold, then the search is halted and the results thus far are given as the solution.

Forward search is essentially a hill climbing method.

Backwards elimination runs the forward search process in reverse, until the subset achieved on the final iteration causes an error gain that has been deemed to substantial.

Choosing the criterion/optimization function is using domain knowledge, however it is unable to take advantage of the bias of the learner.

Wrapping can take advantage of that bias, but it needs to deal with the running time of the learning algorithm and how that affects the exponential nature of feature selection.

4.2 Relevance and Usefulness

A feature x_i is strongly relevant if removing it degrades the Bayes Optimal Classifier's performance. This considers the BOC for all the features and then compares that to the performance of the BOC without the feature x_i.

A feature x_i is weakly relevant if

1. It is not strongly relevant

2. $\exists$ a subset of features S such that adding x_i to S improves the Bayes Optimal Classifier's performance.

This compares the performance of the BOC when using a subset without the feature x_i and then comparing it to the performance of the BOC utilizing the subset with x_i included

A feature x_i if not strongly relevant or weakly relevant, is irrelevant.

Usefulness measures the effect of a feature x_i on a particular predictor (that is to the actual case).

Relevance is concerned with information, that is the maximum performance cap of any particular predictor.

Whereas usefulness is concerned with a feature's effect on the accuracy of any particular predictor (which is guaranteed to be *less* than the performance of the BOC), that is its concerned with the error of some particular model or learning algorithm.

Clustering can be used as a feature transformation method.

Relevance is usefulness with respect to Bayes Optimal Classifier.

5 PCA

Principle Component Analysis is known as PCA.

Depending on the structure of the data, it it is mappable to a function f, then we can use that function to reduce that dataset. Not unlike the kernel in SVMs.

Principal Component Analysis can only using linear transformations to data, therefore data that has complex behavior will not be as ready for PCA as simpler explanations.

PCA finds a new coordinate system for a given dataset by translation and rotation only.

The origin on the new coordinate system is moved towards the center of the data.

The x-axis is lain along the principle axis of greatest variation, that is the line where the data changes the most.

Any other axes are mapped orthogonally to this axis, in $x - 1$ additional dimensions where x represents the dimensionality of the dataset.

PCA outputs vectors, normalized to a length of 1.

Eigenvalues are used to decompose PCAs.

PCA is applicable to complicated data, whether that transformation is useful is another matter.

The major axis 'dominating' implies that its eigenvalue is much larger than the other axes.

5.1 Measurable vs. Latent Features

Latent variables are variables that cannot be directly measured but have aspects which are quantifiable. These quantifiable aspects are measurable features.

One method of computing latent variables is to use feature selection.

There are two methods of granularly choosing how many features to keep via feature selection and those are:

- KBest (the top k features)
- Percentile (the top k% of features)

Preserving information, and feature selection is done in a scenario where there is a smaller number of features then those captured that actually drive the phenomenon being analyzed. Furthermore, the process of feature transformation is done by making a component feature (via PCA or other means) that tries to more directly probe the phenomenon.

5.2 PCA continued

PCA tries to find that minimizes the amount of information lost. This is why we choose the axis of greatest variance.

PCA projects the data onto the axis of greatest variance.

The variance is the willingness of flexibility of an algorithm to learn it also refers to the spread of the data in a distribution (related to the standard deviation).

An area surrounding the data is taken, where the points within are considered in order to choose the dimensionality of maximum variance.

The longer axis in that area is the axis of maximal variance.

This axis is referred to as the principle component.

The amount of information lost is represented by the distance between the data-points and the principle component axis. That is the amount of information lost is $\sum_i \text{dist}(x_i, p_i)$ where x_i is one of the points from the dataset and p_i is the nearest point on the principle component axis.

It is possible to use gradient descent to determine the PC axis assuming all is computable.

Choosing the PCA is choosing the axis that minimizes information loss.

PCA can combine the measurable features into new combinations and then rank those combinations to find the most useful feature reductions.

These ranked features are called first, second, third, ...nth principle components.

The maximum number of principle components possible is the total number of features of the dataset.

PCA is a systematized way to transform input features into principle components. Those principle components are then used as new features in training a model. The principle components are the directions in the data that maximize variance (minimize the information loss). These principle components are ranked by how effectively they minimize information loss, those that more effectively minimize it are ranked higher.

PCA is used when there are latent features responsible for the phenomenon in the data. Dimensionality reduction is another reason as it allows you to visualize high dimensional data, to reduce noise, and to use for preprocessing for other algorithms (to prevent those algorithms from suffering from issues around high dimensionality).

6 Feature Transformation

Feature transformation is the problem of preprocessing a set of features to create a (smaller/more compact) feature set. This is done while retaining as much relevant information as possible.

Feature selection is a subset of feature transformation.

We consider linear feature transformation.

Given a feature x from a set of features F^n where n is the dimensionality of the feature set. We then transform this feature space such that we arrive at another feature set F^m where $m < n$ usually and $P^T x$ is the linear transformation operator with a projection matrix p which is used to transform points in the instance space into the secondary space.

The final feature set F^M is a set of features that are linear combinations of the features of F^n.

Perceptrons are capable of projecting in higher dimensional feature transformations (that means we intermittently use modified/complex) features that consist of some modification to the inputs.

Feature transformation is designed to help reduce the impact of the curse of dimensionality.

The ad-hoc retrieval problem is defined by a database from which we try to obtain relevant material. The problem is then to retrieve the relevant subset of the database to a given query. There is no sequenced access pattern and no a-priori knowledge of the querying pattern. No purposeful caching can be doe

Using words as features is useful as they are good indicators of meanings, however words dubious in meaning via having multiple meanings for the same word and multiple words having the same meanings. Polysemy causes false positives and synonomy causes false negatives.

PCA is a correlation algorithm that maximizes variance while doing its reconstruction of features.

6.1 ICA: Independent Component Analysis

ICA tries to maximize independence, where the new features are maximized to be as independent as possible.

We try to minimize $I(y_i, y_j) = 0$ where I is the mutual information. We also want to maximize $I(X, Y)$ where X and Y are the feature sets where ICA$(X) = Y$.

ICA makes the assumption that there are a number of hidden variables that are independent of each other.

Blind-Source Separation problem, also known as the cocktail party problem is a potential application of ICA. It is the problem of filtering out individual sources of the sound.

ICA assumes linear combinations of the input features.

Create a matrix with samples of all the input features. Every row represents a feature and every column represents a sample.

ICA could be done for the ad-hoc retrieval problem.

Mutual information is a measure of how much one variable tells you about another variable.

PCA tends to find uncorrelated dimensions, and CAN find mutually independent feature sets.

By the central limit theorem, any linear combination of independent features results in a Gaussian. So using maximal variance when we assume there are hidden variables is bad as it would corrupt that information.

ICA does not concerned with ordering. There is no direct notion of ordering in ICA. However the use and modeling of kurtosis can be used potentially to help.

The fundamental assumptions of ICA and PCA are different.

ICA is directional in that it requires the features be rows and samples be columns.

ICA for faces finds actual facial features, whereas PCA finds eigenfaces.

ICA for natural scenes finds edges.

6.2 Alternatives

Random Components Analysis (RCA) also known as random projection. Generates random directions. The projection matrix P^T consists of randomly generated linear components. Works surprisingly effectively for pre-modification for classification problems. Simply removing a large number of dimensions is effective because we wind up with a large sample over a small dataset and we still manage to find some correlations. Mainly RCA helps combat the curse of dimensionality.

Perceptrons can project into a large number of dimensions such as in the case of XOR. RCA is computationally inexpensive.

Good quote 'You must earn your complexity'.

Linear discriminant analysis (LDA) finds a projection that discriminates based on the label. Finds linear separators between several clusters of points. The value of that projection represents the data. LDA is concerned with how the data will be used, i.e. that is it is concerned with the labels.

Another method is Latent Dirichlet Allocation (LDA) newer method.

ICA can be used to find data structure.

ICA is a probabilistic modeling technique whereas PCA is a linear algebra modeling technique.

Sometimes ICA fails and is more expensive.

Semi-supervised learning.