Machine Learning
Lecture 9

Lecture by Dr. Shai Fine
Typeset by Steven Karas

2017-12-24
Last edited 23:12:21 2017-12-24

Disclaimer These lecture notes are based on the lecture for the course Machine Learning, taught by Dr. Shai Fine at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Shai Fine.

Homework 4 Homework 4 will be published this week.

Agenda

- Unsupervised Learning
- EM - again

1 Unsupervized Learning

Unsupervised learning is not given target values (or labels).

Applied for data compression (vector quantization), structure discovery, dimensionality reduction, recommendation engines, and many more.

2 Clustering

Clustering is a poorly defined problem. Formally, partitions the feature space into subsets of clusters. A cluster is a volume of high-density samples separated from other clusters by a relatively low density volume.

There is no perfect method to cluster, as it is purely subjective.

Clustering can also be considered as building a weighted graph between groups of samples, where clusters are cliques.

Soft clustering provides a probability for each instance belonging to a given cluster.

Connection to Dimensionality Reduction Dimensionality reduction is producing a soft partition of the features, whereas clustering produces a soft partition of the instances.

2.1 Tree using similarity metric

El-Yaniv, Fine, Tishby, "Agnostic classification of Markovian sequences", NIPS, 1997 presented an approach that produced an ontology tree using LZ pairwise compression.

2.2 k-Means

Given the number of clusters k to fit, and a set of instances $x \in \mathcal{X}$. Initialize m_i for $i \in \{1, \dots, k\}$ to k random x_t. Denote the cluster membership of an instance x_t in cluster i as b_i^t.

In each round of the algorithm:

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } ||x_t - m_i|| = \min_j ||x_t - m_j|| \\ 0 & \text{else} \end{cases}$$

$$\forall i \in \{1, \ldots, k\} \quad m_i \leftarrow \frac{\sum_t b_i^t x_t}{\sum_t b_i^t}$$

Until m_i converges.

Objective We want to minimize the reconstruction error (total sum squared distance to the cluter centroid):

$$E(\{m_i\}_{i=1}^k \mid \mathcal{X}) = \sum_t \sum_i b_i^t ||x_t - m_i||^2$$

Take the derivative and set to zero:

$$m_i = \frac{\sum_t b_i^t x_t}{\sum_t b_i^t}$$

Note that each step is optimal, and the distance vectors are convex, but the cluster membership itself is nonconvex, and therefore we converge on a local optimum, but the overall problem is NP-Hard.

3 Expectation Maximization

Soft version of k-means.

Assumes a probability model $\Pr(c_j \mid x)$ for the distribution of each cluster c_j for a given example x. We introduce latent variables that indicate the source distribution, which we estimate from the data and the current estimation of distribution parameters, using the current values of the latent variables to refine parameter estimation.

Formally Log likelihood for a mixture model:

$$\mathcal{L}(X \mid \Theta) = \log \prod_i \Pr(x_i \mid \Theta) = \sum_i \log \sum_{j=1}^k \Pr(x_i \mid C_j) \Pr(C_j)$$

We assume latent variables Z, which make the optimization simpler.

Denote the complete likelihood as $\mathcal{L}_C = (X, Z \mid \Theta)$ in terms of both X and Z. Denote the incomplete likelihood as $\mathcal{L} = (X \mid \Theta)$ in terms of X.

Because we don't know the value of Z, we cannot compute $\mathcal{L}_C(X, Z \mid \Theta)$. But we can compute its conditional expected value given X and an old Θ_t:

$$\mathcal{Q}(\Theta; \Theta_t) = E_Z[\mathcal{L}_C(X, Z \mid \Theta) \mid X, \Theta_t] = \sum_Z \Pr(Z \mid X, \Theta_t) \log \Pr(X, Z \mid \Theta)$$

Algorithm Requires an initial guess Θ_0.

In the E-step, we compute the posterior probability $\Pr(Z \mid X, \Theta_t)$ using the current estimates:

$$\mathcal{Q}(\Theta; \Theta_t) = E[\mathcal{L}_C(X, Z \mid \Theta) \mid X, \Theta_t]$$

In the M-step, we update the parameter estimates to get Θ_{t+1} by maximizing $\mathcal{Q}(\Theta; \Theta_t)$:

$$\Theta_{t+1} = \arg\max_\Theta \mathcal{Q}(\Theta; \Theta_t)$$

Each step is gauranteed to increase the log-likelihood of the observed data $\log \Pr(X \mid \Theta)$ until a local maximum is reached, as an increase in $\mathcal{Q}$ increases the incomplete likelihood.

$$\mathcal{L}(X \mid \Theta_t) \geq \mathcal{Q}(\Theta \mid \Theta_t)$$

We use the log-likelihood because it's monotonically increasing, and as a log, it turns multiplication into addition.

4 Mixture Models

4.1 Gaussian Mixure Models

Let $\Pr(x \mid \mu_j, \Sigma_j)$ denote the Gaussian probability density function:

$$\Pr(x \mid \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x - \mu_j)^{\intercal} \Sigma_j^{-1} (x - \mu_j) \right]$$

Let Z be a random variable corresponding to the component identity. Denote $\alpha_j = \Pr(z = j)$ as the probability that the jth component is selected.

The probability of generating x by GMM is:

$$\begin{aligned} \Pr(x) &= \sum_z \Pr(x, z) \\ &= \sum_{j=1}^{k} \Pr(z = j) \Pr(x \mid z = j) \\ &= \sum_{j=1}^{k} \alpha_j \Pr(x \mid \mu_j, \Sigma_j) \end{aligned}$$

4.1.1 Derivation of EM for GMM

We want to fit a gaussian mixture of k components to a training set $\{x_1, \dots, x_n\}$. Compose the training data with the component labels $z_1, \dots, z_n$ to get pairs $x_i, z_i;\ i \in \{1, \dots, n\}$. The parameter vector is $\Theta = \alpha_1, \dots, \alpha_k, \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k$.

Let the log likelihood be:

$$\mathcal{L} = \sum_i \log \sum_{j=1}^{k} \alpha_j \Pr(x \mid \mu_j, \Sigma_j)$$

The parameters can be found by maximizing $\mathcal{L}$, but this does not have a closed form solution, so we fall back on the EM algorithm.

E-step The E-step computes the posterior probability of the missing data $z_1, \dots, z_n$:

$$\Pr(z_i = j \mid x_i, \Theta) = \frac{\alpha_j \Pr(x \mid \mu_j, \Sigma_j)}{\Sigma_{j'} \alpha_{j'} \Pr(x \mid \mu_{j'}, \Sigma_{j'})}$$

Denote $r_{ij} = \Pr(z_i = j \mid x_i, \Theta)$.

Compute $\mathcal{Q}(\Theta; \Theta_t)$:

$$\begin{aligned} \mathcal{Q}(\Theta; \Theta_t) &= \sum_i \sum_{j=1}^{k} \overbrace{\Pr(z_i = j \mid x_i, \Theta)}^{\text{weight of } z} \overbrace{\log \Pr(x_i, z_i = j \mid \Theta)}^{\text{complete log likelihood}} \\ &= \sum_i \sum_{j=1}^{k} r_{ij} [\log \Pr(z_i = j \mid \Theta) + \log \Pr(x_i \mid z_i)] \\ &\dots \\ &= \sum_i \sum_{j=1}^{k} r_{ij} \log \alpha_j - \frac{1}{2} \sum_{j=1}^{k} \log |\Sigma_j| \sum_i r_{ij} - \frac{1}{2} \sum_i \sum_{j=1}^{k} r_{ij} (x_i - \mu_j)^{\intercal} \Sigma_j^{-1} (x_i - \mu_j) + \text{Constant terms} \end{aligned}$$

M-step To maximize $\mathcal{Q}(\Theta; \Theta_t)$ with respect to μ_j, set the gradient to zero:

$$\frac{\partial}{\partial \mu_j} \mathcal{Q}(\Theta; \Theta_t) = \sum_{i=1}^{n} r_{ij} \Sigma_j^{-1} (x_i - \mu_j) = 0$$

$$\widehat{\mu_j} = \frac{\sum_{i=1}^{n} r_{ij} x_i}{\sum_{i=1}^{n} r_{ij}}$$

Where $r_{ij} = \Pr(z_i = j \mid x_i, \Theta_t)$ are computed based on the parameters at the t-th iteration.

Similarly, we have:

$$\widehat{\Sigma_j} = \frac{\sum_{i=1}^n r_{ij}(x_i - \mu_j)(x_i - \mu_j)^\intercal}{\sum_{i=1}^n r_{ij}}$$

If all the $r_{ij} \in \{0, 1\}$, then the component labels are known and the update expressions reduce to the standard parameter estimation for the mean and covariance matrix.

Update terms for M-step Maximize $\mathcal{Q}$ with respect to α_j, subject to $\sum_j \alpha_j = 1$.

Introduce Lagrange multiplier λ and set gradient to zero:

$$\frac{\partial}{\partial \alpha_j} \mathcal{Q}(\Theta; \Theta_t) + \lambda(1 - \sum_j \alpha_j) = \frac{\sum_i r_{ij}^{(t)}}{\alpha_j} - \lambda = 0$$

$$\sum_i r_{ij}^{(t)} = \lambda \alpha_j$$

Summing both sides with respect to j and plug in the constraint:

$$\sum_j \sum_i r_{ij}^{(t)} = \lambda \sum_j \alpha_j \Rightarrow \sum_j \sum_i r_{ij}^{(t)} = \lambda \Rightarrow n = \lambda$$

Substitute λ back in the gradient to get the update term:

$$\widehat{\alpha_j} = \frac{\sum_{i=1}^n r_{ij}^{(t)}}{n}$$

Full algorithm [1]

1. Let $\Theta_0 = (\alpha_1, \ldots, \alpha_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k)$ be the initial guess.
2. Set $t = 0$.
3. While the log likelihood $\mathcal{L} = \log \Pr(x_i \mid \Theta_t)$ is increasing:
   (a) Perform the E-step by computing $r_{ij} = \Pr(z_i = j \mid x_i, \Theta_t)$
   (b) Perform the M-step by re-estimating the parameters using the following update equation:

$$\widehat{\alpha_j} = \frac{\sum_{i=1}^n r_{ij}}{n} \qquad \widehat{\mu_j} = \frac{\sum_{i=1}^n r_{ij} x_i}{\sum_{i=1}^n r_{ij}}$$

$$\widehat{\Sigma_j} = \frac{\sum_{i=1}^n r_{ij}(x_i - \mu_j)(x_i - \mu_j)^\intercal}{\sum_{i=1}^n r_{ij}}$$

   (c) Let $\Theta_{t+1} = (\widehat{\alpha_1}, \ldots, \widehat{\alpha_k}, \widehat{\mu_1}, \ldots, \widehat{\mu_k}, \widehat{\Sigma_1}, \ldots, \widehat{\Sigma_k})$.
   (d) Set $t = t + 1$.

5 Spectral Clustering

Methods such as k-means look for compact clustering structures. Spectral clustering[2] provides a partition of the similarity graph that finds the min-cut.

Applications include object/region detection in images.

Details can be found on the backup slides.

6 Hierarchical Clustering

Generates a tree of clusters. Top down approach recursively splits clusters. Bottom up combines clusters to form larger groups.

[1] On the exam, we will need to derive EM for a non-gaussian distribution

[2] Shai mentioned this will not be on the exam

7 Measuring Cluster Validity

Domain knowledge is king.

7.1 Internal Index

Internal index measures the clustering goodness based on the clustered data.

7.1.1 Dunn Index

Identifies sets of clusters that are compact (small variance) and well separated (centroids are far apart):

$$\frac{\min \delta(C_i, C_j)}{\max \Delta_k}$$

Where $\delta(C_i, C_j)$ is a distance metric between clusters C_i, C_j; Δ_k is the distance within a cluster k (the cluster diameter).

A higher Dunn Index indicates better clustering.

7.1.2 Davies-Bouldin Index

$$\frac{1}{k}\sum_i \max_{i \neq j} \frac{\sigma_i + \sigma_j}{\delta(C_i, C_j)}$$

Similar to the Dunn Index, but both within and between distances are measured w.r.t. the cluster centroids.

A lower value indicates better clustering.

7.1.3 Silhouette Index

$$-1 \leq S(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}} \leq 1$$

where $a(i)$ is the average dissimilarity of x_i with all other instances within the same cluster; and $b(i)$ is the average dissimilarity of x_i with all other instances of the less dissimilar cluster.

$S(i) \to 1$ indicates the instance belongs to its cluster, whereas $S(i) \to -1$ indicates it matches better to the "competing" cluster.

7.2 External Index

External index measures the clustering goodness against externally given class labels.

7.2.1 Error rates

As in supervised learning

7.2.2 F measure

Harmonic mean of precision and recall:

$$F = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2\frac{TP}{2TP + FP + FN}$$

7.2.3 Jaccard measure

Measures the similarity between sample sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

7.2.4 Purity

Each cluster is assigned to the most frequent label y.

$$\text{Purity}(C) = \frac{1}{n} \sum_k \max_j |C_k \cap y_j|$$

This simply measures the number of correctly assigned labels divided by the dataset size.

7.3 Relative/Stability Index

Relative/Stability index measures two different clustering methods (or multiple runs of the same method).

7.3.1 Rand Index

Measures the normalized number of agreements between clustering runs:

$$R = \frac{u + v}{\binom{n}{2}}$$

where u is the number of pairs in the same cluster in both runs, and v is the number of pairs that differ.

8 Next week