EE 381V: Large Scale Learning **Spring 2013**

Lecture 11 — February 19

Lecturer: Caramanis & Sanghavi *Scribe: Ryan Buckley*

11.1 Overview

In this lecture we continue our discussion of spectral clustering of Gaussian mixtures, which has been the topic of the last two lectures. We then discuss a different way of doing dimensionality reduction, using the Johnson-Lindenstrauss lemma.

11.2 Review: Clustering Isotropic Gaussians

The last two lectures focused on the algorithms and analysis of spectral clustering for Gaussian mixtures in the isotropic case, in which the i-th Gaussian has the distribution $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2 I)$. Note that the covariance matrix is simply a multiple of the identity matrix, so each Gaussian is distributed spherically, as depicted in figure 11.1. Recall that in this case, we reduced dimension by projecting onto the subspace spanned by the top r components of the SVD.

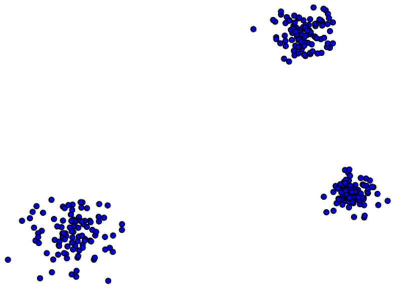

Figure 11.1. Three Isotropic Gaussians

11.3 Anisotropic Distributions

In this lecture we consider mixtures of Gaussians which are not spherically distributed. Instead, the distributions we consider will have a high variance orthogonal to the direction between the means of the distributions. Nevertheless, we will still require that the variance

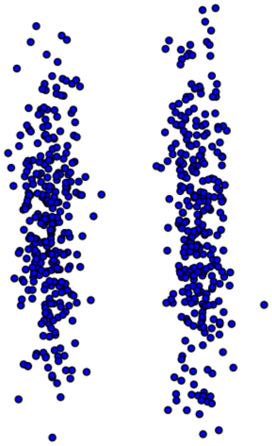

Figure 11.2. Parallel Pancakes: the distributions may contain directions of high variance, but the distance between their means is greater than the variance in that direction.

along the direction between the means be smaller than the distance between the means, and that for each cluster, there is a hyperplane which separates it from all the other clusters.

This problem, which we will call the "pancake problem," is still a Gaussian mixture model, but we will need a new algorithm in order to reduce the dimension. Suppose we attempted to reduce the dimension using the procedure developed for the isotropic case, in which we took the top r components of the SVD of the points in figure 11.2. This would be a disaster, since the top vectors of the SVD point in the directions of highest variance. In this case, that would mean projecting onto the vertical direction, which would project the two distributions onto each other.

We know that the best direction for projection is the direction between the means of the distributions. To find this direction, we use an algorithm by Brubaker and Vempala [1]. The algorithm as discussed here is simplified to assume that the direction of high variance is orthogonal to the direction between the means. The algorithm consists of five steps, listed in algorithm 1.

Algorithm 1 Algorithm for Pancake Problem

1: Isotropy: First we make the overall distribution isotropic (i.e. the covariance matrix is I). For this we need a linear map from $\mathbb{R}^n$ to $\mathbb{R}^n$ which makes the variance equal in all directions. We calculate the empirical covariances of sample points x_i, for which we define

$$\hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} = (x_i - \hat{\mu})(x_i - \hat{\mu})^T, \tag{11.1}$$

where the $\hat{\mu}$ is the sample mean: $\hat{\mu} = \frac{1}{N}\sum x_i$. We perform the mapping using the equation

$$y = \hat{\Sigma}^{-1/2}(x - \hat{\mu}). \tag{11.2}$$

Observe that with this mapping the points y_i are distributed according to a Gaussian with mean zero and covariance I.

2: Reweight: Give each point y a weight w according to

$$w(y) = e^{-\|y\|^2/\alpha}. \tag{11.3}$$

Note that points close to the origin will be given higher weight than points far from the origin.

3: Form the reweighted covariance matrix

$$\hat{M} = \frac{1}{N}\sum_{i=1}^{N} w(y_i) y_i y_i^T. \tag{11.4}$$

The best spectral direction will be the top singular vector of $\hat{M}$, which we denote by h.

4: Project all the points y_i onto h.

5: Use some algorithm to cluster the points now that dimensionality has been reduced.

11.3.1 Intuition

Figure 11.3 depicts the output of the transformation in step 1, the isotropy step. We apply a linear transformation and the output is another Gaussian mixture model, but with the same variance in every direction. Note that this uniformity of variance makes the SVD useless.

Note that if our input data is consists of highly separated, very tight clusters (i.e. very good data for clustering), the isotropy step will still result in distributions that look roughly like those in figure 11.3, and can actually make the data worse. Fortunately, the transformed data is still good enough to cluster.

Figure 11.4 shows the result of the weighting step. After weighting is applied, the variance is concentrated along the direction between the distributions.

Since the horizontal direction has the most weight along it, the algorithm will therefore select it as the best direction, and will project along this direction.

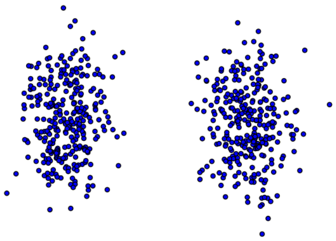

Figure 11.3. Points output by the algorithm's isotropy step

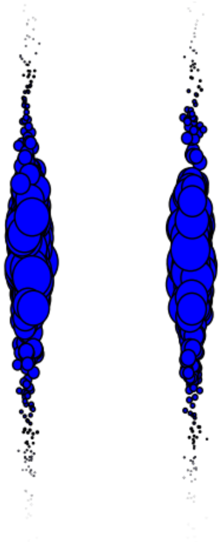

Figure 11.4. Weighted Pancakes: Points close to the origin have higher weight, so there is effectively more variance along the direction between the distributions

11.3.2 Analysis of Algorithm

We begin the analysis of the algorithm with a small caveat. In step 1 of the algorithm, we said that the Gaussians underwent a linear mapping. In fact, the mapping is not quite linear since it depends on the points that are being mapped. To illustrate why this map is not linear, we consider a one-dimensional example. Let X be a random variable with $X \sim \mathcal{N}(0, \sigma^2)$. If we define a $Y = \alpha X$ where α is a constant, then clearly Y is a Gaussian; its distribution is $\mathcal{N}(0, \alpha^2\sigma^2)$. Now suppose the scaling factor α is a function of a realization of X, and $y = \alpha(x)x$. This mapping is no longer linear, so Y will not be a Gaussian.

A formal analysis of this algorithm depends on the mapping being linear, so for such an analysis one can choose a few sample points, construct the linear map, and apply the map to the other points. In practice, the algorithm works well even without a truly linear map.

The following theorem shows that the algorithm correctly partitions the space, with high probability.

Theorem 1. *Let two Gaussians be distributed according to $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$. If there exists a direction v such that*

$$\|proj_v(\mu_1 - \mu_2)\| \geq c\left(\sqrt{v^T\Sigma_1 v} + \sqrt{v^T\Sigma_2 v}\right)\log\left(\frac{1}{\delta} + \frac{1}{n}\right),$$

then with probability greater than $1-\delta$, the algorithm partitions the space with errors in $\leq \eta$ fraction of points, using a number of samples N such that $N > d\log d\log 1/\delta$.

If there are more than two clusters, then we apply the algorithm recursively. Of course, we still require that for each cluster, some hyperplane separates that cluster from all the others.

One surprising result of this algorithm is that huge amounts of noise have no effect on clustering, provided the noise does not prevent us from separating the clusters with hyperplanes.

11.4 Johnson-Lindenstrauss Lemma

For the second part of the lecture, we turn to the Johnson-Lindenstrauss lemma.

In previous lectures we saw that dimensionality reduction via random projection did not work well for spectral clustering on Gaussian mixture models. There are many applications, however, where random projection is effective. For those cases, the result stated in the Johnson-Lindenstrauss lemma allows us to do dimensionality reduction without losing structure. Specifically, the lemma says that there is a random projection onto a space of lower dimension which preserves all pairwise distances simultaneously.

Lemma 1. *There exists a map $f : \mathbb{R}^d \to \mathbb{R}^k$ where $k \geq \frac{4\log n}{\epsilon^2/2 - \epsilon^2/3}$ such that*

$$(1-\epsilon)\|u-v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$$

Somewhat remarkably, nothing in this lemma depends on the ambient space d. In addition to the lower bound on k provided in the lemma, $k \in O(\log n/\epsilon^2)$, it is known that $k \in o(\log n/\epsilon^2)$ does not suffice. That is, this lemma and its simple proof provide a fairly tight bound on k [2].

The proof of this lemma consists of two major steps. First we show that with high probability, the squared length of a random vector will be tightly concentrated around its mean when projected into a random k-dimensional subspace. Second, we do a union bound over all pairs.

Part 1: The length of a fixed unit vector projected onto a random subspace has the same distribution as the length of a uniform random unit vector projected onto a fixed subspace. We form $X_i \sim \mathcal{N}(0,1)$ and $X = (X_1, X_2, \ldots, X_d)$. Then $Y = \frac{X}{\|X\|}$, so Y is a uniform random unit vector. We can project onto any k-dimensional subspace we choose, so we choose the subspace spanned by the first k components of Y. Define $L = |y_1|^2 + |y_2|^2 + \ldots + |y_k|^2$. Then $\mathbb{E}[L] = k/d$.

Lemma 2. *Let $k < d$. Then*

- *If $B < 1$, then*

$$Pr\left[L \leq \frac{\beta k}{d}\right] \leq \beta^{k/2}\left(1 + \frac{(1-\beta)k}{d-k}\right)^{(d-k)/2} \leq \exp\left(\frac{k}{2}(1-\beta+\ln\beta)\right) \tag{11.5}$$

- *If $B > 1$, then*

$$Pr\left[L \geq \frac{\beta k}{d}\right] \leq \beta^{k/2}\left(1 + \frac{(1-\beta)k}{d-k}\right)^{(d-k)/2} \leq \exp\left(\frac{k}{2}(1-\beta+\ln\beta)\right) \tag{11.6}$$

To use the lemma we fix any pair v_i, v_j and let their projections onto $\mathbb{R}^k$ be v_i', v_j', respectively. Then the probability of the

$$\begin{aligned}
\Pr\left[\|v_i' - v_j'\|^2 \leq (1-\epsilon)(k/d)\|v_i - v_j\|^2\right] &\leq \exp\left(\frac{k}{2}\left(1-(1-\epsilon)+\ln\left(1-\epsilon\right)\right)\right) && (11.7)\\
&\leq \exp\left(-\frac{k\epsilon^2}{4}\right) && (11.8)\\
&\leq \exp\left(-2\ln n\right) && (11.9)\\
&= 1/n^2. && (11.10)
\end{aligned}$$

Similarly,

$$\Pr\left[\|v_i' - v_j'\|^2 \geq (1+\epsilon)(k/d)\|v_i - v_j\|^2\right] \leq 1/n^2. \tag{11.11}$$

Therefore, the probability of a bad event for two points v_i, v_j is less than or equal to $2/n^2$. By taking a union bound over all points, the probability of having a bad pair is less than $\binom{n}{2}2/n^2 = 1 - 1/n$. So the probability of success is $1/n$. If we repeat the projection $O(n)$ times, we can increase the probability of success to the desired constant.

Bibliography

[1] S. Charles Brubaker and Santosh S. Vempala, "Isotropic PCA and Affine-Invariant Clustering." *49th Annual IEEE Symposium on Foundations of Computer Science,* 2008.

[2] Sanjoy Dasgupta and Anupam Gupta, "An Elementary Proof of a Theorem of Johnson and Lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2002.