CS281B/Stat241B: Advanced Topics in Learning & Decision Making

More Spectral Clustering and an Introduction to Conjugacy

Lecturer: Michael I. Jordan *Scribe: Marco Barreno*

Monday, April 5, 2004.

1 Back to spectral clustering

Last time we looked at a couple of algorithms for spectral clustering. At the end, we saw an approximation that relaxed the constraint that the indicators (the E matrix) be 0-1-valued and let E be a real-valued matrix. This made the optimization tractable at the cost of muddying the cluster assignments: E no longer directly represents cluster membership. To salvage the assignments, we need to perform *rounding*: we transform U, the matrix of eigenvectors for the relaxed problem, from a continuous matrix to a discrete matrix. But how?

1.1 One approach to rounding

Recall that the columns of the matrix U are the eigenvectors of the relaxed problem from last lecture, and solutions (to the relaxed problem) are of the form $Y' = UB$ for any arbitrary rotation matrix B. We need to find a piecewise constant (discrete) matrix as close to one of these solutions Y' as possible. The requirement that $D^{-1/2}Y = E\Lambda$ for some matrix Λ is equivalent to specifying that Y must be piecewise constant, so we will look for Y that satisfy this requirement that are as close as possible to a solution Y' of the relaxed problem. Remember that e_r is the indicator vector for cluster A_r, and E is the matrix of vectors e_r.

Because of the arbitrary rotation term, it is convenient to consider the subspaces spanned by the matrices. To compare subspaces, we use the CS transform, which is a notion of finding the best vectors in the subspaces between which to take the angle. In practice, this means that we compare projection operators in the Frobenius norm. If we let

$$\Pi_0 = \sum_{r=1}^{R} \frac{D^{1/2} e_r e_r^\top D^{1/2}}{e_r^\top D e_r}, \tag{1}$$

then

$$\begin{aligned} J(W,e) &= \frac{1}{2} \left\| UU^\top - \Pi_0 \right\|_F & (2)\\ &= 1/2 \text{ tr } UU^\top UU^\top + 1/2 \text{ tr } \Pi_0^2 - \text{ tr } UU^\top \Pi_0 & (3)\\ &= R - \sum_{r=1}^{R} \frac{e_r^\top D^{1/2} UU^\top D^{1/2} e_r}{e_r^\top D e_r}, & (4) \end{aligned}$$

where Equation 4 follows because we can rotate the order of terms inside the trace function (and therefore the first two terms turn out to be $R/2$ each—remember that R is the number of eigenvectors in U).

At this point, we can either fix E (that is, choose a partition with indicators $(e_1, \ldots, e_R)$) and learn the affinity matrix, or we can fix W and "round" to get E.

The following theorem provides a variational representation for our cost function.

Theorem 1

$$J(W,e) = \min_{(\mu_1,\ldots,\mu_R)\in\mathbb{R}^{R\times R}} \sum_r \sum_{p\in A_r} d_p \|u_p d_p^{-1/2} - \mu_r\|^2 \tag{5}$$

Proof: This proof is reproduced from the Bach and Jordan paper “Learning Spectral Clustering” [1] available on the course website.

Let $D(\mu, A) = \sum_r \sum_{p\in A_r} d_p \|u_p d_p^{-1/2} - \mu_r\|^2$. Minimizing $D(\mu, A)$ with respect to μ is an unconstrained least-squares problem and we get:

$$\begin{aligned}
\min_u D(\mu, A) &= \sum_r \sum_{p\in A_r} u_p^\top u_p - \sum_r \frac{\left\|\sum_{p\in A_r} d_p^{1/2} u_p\right\|^2}{\sum_{p\in A_r} d_p} &(6)\\
&= \sum_p u_p^\top u_p - \sum_r \frac{\sum_{p,p'\in A_r} d_p^{1/2} d_{p'}^{1/2} u_p^\top u_{p'}}{e_r^\top D e_r} &(7)\\
&= R - \sum_r \frac{e_r^\top D^{1/2} U U^\top D^{1/2} e_r}{e_r^\top D e_r} &(8)\\
&= J(W,e) &(9)
\end{aligned}$$

■

1.2 A new spectral clustering algorithm

The representation in Theorem 1 suggests treating the minimization of the cost function as a weighted k-means problem. We now see a new algorithm for spectral clustering:

- Compute R eigenvectors of $D^{-1/2} W D^{-1/2}$, where $D = \mathrm{diag}(W\mathbf{1})$
- Let $U = (u_1, \ldots, u_R)$ and $d_p = D_{pp}$.
- Weighted k-means:
  - For all r, $\mu_r = \frac{\sum_{p\in A_r} d_p^{1/2} u_p}{\sum_{p\in A_r} d_p}$
  - For all p, assign p to A_r with $r = \arg\min_{r'} \|u_p d_p^{-1/2} - \mu_r\|^2$
  - Repeat until the partitioning into A_r is stationary.
- Output $A = (A_r)$

The k-means procedure changes both the μ_r and the e_r; optimizing over μ_r makes the variational statement equal to the definition of $J(W,e)$ as in the theorem, while optimizing over E finds the right partition. This algorithm doesn't explicitly normalize, but the weights sort of normalize to the unit sphere.

To initialize the k-means algorithm, we want vectors that are roughly orthogonal to each other. To achieve this, we go to the U matrix and pick the first row, then pick another one about 90° away, then again, until we have R $(= k)$ roughly orthogonal vectors to initialize k-means.

1.3 Cluster kernels

We briefly mentioned cluster kernels, which are basically kernels created by spectral clustering so that $\kappa(x, y)$ has a high value if x and y are in the same cluster and a low value otherwise. These can be used in *semi-supervised learning*, the very common case in which some data points are labeled but many are not. For more on this topic, see the paper "Cluster Kernels for Semi-Supervised Learning" [2] available on the course website.

2 Bayesian Methods and Conjugacy

Now we're going to move in another direction and talk about Bayesian things. We eventually will get to Dirichlet processes (in a future lecture), but first we need to explore some background. The focus of the rest of today's lecture will be *conjugacy*.

2.1 Quick Bayesian background

Recall that in the Bayesian framework, we treat parameters as random variables and endow them with distributions. We often talk about the following probabilities concerning variables y and parameters θ:

- likelihood: $p(y|\theta)$
- prior: $p(\theta)$
- posterior: $p(\theta|y)$

Bayes' rule gives us the relationship

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}, \tag{10}$$

which is often used when $p(y|\theta)$ is easier to compute than $p(\theta|y)$. We can compute the marginal probability of y as

$$p(y) = \int d\theta\; p(\theta)\, p(y|\theta). \tag{11}$$

2.2 Basics of conjugacy

The first two distributions we will look at in the context of conjugacy are the binomial and beta distributions. The beta distribution is a conjugate prior for the binomial distribution. This means that if the parameter θ for the binomial distribution is endowed with a beta prior, the posterior will also be beta-distributed. A graphical model of the situation, including y, θ, and the hyperparameters α and β is shown in Figure 1.

The binomial distribution gives the probability that y of n Bernoulli trials (think coin flips) will be a one (or heads) when the probability of each trial resulting in a one is θ. For $y \in \{0, 1, \ldots, n\}$ it is given by

$$p(y|\theta) = \binom{n}{y} \theta^y (1-\theta)^{n-y}. \tag{12}$$

Note that the random variable y is "upstairs" in the exponent. The beta distribution for $\theta \in [0, 1]$ is

$$p(\theta|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1}. \tag{13}$$

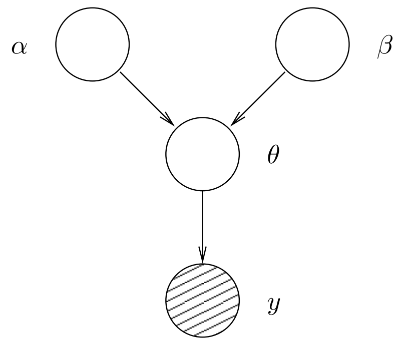

Figure 1: The variable $y \in \{0, 1, \ldots, n\}$ is distributed as $p(y|\theta) = \text{Bin}(n, \theta) = \binom{n}{y}\theta^y(1-\theta)^{n-y}$. We endow $\theta \in [0, 1]$ with the distribution $p(\theta|\alpha, \beta) = \text{Beta}(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$. The hyperparameters α and β are shown in the model.

Here the random variable θ is "downstairs" in the base of the exponentiation. The parameters α and β here are called *hyperparameters* because they are parameters in a distribution for another parameter θ. Beta distributions for three different settings of α and β are shown in Figure 2. Note that the -1 in the exponents is a convention that leads to a nice form for the mean:

$$E(\theta) = \frac{\alpha}{\alpha + \beta}. \tag{14}$$

We now show that conjugacy holds; that is, we show that when θ has a beta-distributed prior, it also has a beta posterior:

$$\begin{aligned} y|\theta &\sim \text{Bin}(n, \theta) && (15)\\ \theta|\alpha, \beta &\sim \text{Beta}(\alpha, \beta) && (16)\\ p(\theta|y) &\propto p(\theta)\,p(y|\theta) && (17)\\ &\propto \theta^y(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} && (18)\\ &= \theta^{\alpha+y-1}(1-\theta)^{\beta+n-y-1} && (19) \end{aligned}$$

So $\theta|y \sim \text{Beta}(\alpha + y, \beta + n - y)$. In this case, we can think of the hyperparameters as specifying a prior sample size (α is the prior number of ones and β the prior number of zeros). The mean is:

$$\begin{aligned} E(\theta|y) &= \frac{\alpha + y}{\alpha + \beta + n} && (20)\\ &= \frac{\alpha+\beta}{\alpha+\beta+n} \cdot \underbrace{\frac{\alpha}{\alpha+\beta}}_{A} + \frac{n}{\alpha+\beta+n} \cdot \underbrace{\frac{y}{n}}_{B} && (21) \end{aligned}$$

We see that the posterior mean is a weighted combination of the prior mean (A) and the maximum likelihood solution (B).

2.3 Gaussian distributions

We will now present conjugate priors for three different cases of Gaussian distributions. In each case, let $y = (y_1, \ldots, y_n)$ be the n data points observed.

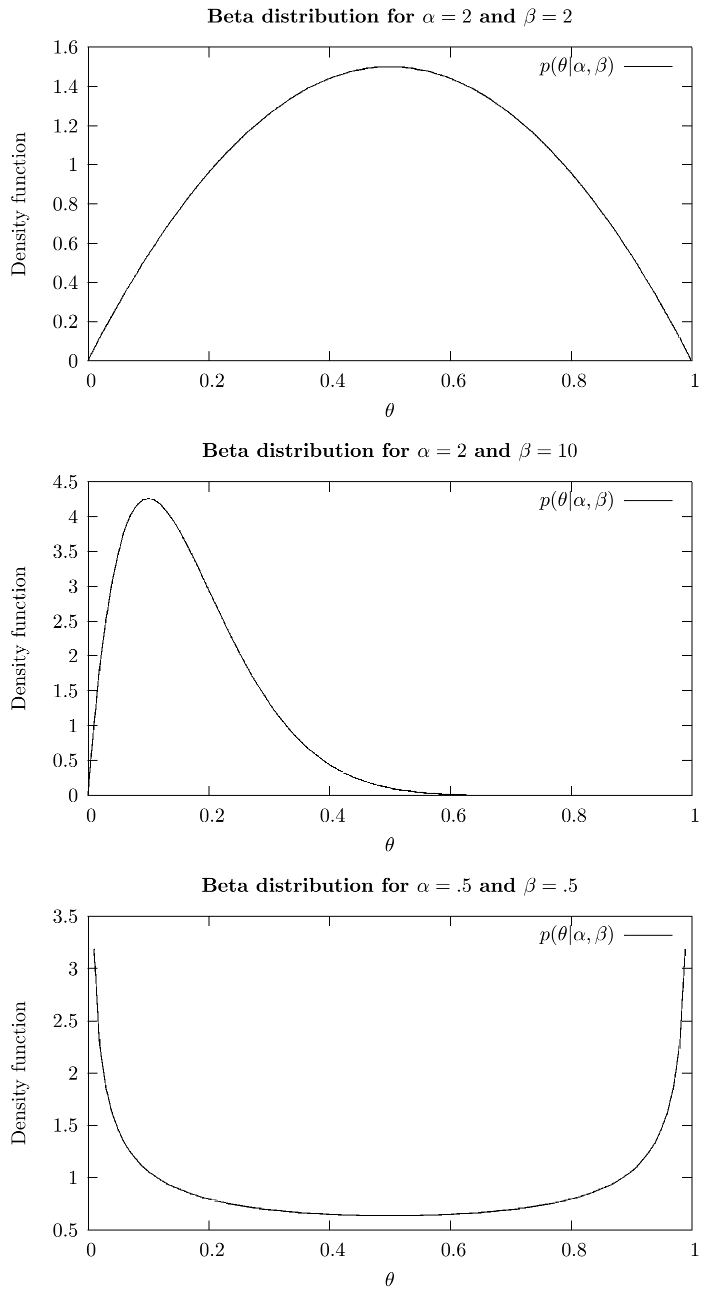

Figure 2: Beta distributions for various values of α and β. The distribution is unimodal for $\alpha > 1$ and $\beta > 1$ and symmetric when $\alpha = \beta$, as in the top graph. When $\alpha = \beta = 1$, the distribution is uniform (not shown). In the middle graph, we can see how the mean can be pushed one way or the other when $\alpha \neq \beta$. For $\alpha < 1$ and/or $\beta < 1$, as in the third graph, the probability mass is concentrated at one or both ends of the range for θ; this can be used to approximate a discrete distribution.

Univariate Gaussian, known σ^2

The Gaussian distribution is a conjugate prior in this case.

$$\begin{aligned}
p(y_i|\mu) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i-\mu)^2} && (22)\\
p(\mu) &\propto e^{-\frac{1}{2\tau_0^2}(\mu-\mu_0)^2} && (23)\\
p(\mu|y) &\propto e^{-\frac{1}{2\tau_1^2}(\mu-\mu_1)^2} && (24)
\end{aligned}$$

where

$$\begin{aligned}
\mu_1 &= \frac{1/\tau_0^2}{1/\tau_0^2 + n/\sigma^2}\mu_0 + \frac{n/\sigma^2}{1/\tau_0^2 + n/\sigma^2}\bar{y} && (25)\\
\frac{1}{\tau_1^2} &= \frac{1}{\tau_0^2} + \frac{1}{\sigma^2} && (26)
\end{aligned}$$

and $\bar{y} = \frac{1}{n}\sum_i y_i$.

Univariate Gaussian, known μ

The conjugate prior that we use here is the inverse gamma distribution.

$$\begin{aligned}
p(y|\sigma^2) &\propto (\sigma^2)^{-n/2} e^{-\frac{n}{2\sigma^2}\nu} && (27)\\
p(\sigma^2) &\propto (\sigma^2)^{-(\alpha+1)} e^{-\frac{\beta}{\sigma^2}} && (28)\\
p(\sigma^2|y) &\propto (\sigma^2)^{-(\alpha+\frac{n}{2}+1)} e^{-(\frac{\beta+\frac{n}{2}\nu}{\sigma^2})} && (29)
\end{aligned}$$

where $\nu = \sum_i (y_i - \mu)^2$.

Multivariate Gaussian

In this case we put priors on both μ and Σ. The conjugate prior for the mean is $\mu|\Sigma \sim N(\mu_0, \Sigma/\kappa_0)$, and for the covariance we use $\Sigma \sim \text{InverseWishart}(\nu_0, \Lambda_0^{-1})$. The posterior is given by

$$p(\mu, \Sigma|y) \sim N(\mu_n, \Sigma/\kappa_n) \cdot \text{InverseWishart}(\nu_n, \Lambda_n^{-1}), \qquad (30)$$

where

$$\begin{aligned}
\mu_n &= \frac{\kappa_0}{\kappa_0 + n}\mu_0 + \frac{n}{\kappa_0 + n}\bar{y} && (31)\\
\kappa_n &= \kappa_0 + n && (32)\\
\nu_n &= \nu_0 + n && (33)\\
\Lambda_n &= \Lambda_0 + S + \frac{\kappa_0 n}{\kappa_0 + n}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^\top && (34)\\
(S &= \sum_i (y_i - \bar{y})(y_i - \bar{y})^\top). && (35)
\end{aligned}$$

There are four free parameters: ν_0, Λ_0^{-1}, μ_0, and κ_0. Note that the prior for μ must be conditional on Σ in order to be conjugate because in the posterior μ and Σ are dependent.

References

[1] F. Bach and M. I. Jordan. Learning spectral clustering. In *Advances in Neural Information Processing (NIPS) 16*. MIT Press, 2004.

[2] O. Chapelle, J. Weston, and B. Schoelkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing (NIPS) 14*. MIT Press, 2002.