Small Variance Asymptotics for Non-parametric Bayesian Clustering

Abhishek Kumar
Roll No. 150035

Manish Kumar Bera
Roll No. 150381

Anubhav Mittal
Roll No. 150116

1 Introduction

Learning the correct model size for a specific data-set is one of the biggest challenges in machine learning model. For clustering problem we need to know the number of classes. For factor analysis we need to know number of factors[3].

One possible way to tackle this problem is to try out several models and use the one which gives best result on validation set. Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix the number of parameters upfront. Bayesian methods offer great flexibility, for example we can even infer hierarchical models.

Despite the huge success of Bayesian framework, simpler non-Bayesian methods such as k-means have been more popular for large scale data, due to their simplicity in implementation and high scalability. In this project, we study recent attempts to reach middle-ground using small variance asymptotics, so that we get a non parametric model which is scalable .

We start with a hard non-parametric clustering algorithm which uses Dirichlet Process. We then study an extension of this algorithm to a hierarchical structure using Hierarchical Dirichlet process. Finally, we generalize the clustering algorithm to use Bregman divergences instead of just euclidean distance.

2 Background

2.1 Dirichlet Process

For a distribution G to be distributed according to a Dirichlet Process, its marginal distributions have to be Dirichlet distributed[2]. Formal definition is provided below

Definition 2.1 G is Dirichlet Process distributed with base distribution H and concentration parameter α, written as $G \sim DP(\alpha, H)$ if $(G(A_1), \ldots, G(A_r)) \sim Dir(\alpha H(A_1), \ldots, \alpha H(A_r))$ for every finite measurable partition $A_1, \ldots, A_r$ of Θ which is support of H. [6]

Here H and α have intuitive roles to play. By using properties of Dirichlet distribution we can see that $\mathbb{E}[G(A)] = H(A)$ and $\mathbb{V}[G(A)] = \frac{H(A)(1-H(A))}{\alpha+1}$. Thus H is like mean distribution for this process and α can be treated like variance parameter.

Constructions like Blackwell-Mac-Queen urn scheme and Stick breaking process(that we saw in class) ensure existence of DP.

2.2 Posterior of DP

[6] Let $G \sim DP(\alpha, H)$ and $\theta_1, \ldots \theta_n$ be i.i.d. draws from G. Let $A_1, \ldots A_r$ be a finite measurable partition of Θ and let $n_k = \#\{i : \theta_i \in A_k\}$ be number of observations in A_k. Since Dirichlet and multinoulli are conjugate to each other, we get following posterior

$$(G(A_1), \ldots, G(A_r)|\theta_1, \ldots \theta_n) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$$

Simple algebra yields:

$$G|\theta_1, \dots \theta_n \sim DP(\alpha + n, \frac{\alpha}{\alpha+n}H + \frac{n}{\alpha+n}\frac{\sum_{i=1}^{n}\delta_{\theta_i}}{n})$$

The predictive distribution can be written as

$$\mathbb{P}(\theta_{n+1} \in A|\theta_1, \dots \theta_n) = \int \mathbb{P}(\theta_{n+1} \in A|G(A))\mathbb{P}(G(A)|\theta_1, \dots \theta_n)dG(A)$$

$$\theta_{n+1}|\theta_1, \dots, \theta_n \sim \frac{1}{\alpha+n}(\alpha H(A) + \sum_{i=1}^{n}\delta_{\theta_i}(A))$$

which is very intuitive. It says that next draw of θ will yield a value from base distribution with probability that is proportional to α and one of the old values with probability which is proportional to number of θ's already drawn of that value. This sequence of predictive distributions is called **Blackwell Mac-Queen Urn Scheme**.

2.3 Dirichlet Process Mixture Model

This can be seen as infinite dimensional generalization of Dirichlet distribution. For a DPMM, we define a generative story as follows: Each new incoming point chooses a cluster c with probability π_c, and then generates an observation from the Gaussian distribution corresponding to that cluster. The means of the clusters are drawn from some prior distribution G_0, and we fix the co-variance as $\sigma^2 I$. The above is summarized as follows:

$$\begin{aligned}
&\mu_1, ..., \mu_k \sim G_0 \\
&\pi \sim \text{Dir}(\frac{\alpha}{k}, \frac{\alpha}{k}, ...\frac{\alpha}{k}) \\
&z_1, ..., z_n \sim \text{Multinoulli}(\pi) \\
&x_1, ..., x_n \sim \mathcal{N}(\mu_{z_i}, \sigma^2 I)
\end{aligned}$$

We let $k \to \infty$ to get infinite mixture model.

2.4 Inference

We can use Gibbs sampling for inference in the model(Neal 2000). For each point x_i, we assign it to cluster c with probability $\frac{n_{-i,c}}{Z}.\mathcal{N}(x_i|\mu_c, \sigma^2 I)$ where $n_{-i,c}$ is number of points in cluster c excluding point x_i. With probability $\frac{\alpha}{Z}.\int \mathcal{N}(x_i|\mu, \sigma^2 I)dG_0(\mu)$, we start a new cluster. Z is the normalization constant. For this newly formed cluster, we compute the means using the prior G_0 and the point x_i which created this cluster. After assigning cluster to each point, we compute the means of all clusters using the points assigned to them and the prior. We repeat above steps in cyclic manner until convergence.

3 DP Means Hard clustering

We first define G_0 (the prior distribution over the means) as $\mathcal{N}(0, \rho I)$. Probability of starting new cluster : $\frac{\alpha}{Z}(2\pi(\rho + \sigma^2))^{-\frac{d}{2}} \cdot exp(-\frac{1}{2(\rho+\sigma^2)}||x_i||^2)$ Probability of getting assigned to cluster c is : $\frac{n_{-i,c}}{Z}(2\pi\sigma^2)^{-\frac{d}{2}} \cdot exp(-\frac{1}{2\sigma^2}||x_i - \mu_c||^2)$. Now we let $\sigma \to 0$. But if let α and σ independent of each other then each point will create a new cluster consisting of only this point. Clearly this trivial clustering is useless and we would like α to depend on σ. The authors of this[5] paper chose $\alpha = (1 + \frac{\rho}{\sigma})^{\frac{d}{2}} \cdot exp(-\frac{\lambda}{2\sigma})$. After simplification we obtain:

$$\text{Probability of starting new cluster } = \frac{exp(-\frac{\lambda}{2\sigma} - \frac{||\boldsymbol{x}_i||}{2(\rho+\sigma)})}{exp(-\frac{\lambda}{2\sigma} - \frac{||\boldsymbol{x}_i||}{2(\rho+\sigma)} + \sum_{j=1}^{K} n_{-i,j} exp(-\frac{1}{2\sigma}||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2)}$$

$$\text{Probability of getting assigned to cluster } c = \frac{exp(-\frac{\lambda}{2\sigma} - \frac{||\boldsymbol{x}_i||}{2(\rho+\sigma)})}{exp(-\frac{\lambda}{2\sigma} - \frac{||\boldsymbol{x}_i||}{2(\rho+\sigma)} + \sum_{j=1}^{K} n_{-i,j} exp(-\frac{1}{2\sigma}||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2)}$$

In hard clustering, we make $\sigma^2 \to 0$. The probabilities become binary as they will be dominated by $min(||\boldsymbol{x}_i - \boldsymbol{\mu}_1||^2, \ldots, ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2, \lambda)$ and resulting update turns out to be analogous to $k-means$ where we assign the point to closest mean. However one subtle difference is that if the distance to closest mean is is greater than $\lambda(\alpha)$, then the probabilities corresponding to each of the existing cluster falls to zero and we start a new cluster.

3.1 Underlying Objective function

The hard clustering algorithm minimizes the objective function:

$$min_{\{l_j\}_{j=1}^k} \sum_{c=1}^{k} \sum_{x \in l_c} ||x - \mu_c||^2 + \lambda k \quad (1)$$

$$\text{where } \mu_c = \frac{\sum_{x_i \in l_c} x_i}{|l_c|}$$

This is similar to the K-means algorithm, only value of k is not fixed and the objective penalizes large k. Here is the DP means algorithm

Input: $x_1, ..., x_n, \lambda$: cluster penalty parameter.
Output: Clustering of points in $l_1, ..., l_k$ and no. of cluster k.

1. Initialize $k = 1, l_1 = x_1, ..., x_n, \mu_1 = \frac{\sum x_i}{n}$ and $z_i = 1$ for each point.
2. Repeat until convergence:
   - For each point x_i,
     - Compute distance from all means i.e. $d_{ic} = ||x_i - \mu_c||^2$ for all c.
     - if $min_c d_{ic} > \lambda$, set $k = k + 1, z_i = k, \mu_k = x_i$.
     - Else, set $z_i = min_c d_{ic}$
   - Assign points x_i with $z_i = c$ to the cluster l_c.
   - For each cluster c, $\mu_c = \frac{\sum_{x_i \in l_c} x_i}{|l_c|}$.

Algorithm 1: Hard Clustering Algorithm[5]

Theorem 2.1 *Algorithm 1 monotonically decreases the objective given above until local convergence*.
Proof For all distances greater than λ, we generate a new cluster, assign our point to it, and pay a penalty of λ, thus we decrease our objective function. Hence, with every step, we update in such a way that our objective never increases. In the mean-assignment step, we also end up decreasing the function, since taking the mean of the points as the mean of the cluster decreases the sum of their euclidean distances from the mean, this follows from the simple k-means proof. Hence, as we decrease our objective in every step, and only a finite no. of possible clustering of data, we will always converge to a local optima.

4 Hierarchical Dirichlet process

Assume that we have J data-sets with each having n_j data-points. But instead of learning them inependently, we want to learn clusters over these data-sets jointly. In other words we want them to share parameters and be related to each other. HDP is a non parametric prior which allows mixture models to share components. Formally it can be defined as follows[7]:

$$G_0|\gamma, H \sim DP(\gamma, H) \quad G_j|\alpha, G_0 \sim DP(\alpha, G_0)$$
$$\phi_{ji}|G_j \sim G_j \qquad x_{ji}|\phi_{ji} \sim F(\phi_{ji})$$

where G_0 is global measure and G_j's are specific to data-sets. This allows mixture models to share components and it is non-parametric.
There is a metaphor called *Chinese Restaurant Franchise* that gives an alternative view of HDP.

4.1 Hierarchical Dirichlet process Mixture Models

Similar to previous case, we can achieve a hard clustering based algorithm by applying small variance asymptotics to HDP model. One can show that HDP minimizes following objective :

$$\min_{\{l_p\}_{p=1}^{g}} \sum_{p=1}^{g} \sum_{x_i j \in l_p} ||x_{ij} - \mu_p||^2 + \lambda_l k + \lambda_g g$$

where k, g is total number of local and global clusters respectively. λ_l, λ_g are regularization parameters, l_p is the set points assigned to cluster p and $\mu_p = \frac{1}{|l_p|} \sum_{x_{ij} \in l_p} x_{ij}$. The proof is very similar to the previous case hence we skip it. Here is the complete HDP algorithm :

Input: $\{x_{ij}\}$: input data, λ_l: local cluster penalty parameter, λ_g: global cluster penalty parameter,
Output: Global clustering $l_1, ..., l_g$ and no. of cluster k_j , for each data set j.

1. Initialize $g = 1, k_j = 1 \; \forall j$ and μ_1 to be global mean across all data sets. Initialize all local cluster indicators $z_{ij} = 1 \;\; \forall i, j$ and global cluster associations $v_{j1} = 1 \;\; \forall j$.
2. Repeat the following steps until convergence:
3. For each point x_{ij},
   - Compute $d_{ijp} = ||x_{ij} - \mu_p||^2$ for $p = 1, 2, ..., g$.
   - For all p such that $v_{jc} \neq p$ for all $c = 1, 2, ..., k_j$, set $d_{ijp} = d_{ijp} + \lambda_l$.
   - If $min_p d_{ic} > \lambda_l + \lambda_g$, set $k_j = k_j + 1, z_{ij} = k_j, g = g + 1, \mu_g = x_{ij}, v_{jk_j} = g$.
   - Else, let $\hat{p} = \arg\min_P d_{ijP}$
     -If $v_{jc} = \hat{p}$ for some c, set $z_{ij} = c$ and $v_{jc} = \hat{p}$
     -Else, set $k_j = k_j + 1, z_{ij} = k_j$, and $v_{jk_j} = \hat{p}$.
4. For all clusters:
   - Let $S_{jc} = \{x_{ij} | z_{ij} = c\}$ and $\mu'_{jc} = \frac{1}{|S_{jc}|} \sum_{x \in S_{jc}} x$.
   - Compute $d'_{jcp} = \sum_{x \in S_{jc}} ||x - \mu_p||^2$ for all $p = 1, 2, .., g$.
   - If $\min_p d'_{jcp} > \lambda_g + \sum_{x \in S_{jc}} ||x - \mu'_{jc}||^2$, set $g = g + 1, v_{jc} = g, \mu_g = \mu'_{jc}$.
   - Else set $v_{jc} = \arg\min_p d'_{icp}$.
5. For each global cluster $p = 1, 2, .., g$, re-compute means:
   - Let $l_p = \{x_{ij} | z_{ij} = c \text{ and } v_{jc} = p\}$.
   - Compute $\mu_p = \frac{1}{|l_p|} \sum_{x \in l_p} x$.

Algorithm 2: Hard Gaussian HDP[5]

5 Extending to Exponential Family

5.1 Background

An exponential family distribution has following pdf: $p(\boldsymbol{x}|\theta) = h(\boldsymbol{x}) exp(\langle \boldsymbol{x}, \theta \rangle - \psi(\theta))$. The best thing about exponential family distributions is that they have conjugate Prior : $p(\theta|\tau, \eta) = exp(\langle \theta, \tau \rangle - \eta\psi(\theta) - m(\tau, \eta))$. Posterior has same form as prior with $\tau = \tau + \boldsymbol{x}_i$ and $\eta = \eta + 1$

Definition 5.1 (Bregman, 1967) Let $\phi : S \to \mathbb{R}$ be a strictly convex function defined on convex set S such that ϕ is differentiable on interior of S. The bregman divergence is defined as $d_\phi = \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\phi(\boldsymbol{y}) \rangle$.

Squared euclidean distance is a bregman divergence with $\phi(x) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$. If $\boldsymbol{\pi}$ is probability vector, then negative entropy $\phi(\boldsymbol{\pi}) = \sum_{j=1}^{D} p_j \log p_j$ is a convex function. The corresponding bregman divergence is $d_\phi(\boldsymbol{\pi}, \boldsymbol{x}) = KL(\boldsymbol{\pi}||\boldsymbol{x})$

Definition 5.2 (Rockfellar 1970) Let ψ be a **proper**, **closed**, convex function with $\Theta = interior(domain(\psi))$. The pair (Θ, ψ) is called a convex function of legendre type if following are satisfied :

- Θ is nonempty
- ψ is strictly convex and differentiable on Θ
- $\forall \theta_b \in bd(\Theta), \lim_{\theta \to \theta_b} ||\nabla \psi(\theta)|| \to \infty, \theta \in \Theta$

Lemma 5.1(Barndoff 1978) Let ψ be the cumulant function of a regular exponential family with natural parameter space $\Theta = dom(\psi)$. Then (Θ, ψ) is a convex function of legendre type.

Definition 5.3(Rockfellar 1970) Let ψ be a real valued function on $\mathbb{R}^d$. Then its conjugate function ψ^* is given by $\psi^*(t) = sup\{\langle t, \theta \rangle - \psi(\theta)\}$ and if ψ is a proper closed convex function then ψ^* is also a proper closed convex function and $\psi^{**} = \psi$

If ψ is convex and differentiable then we can obtain unique θ^+ corresponding to supremum[1] by setting derivative to 0, $\theta^+ = \psi^{-1}(t)$. Then $\psi^*(t) = \langle t, \theta^+ \rangle - \psi(\theta^+)$. Following theorem establishes connection between ψ, ψ^* formally :

Theorem 5.1(Rockfellar) Let ψ be proper, closed strictly convex function with conjugate function ψ^*. Let $\Theta = int(dom(\psi))$ and $\Theta^* = int(dom(\psi^*))$. If (θ, ψ) is a convex function of legendre type then :

- (θ^*, ψ^*) is a convex function of legendre type.
- (θ^*, ψ^*) and (θ, ψ) are called legendre duals of each other.
- The gradient function $\nabla \psi$ is a one to one function from open convex set Θ onto the open convex set Θ^*.
- $\nabla \psi^* = (\nabla \psi)^{-1}$

Let $\boldsymbol{\mu}(\theta)$ denote expectation parameter of an exponential family $p_{\psi,\theta}$ We know that $\boldsymbol{\mu}(\theta) = \nabla \psi(\theta)$. Let us define ϕ as conjugate of ψ. Using theorem 5.1 and lemma 5.1, (Θ, ψ) and $(int(dom(\phi)), \phi)$ are legendre dual of each other. More importantly, $\nabla \psi^{-1}(\boldsymbol{\mu}) = \theta(\boldsymbol{\mu}) = \nabla \phi(\boldsymbol{\mu}) \quad (1) \implies \phi(\boldsymbol{\mu}) = \langle \theta(\boldsymbol{\mu}), \boldsymbol{\mu} \rangle - \psi(\theta(\boldsymbol{\mu})) \quad (eqn\ 5.1)$. This eqn gives us a way to express exponential family distribution in terms of its expectation parameter as we shall see in next section:

5.2 Relation with Exponential Family

Theorem 5.2 Let $p_{\psi,\theta}(\boldsymbol{x})$ be pdf of regular exponential family family distribution. Let ϕ be the conjugate of ψ. Let θ be natural parameter and $\boldsymbol{\mu}$ be expectation parameter. Let d_ϕ be the bregman divergence derived from ϕ. Then $p_{\psi,\theta}(\boldsymbol{x})$ can be uniquely expressed as $p_{\psi,\theta}(\boldsymbol{x}) = exp(-d_\phi(\boldsymbol{x}, \boldsymbol{\mu}))b_\phi(\boldsymbol{x})$ where $b_\phi(\boldsymbol{x}) = exp(\phi(\boldsymbol{x}))h(\boldsymbol{x})$

Proof

$$\begin{aligned} p_{\psi,\theta}(\boldsymbol{x}) &= h(x)exp(\langle \boldsymbol{x}, \theta \rangle - \psi(\theta)) \\ &= h(x)exp(\phi(\boldsymbol{\mu}) + \langle \boldsymbol{x} - \boldsymbol{\mu}, \nabla \phi(\boldsymbol{\mu}) \rangle) \\ &= h(x)exp(-(\phi(\boldsymbol{x}) - \phi(\boldsymbol{\mu}) - \langle \boldsymbol{x} - \boldsymbol{\mu}, \nabla \phi(\boldsymbol{\mu}) \rangle) + \phi(\boldsymbol{x})) \\ &= exp(-d_\phi(\boldsymbol{x}, \boldsymbol{\mu}))b_\phi(\boldsymbol{x}) \text{ where } b_\phi(\boldsymbol{x}) = exp(\phi(\boldsymbol{x}))h(\boldsymbol{x}) \end{aligned}$$

This theorem basically tells us that given a regular exponential family distribution, we can obtain a unique Bregman Divergence. The converse of this statement is also true is we put some restrictions on the function $\phi(\cdot)$ of Bregman divergence. We skip those restrictions as they were not our focus point. For our purpose, we assume that those restrictions are satisfied. Interested readers should refer to this [1] paper. We state following theorem without proof.

Theorem(Banerjee et al[1]) There is a bijection between regular exponential families and regular bregman divergences.

Some examples are :

- For 1-d Gaussian distribution $p(x|\mu) = \frac{1}{\sqrt{2\pi}} exp(-\frac{(x-\mu)^2}{2})$, the corresponding bregman divergence is $(x - \mu)^2$

- For d-D multinoulli $p(\boldsymbol{x}|\boldsymbol{\pi}) = \frac{N!}{\prod_{j=1}^{d} x_j!} \prod_{j=1}^{D} q_j^{x_j}$, the corresponding bregman divergence is $\sum_{j=1}^{D} x_j \log(\frac{x_j}{\mu_j}) - \sum_{j=1}^{D}(x_j - \mu_j) \quad (eqn\ 5.2)$

We can use this idea in the previous DP-means and HDP-means to obtain a new algorithm for hard clustering by replacing euclidean distance with any bregman divergence.

5.3 Bregman Hard clustering

- **Input** $x_1, x_2, ...x_n, \lambda$
- **Initialize** $\mu_1 = \frac{1}{n} \sum_{i=1}^{n} x_n$
- **Assignment** For each x_i,
  - Compute bregman divergence of the x_i with current cluster centers.
  - If $\min\limits_{c}\ d_\phi(\boldsymbol{x}, \boldsymbol{\mu}_c) < \lambda$, then assign it to cluster $\underset{c}{argmin}\ d_\phi(\boldsymbol{x}, \boldsymbol{\mu})$
  - Else, define a new cluster with its mean as x_i and assign x_i to this cluster.
- **Mean Update** For each cluster, set its means $\mu_c = \frac{1}{|l_j|} \sum_{\boldsymbol{x} \in l_j} \boldsymbol{x}$ where l_j is the set of points in j^{th} cluster

Algorithm 3: Bregman Hard Clustering [4]

The corresponding algorithm for Hierarchical Dirichlet process is similar, where we replace euclidean distance with the above defined bregman divergence

6 Experiments

For the purpose of this course project, we wrote the code for DP-means and HDP from scratch using C++. We used Bregman Divergence defined by $eqn\ 5.2$. We ran the code on *iris* data-set. We measured three parameters for each experiment;

1. Number of CLusters
2. NMI
3. Custom Validation(we will describe it later in this section)

6.1 NMI

$$\mathbb{NMI}(Y, C) = \frac{2 \times \mathbb{I}(Y; C)}{\mathbb{H}(Y) + \mathbb{H}(C)}$$

where:

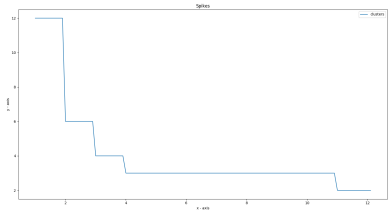
1. Y := class labels
2. C := cluster labels
3. $\mathbb{H}(.)$:= Entropy
4. $\mathbb{I}(Y; C)$:= Mutual Information b/w Y and C
   $\mathbb{I}(Y; C) := \mathbb{H}(Y) - \mathbb{H}(Y|C)$

Note that NMI not only penalizes bad clustering(putting together points having different labels into the same cluster, or putting points having same label into different clusters) but also discourages different number of clusters.

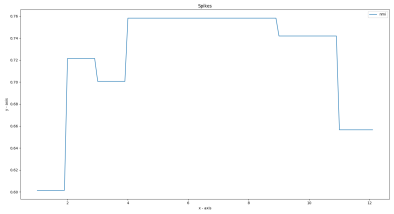
6.2 Custom Validation

6.3 Custom Validation

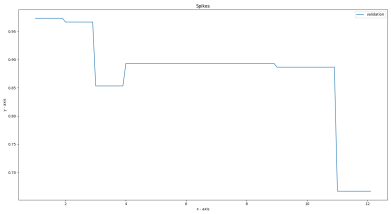
For each generated cluster label, we find the original cluster label that it maps to. So, by this we get a mapping from the cluster assignment to the original label. This acts as a predictor. We test this



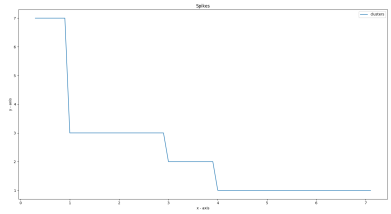
(a) Number of clusters : DP Means



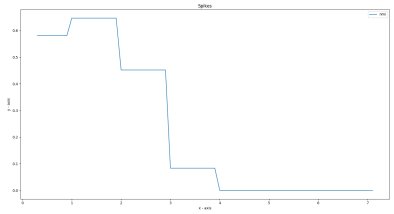
(b) NMI : DP Means



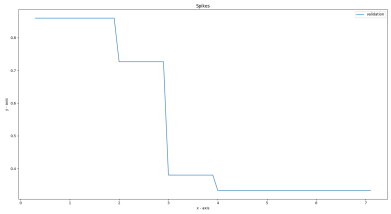
(c) Custom Validation : DP Means



(d) Number of clusters : DP Means with Bregman



(e) NMI : DP Means with Bregman



(f) Custom Validation : DP Means with Bregman

Figure 1: Note : On some pdf readers you may have to zoom in to see these images clearly

predictor on the clustering that we have obtained, and report the accuracy of this predictor. Note that, although this method does give a measure on how well the clustering is classifiable into the original labels, it does not discourage high number of clusters, and infact, it encourages higher number of clusters than number of labels.

6.4 Plots

For **DP-means** : As we can see from Figure 1a and Figure 1d, number of clusters decreases with λ. From Figure 1b and Figure 1e, NMI first increases then decreases with λ. From Figure 1c and Figure 1f, Custom Validation also first increases then decreases with λ.

For **HDP** : Figure 2-7 are 3d plots showing how number of clusters, NMI and Custom Validation for HDP means with Euclidean and Bregman Divergences respectively vary with λ_l and λ_g.

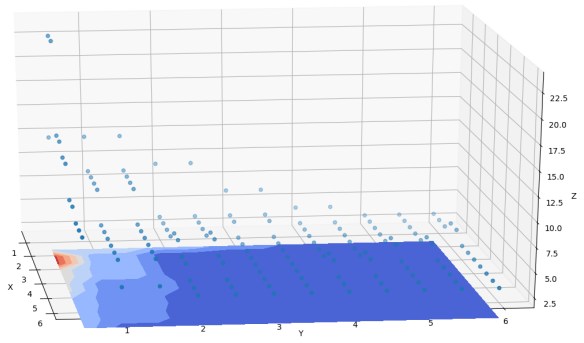


Figure 2: Number of clusters : HDP Means

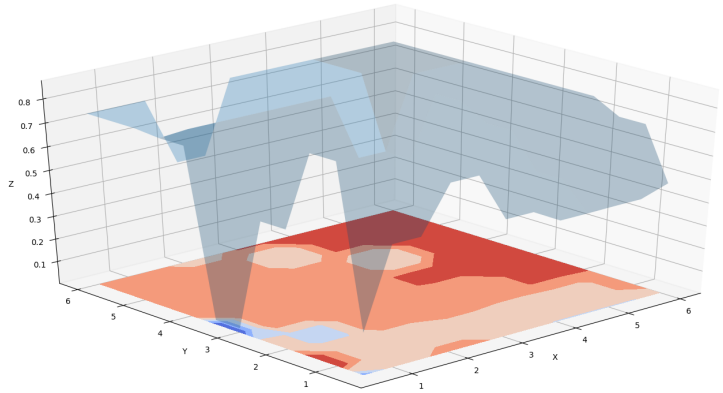


Figure 3: NMI : HDP Means

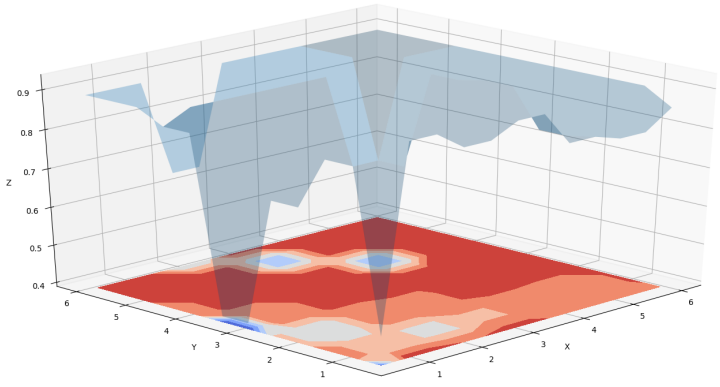


Figure 4: Custom Validation : HDP Means

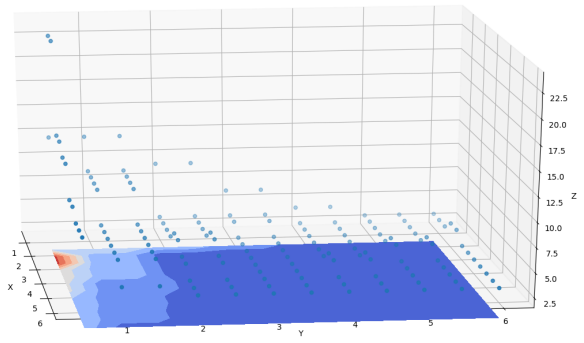


Figure 5: Number of clusters : HDP Means with Bregman

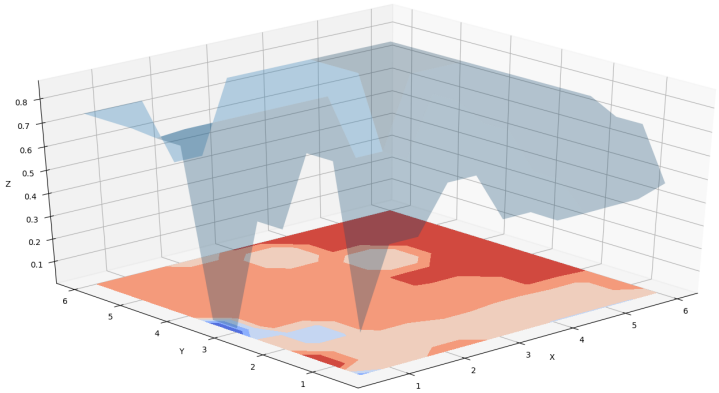


Figure 6: NMI : HDP Means with Bregman

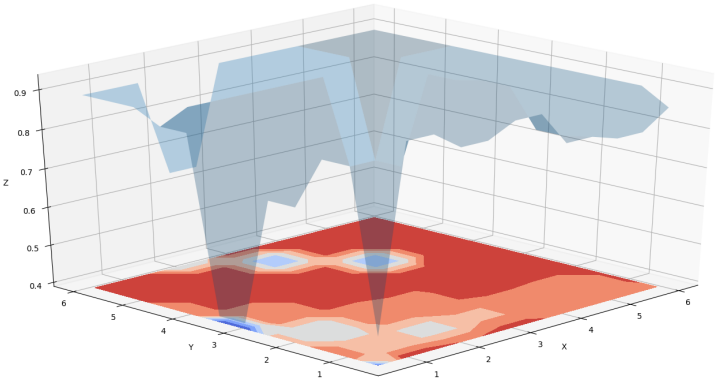


Figure 7: Custom Validation: HDP Means with Bregman

7 Take Aways and Future Work

We learned one should not code ML model in C++ (unless absolutely required) :p. We learned the concepts of Dirichlet and Hierarchical Dirichlet Processes.We got to know about generalization of distance namely Bregman Divergences and their relation with exponential family. We saw how small variance asymptotics can be very useful.

For future work, we can try soft assignment of points to clusters instead of hard assignment. The assignment vector for each point can be used as new representation for the point and can be tested on classification task. We can also test these algorithm on topic modelling task.

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, December 2005.

[2] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1(2):209–230, 03 1973.

[3] Samuel J. Gershman and David M. Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1 – 12, 2012.

[4] Ke Jiang, Brian Kulis, and Michael I. Jordan. Small-variance asymptotics for exponential family dirichlet process mixture models. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3158–3166. Curran Associates, Inc., 2012.

[5] Brian Kulis and Michael I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *CoRR*, abs/1111.0352, 2011.

[6] Yee Whye. Teh. Dirichlet process, 2010.

[7] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.