Small Variance Asymptotics for Non-parametric Bayesian Clustering

Abhishek Kumar | Manish Bera | Anubhav Mittal
Supervised by: Dr. Piyush Rai
CS698X 2017-18 II

November 16, 2019

Table of contents

1. Introduction
2. Dirichlet Process
   - Inference
   - Hard clustering
3. Hierarchical Dirichlet process Mixture Models
4. Extending to Exponential Family
   - Background
   - Bregman Hard clustering
5. Evaluation

Introduction

- Learning the correct model size is one of the biggest challenges

Introduction

- Learning the correct model size is one of the biggest challenges
- Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix number of parameters upfront.

Introduction

- Learning the correct model size is one of the biggest challenges
- Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix number of parameters upfront.
- 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.

Introduction

- Learning the correct model size is one of the biggest challenges
- Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix number of parameters upfront.
- 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 midddleground, so that we get a non parametric model which is scalable.

Introduction

- Learning the correct model size is one of the biggest challenges
- Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix number of parameters upfront.
- 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 midddleground, so that we get a non parametric model which is scalable.
- We start with a hard non-parameteric clustering algorithm.

Introduction

- Learning the correct model size is one of the biggest challenges
- Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix number of parameters upfront.
- 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 midddleground, so that we get a non parametric model which is scalable.
- We start with a hard non-parametric clustering algorithm.
- We then extend this algorithm to a hierarchical structure using Hierarchical Dirichlet process.

Introduction

- Learning the correct model size is one of the biggest challenges
- Bayesian frameworks offer ways to model infinite mixture models and models in which we don't fix number of parameters upfront.
- 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 midddleground, so that we get a non parametric model which is scalable.
- We start with a hard non-parameteric clustering algorithm.
- We then extend this algorithm to a hierarchical structure using Hierarchical Dirichlet process.
- Finally, we generalize the clustering algorithm, to use bregman divergence instead of just euclidean distance.

Dirichlet Process

For a random distribution to be G to be distributed according to a Dirichlet Process, its marginal distributions have to be Dirichlet distributed

Dirichlet Process

For a random distribution to be G to be distributed according to a Dirichlet Process, its marginal distributions have to be Dirichlet distributed

Definition

(Ferguson) We say 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

Dirichlet Process

For a random distribution to be G to be distributed according to a Dirichlet Process, its marginal distributions have to be Dirichlet distributed

Definition

(Ferguson) We say 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

- Intuitive roles for H and α as $\mathbb{E}[G(A)] = H(A)$ and $V[G(A)] = \frac{H(A)(1-H(A))}{\alpha+1}$

Dirichlet Process

For a random distribution to be G to be distributed according to a Dirichlet Process, its marginal distributions have to be Dirichlet distributed

Definition

(Ferguson) We say 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

- Intuitive roles for H and α as $\mathbb{E}[G(A)] = H(A)$ and $V[G(A)] = \frac{H(A)(1-H(A))}{\alpha+1}$

- Constructions like Blackwell-MacQueen urn scheme and Stick breaking process ensure existence of DP.

Posterior of DP

- 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 $n_k = \# \{i : \theta_i \in A_k\}$

Posterior of DP

- 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 $n_k = \# \{i : \theta_i \in A_k\}$
- Using conjugacy: $(G(A_1), \ldots, G(A_r) | \theta_1, \ldots \theta_n) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$

Posterior of DP

- 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 $n_k = \# \{i : \theta_i \in A_k\}$
- Using conjugacy: $(G(A_1), \ldots, G(A_r) | \theta_1, \ldots \theta_n) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$
- Simple algebra yields: $G | \theta_1, \ldots \theta_n \sim DP(\alpha + n, \frac{\alpha}{\alpha+n} H + \frac{n}{\alpha+n} \frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n})$

Posterior of DP

- 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 $n_k = \#\{i : \theta_i \in A_k\}$
- Using conjugacy: $(G(A_1), \ldots, G(A_r)|\theta_1, \ldots \theta_n) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$
- Simple algebra yields: $G|\theta_1, \ldots \theta_n \sim DP(\alpha + n, \frac{\alpha}{\alpha+n} H + \frac{n}{\alpha+n} \frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n})$
- The predictive distribution can be written as $\theta_{n+1} \in A|\theta_1, \ldots \theta_n \sim \frac{1}{\alpha+n}(\alpha H(A) + \sum_{i=1}^{n} \delta_{\theta_i}(A))$ ..... very very intuitive ☺

Posterior of DP

- 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 $n_k = \# \{i : \theta_i \in A_k\}$
- Using conjugacy: $(G(A_1), \ldots, G(A_r) | \theta_1, \ldots \theta_n) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$
- Simple algebra yields: $G | \theta_1, \ldots \theta_n \sim DP(\alpha + n, \frac{\alpha}{\alpha+n} H + \frac{n}{\alpha+n} \frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n})$
- The predictive distribution can be written as $\theta_{n+1} \in A | \theta_1, \ldots \theta_n \sim \frac{1}{\alpha+n}(\alpha H(A) + \sum_{i=1}^{n} \delta_{\theta_i}(A))$ ..... very very intuitive ☺
- This sequence of predictive distributions is called **Blackwell MacQueen Urn Scheme**

Posterior of DP

- 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 $n_k = \# \{i : \theta_i \in A_k\}$
- Using conjugacy: $(G(A_1), \ldots, G(A_r)|\theta_1, \ldots \theta_n) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$
- Simple algebra yields: $G|\theta_1, \ldots \theta_n \sim DP(\alpha + n, \frac{\alpha}{\alpha+n} H + \frac{n}{\alpha+n} \frac{\sum_{i=1}^{n} \delta_{\theta_i}}{n})$
- The predictive distribution can be written as $\theta_{n+1} \in A|\theta_1, \ldots \theta_n \sim \frac{1}{\alpha+n}(\alpha H(A) + \sum_{i=1}^{n} \delta_{\theta_i}(A))$ ..... very very intuitive ☺
- This sequence of predictive distributions is called **Blackwell MacQueen Urn Scheme**
- We have seen CRP and stick breaking process in class.

Dirichlet Process Mixture Model

- Can be seen as infinite dimensional generalization of Dirichlet distribution.

Dirichlet Process Mixture Model

- 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

Dirichlet Process Mixture Model

- 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 $\sigma^2 I$.

Dirichlet Process Mixture Model

- 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 $\sigma^2 I$.
-

$$\mu_1, \ldots.., \mu_k \sim G_0$$

$$\pi \sim \text{Dir}(\frac{\alpha}{k}, \frac{\alpha}{k}, \ldots. \frac{\alpha}{k})$$

$$z_1, \ldots., z_n \sim \text{Multinoulli}(\pi)$$

$$x_1, \ldots., x_n \sim \mathcal{N}(\mu_{z_i}, \sigma^2 I)$$

Dirichlet Process Mixture Model

- 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 $\sigma^2 I$.
- $$\mu_1, ..., \mu_k \sim G_0$$
  $$\pi \sim \text{Dir}(\frac{\alpha}{k}, \frac{\alpha}{k}, ... \frac{\alpha}{k})$$
  $$z_1, ..., z_n \sim \text{Multinoulli}(\pi)$$
  $$x_1, ..., x_n \sim \mathcal{N}(\mu_{z_i}, \sigma^2 I)$$
- Let $k \to \infty$ to get infinite mixture model

Inference

- We use Gibbs sampling for inference in the model.

Inference

- We use Gibbs sampling for inference in the model.
- For each point x_i, we assign it to cluster c with probability $\frac{n_{-i,c}}{Z} . \mathcal{N}(x_i | \mu_c, \sigma^2 I)$.

Inference

- We use Gibbs sampling for inference in the model.
- For each point x_i, we assign it to cluster c with probability $\frac{n_{-i,c}}{Z}.\mathcal{N}(x_i|\mu_c, \sigma^2 I)$.
- With probability $\frac{\alpha}{Z}. \int \mathcal{N}(x_i|\mu, \sigma^2 I) dG_0(\mu)$, we start a new cluster . For this newly formed cluster, we compute the means using the prior G_0 and the point x_i which created this cluster.

Inference

- We use Gibbs sampling for inference in the model.
- For each point x_i, we assign it to cluster c with probability $\frac{n_{-i,c}}{Z}.\mathcal{N}(x_i|\mu_c,\sigma^2 I)$.
- With probability $\frac{\alpha}{Z}.\int \mathcal{N}(x_i|\mu,\sigma^2 I)dG_0(\mu)$, we start a new cluster . 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.

Inference

- We use Gibbs sampling for inference in the model.
- For each point x_i, we assign it to cluster c with probability $\frac{n_{-i,c}}{Z}.\mathcal{N}(x_i|\mu_c, \sigma^2 I)$.
- With probability $\frac{\alpha}{Z}. \int \mathcal{N}(x_i|\mu, \sigma^2 I) dG_0(\mu)$, we start a new cluster . 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.
- Proceed in cyclic manner until convergence.

Hard clustering

- We first define G_0 (the prior distribution over the means) as $\mathcal{N}(0, \rho I)$.

Hard clustering

- We first define G_0 (the prior distribution over the means) as $\mathcal{N}(0, \rho I)$.
- Probability of starting new cluster :
  $\frac{\alpha}{Z}(2\pi(\rho + \sigma^2))^{-\frac{d}{2}} \cdot exp(-\frac{1}{2(\rho+\sigma^2)}||x_i||^2)$

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 :
  $\frac{n_{-i,c}}{Z}(2\pi\sigma^2)^{-\frac{d}{2}} \cdot exp(-\frac{1}{2\sigma^2}||x_i - \mu_c||^2)$

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 :
  $\frac{n_{-i,c}}{Z}(2\pi\sigma^2)^{-\frac{d}{2}} \cdot exp(-\frac{1}{2\sigma^2}||x_i - \mu_c||^2)$
- In hard clustering, we make $\sigma^2 \rightarrow 0$.

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 :
  $\frac{n_{-i,c}}{Z}(2\pi\sigma^2)^{-\frac{d}{2}} \cdot exp(-\frac{1}{2\sigma^2}||x_i - \mu_c||^2)$
- In hard clustering, we make $\sigma^2 \rightarrow 0$.
- The probabilities become binary(exact form in report) and resulting update turns out to be analogous to $k - means$ where we assign the point to closest mean

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 :
  $\frac{n_{-i,c}}{Z}(2\pi\sigma^2)^{-\frac{d}{2}} \cdot exp(-\frac{1}{2\sigma^2}||x_i - \mu_c||^2)$
- In hard clustering, we make $\sigma^2 \rightarrow 0$.
- The probabilities become binary(exact form in report) 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.

Underlying Objective function

1. We will show in report that the hard clustering algorithm minimizes the objective function:

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

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

Underlying Objective function

1. We will show in report that the hard clustering algorithm minimizes the objective function:

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

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

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

Hard Clustering 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, $\mu_c = \frac{\sum_{x_j \in l_c} x_j}{|l_c|}$.

Hierarchical Dirichlet process Mixture Models

- Assume that we have J data-sets with each having n_j data-points.

Hierarchical Dirichlet process Mixture Models

- Assume that we have J data-sets with each having n_j data-points.
- We want to learn clusters over these data-sets but we want them to share parameters and be related.

Hierarchical Dirichlet process Mixture Models

- Assume that we have J data-sets with each having n_j data-points.
- We want to learn clusters over these data-sets but we want them to share parameters and be related.
- HDP is a non parametric prior which allows mixture models to share components.

Hierarchical Dirichlet process Mixture Models

- Assume that we have J data-sets with each having n_j data-points.
- We want to learn clusters over these data-sets but we want them to share parameters and be related.
- HDP is a non parametric prior which allows mixture models to share components.

Hierarchical Dirichlet process Mixture Models

- Assume that we have J data-sets with each having n_j data-points.
- We want to learn clusters over these data-sets but we want them to share parameters and be related.
- HDP is a non parametric prior which allows mixture models to share components.

Definition

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

where G_0 is global measure and G_j's are specific to data-sets. This allows mixture models to share components.

Hierarchical Dirichlet process Mixture Models

- Assume that we have J data-sets with each having n_j data-points.
- We want to learn clusters over these data-sets but we want them to share parameters and be related.
- HDP is a non parametric prior which allows mixture models to share components.

Definition

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

where G_0 is global measure and G_j's are specific to data-sets. This allows mixture models to share components.

- There is a metaphor called *Chinese Restaurant Franchise* that gives an alternative view of HDP.

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.

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 :

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

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 :

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

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 $\mu_p = \frac{1}{|l_p|} \sum_{x_{ij} \in l_p} x_{ij}$

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 hard Gaussian HDP algorithm has not been shown here but will be there in report.

Definitions

- Exponential family distribution :

$$p(\boldsymbol{x}|\theta) = h(\boldsymbol{x})exp(\langle \boldsymbol{x}, \theta \rangle - \psi(\theta))$$

Definitions

- Exponential family distribution :

$$p(\boldsymbol{x}|\theta) = h(\boldsymbol{x})exp(\langle \boldsymbol{x}, \theta \rangle - \psi(\theta))$$

- 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 $\eta = \eta + 1$

Definitions

- Exponential family distribution :

$$p(\boldsymbol{x}|\theta) = h(\boldsymbol{x})exp(\langle \boldsymbol{x}, \theta \rangle - \psi(\theta))$$

- 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

(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 $d_\phi = \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\phi(\boldsymbol{y}) \rangle$

Bregman Divergence

- Squared euclidean distance is a bregman divergence with $\phi(x) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$

Bregman Divergence

- 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

$$d_\phi(\boldsymbol{\pi}, \boldsymbol{x}) = KL(\boldsymbol{\pi} || \boldsymbol{x})$$

Bregman Divergence

- 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

(Rockfellar 1970) Let ψ be a **proper**, **closed**, convex function with $\Theta = \mathit{interior}(\mathit{domain}(\psi))$. 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
- $\forall \theta_b \in bd(\Theta), \lim_{\theta \to \theta_b} ||\nabla \psi(\theta)|| \to \infty, \theta \in \Theta$

Legendre Function

Lemma

(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

Legendre Function

Lemma

(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

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

Legendre Function

Lemma

(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

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

$\psi^*(t) = \langle t, \theta^+ \rangle - \psi(\theta^+)$

Legendre Function

Lemma

(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

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

$$\psi^*(t) = \langle t, \theta^+\rangle - \psi(\theta^+)$$

Theorem

(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

Legendre Dual

Theorem (cntd..)

- (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^**.*
- $\nabla\psi^* = (\nabla\psi)^{-1}$

Legendre Dual

Theorem (cntd..)

- (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 $p_{\psi,\theta}$

Legendre Dual

Theorem (cntd..)

- (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 $\boldsymbol{\mu}(\theta) = \nabla\psi(\theta)$.

Legendre Dual

Theorem (cntd..)

- (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.

Legendre Dual

Theorem (cntd..)

- (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 and lemma, (Θ, ψ) and $(int(dom(\phi)), \phi)$ are legendre dual of each other.

Legendre Dual

Theorem (cntd..)

- (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 and lemma, (Θ, ψ) and $(int(dom(\phi)), \phi)$ are legendre dual of each other.
More importantly, $\nabla\psi^{-1}(\boldsymbol{\mu}) = \theta(\boldsymbol{\mu}) = \nabla\phi(\boldsymbol{\mu}) \quad (1)$
$\implies \phi(\boldsymbol{\mu}) = \langle\theta(\boldsymbol{\mu}), \boldsymbol{\mu}\rangle - \psi(\theta(\boldsymbol{\mu})) \quad (2)$

Relation with Exponential Family

Theorem

Let $p_{\psi,\theta}(\mathbf{x})$ 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 $p_{\psi,\theta}(\mathbf{x})$ can be uniquely expressed as
$p_{\psi,\theta}(\mathbf{x}) = exp(-d_\phi(\mathbf{x}, \boldsymbol{\mu}))b_\phi(\mathbf{x})$ where $b_\phi(\mathbf{x}) = exp(\phi(\mathbf{x}))h(\mathbf{x})$

Proof.

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

□

Relation with Exponential Family

Theorem

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.

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

□

Bijection

Theorem

(Banerjee et al) There is a bijection between regular exponential families and regular bregman divergences

Examples

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

Bijection

Theorem

(Banerjee et al) There is a bijection between regular exponential families and regular bregman divergences

Examples

- 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 $\sum_{j=1}^{D} x_j \log(\frac{x_j}{\mu_j}) - \sum_{j=1}^{D}(x_j - \mu_j)$

Bijection

Theorem

(Banerjee et al) There is a bijection between regular exponential families and regular bregman divergences

Examples

- 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 $\sum_{j=1}^{D} x_j \log(\frac{x_j}{\mu_j}) - \sum_{j=1}^{D}(x_j - \mu_j)$

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 above bregman divergence.

Bregman DP means

- **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 $\underset{c}{min}\ d_\phi(\boldsymbol{x}, \boldsymbol{\mu}_c) < \lambda$, 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

Bregman DP means

- **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 $\underset{c}{min}\ \ d_\phi(\boldsymbol{x}, \boldsymbol{\mu}_c) < \lambda$, 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

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

Evaluation metrics

1. NMI
2. Custom Validation

NMI

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

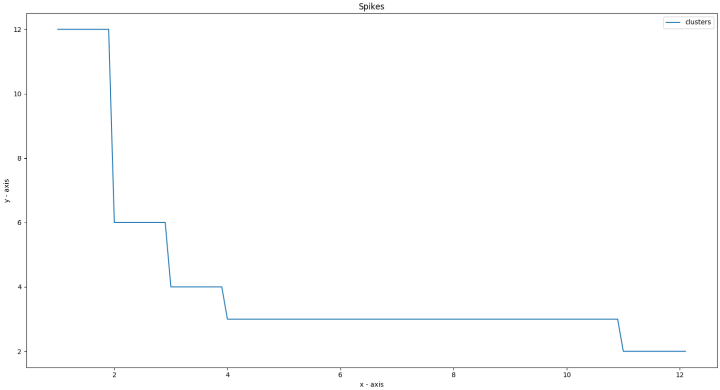
where:

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

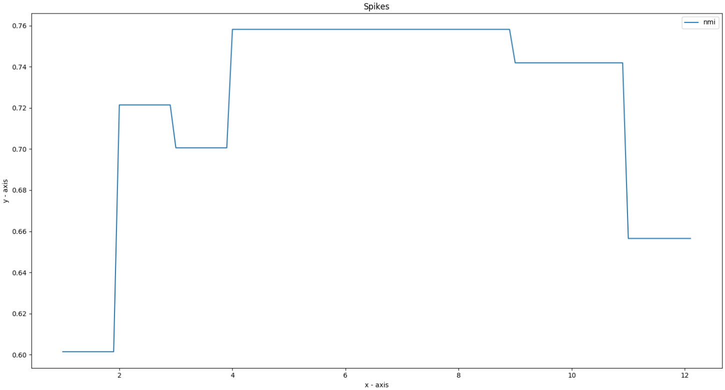
Custom Validation

- For each generated cluster label, we find the original cluster label that it maps to.
- We then find the accuracy of this mapping w.r.t. the clustering.

DP Means: No. of Clusters

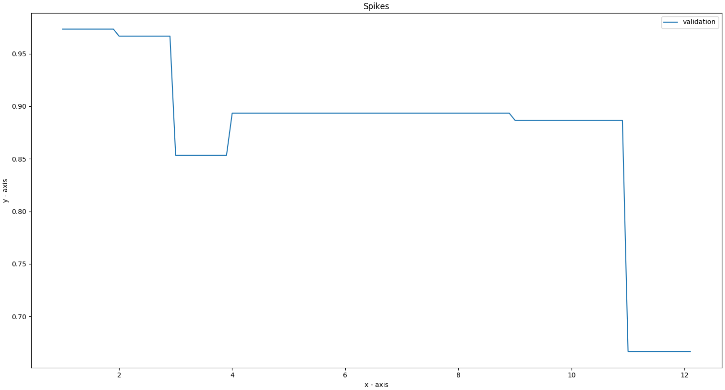


DP Means: NMI

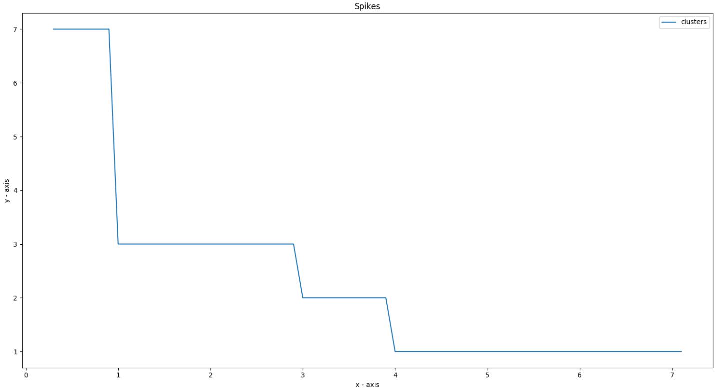


Spikes
nmi
0.76
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0.60
y - axis
2
4
6
8
10
12
x - axis

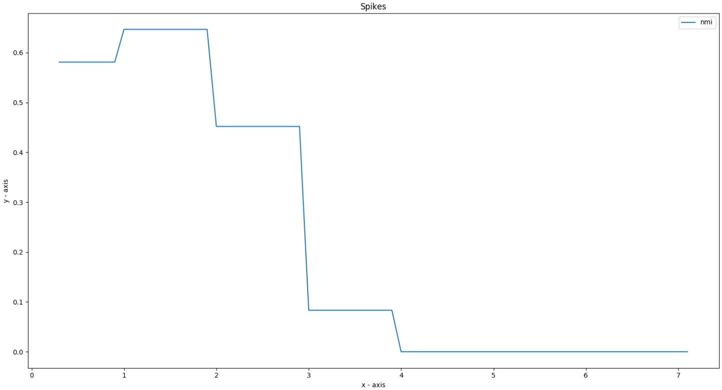
DP Means: Custom Validation



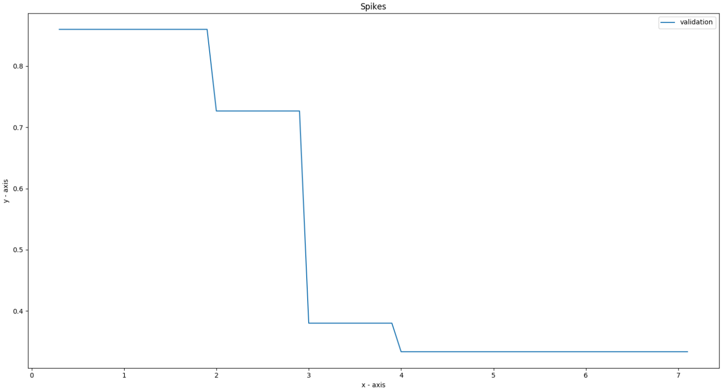
DP Means with Bregman Divergence: No. of Clusters



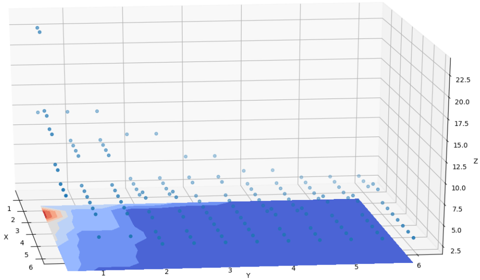
DP Means with Bregman Divergence: NMI



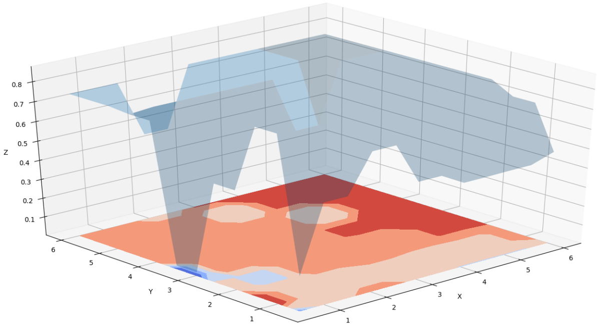
DP Means with Bregman Divergence: Custom Validation



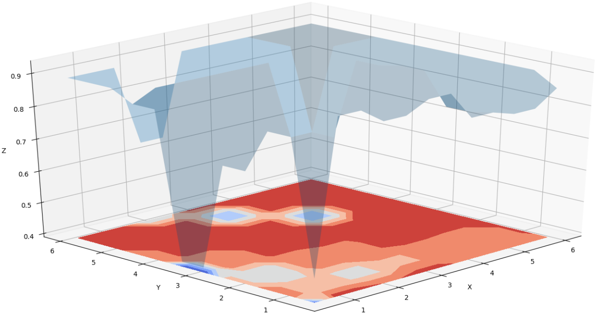
Hierarchical DP: No. of Clusters



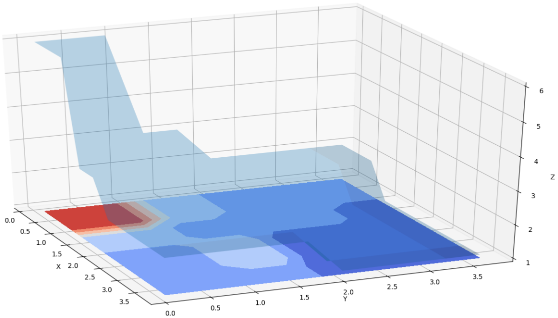
Hierarchical DP: NMI



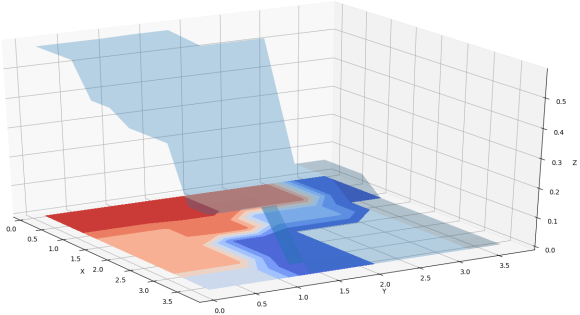
Hierarchical DP: Custom Validation



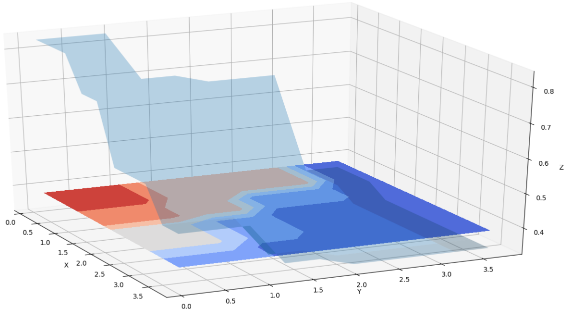
Hierarchical DP with Bregman Divergence: No. of Clusters



Hierarchical DP with Bregman Divergence: NMI



Hierarchical DP with Bregman Divergence: Custom Validation



Things learnt from Project

- Never (ever) code a ML model in C++ (unless absolutely required) :p
- Learnt the concepts of Dirichlet and Hierarchical Dirichlet Prior
- Learnt about Bregman Divergences
- Learnt how small variance asymptotics can be useful

Thank You ☺