Experiment in Taxi Routes in Porto

— GMDIL —

N. Benali, L. Fuentes, R. Maatouk

Université Paris-Saclay

March 27, 2024

Outline

Présentation de l'article

Article

- "Automatic Differentiation Variational Inference"
- Auteurs: *A. Kucukelbir, D.Tran, R. Ranganath, A. Gelman, D.M. Blei*
  Columbia University & Princeton University

Objectif présenté: Appliquer l'inférence variationnelle sur des grands jeux de données de manière plus efficace

Littérature

Soit un modèle probabiliste de distribution jointe $p(x, \theta)$ des variables latentes $\theta \in \mathbb{R}^K$ et des variables observées $x = x_{1:N}$

L'inférence variationnelle

↓

Estimer $p(\theta/x)$

↓

Cette estimation est: $q(\theta; \phi*)$

↓

Avec

$$\phi^* = \underset{\phi}{\text{argmin}}\ KL(q(\theta; \phi)||p(\theta|x)).$$

sous la contrainte $\text{supp}(q(\theta; \phi)) \subseteq \text{supp}(p(\theta|\mathbf{x}))$

Littérature

Problème

$\phi^* = \underset{\phi}{\text{argmin}}\ KL(q(\theta;\phi)||p(\theta|x))$ est défini en fonction de $p(\theta/x)$ qui est inconnu en pratique.

Solution

$$KL(q(\theta;\phi)||p(\theta|x))$$

$$= \mathbb{E}_q[\log(q(\theta;\phi))] - \mathbb{E}_q[\log(p(\theta|x))]$$

$$= \mathbb{E}_q[\log(q(\theta;\phi))] - \mathbb{E}_q[\log(\frac{p(\theta,x)}{p(x)})]$$

$$= \mathbb{E}_q[\log(q(\theta;\phi))] - \mathbb{E}_q[\log(p(\theta,x)) - \log(p(x))]$$

$$= \mathbb{E}_q[\log(q(\theta;\phi))] - \mathbb{E}_q[\log(p(\theta,x))] + log(p(x))$$

Littérature

Conclusion

minimiser $KL(q(\theta;\phi)||p(\theta|x))$ revient à maximiser $\mathbb{E}_q[\log(p(\theta,x))] - \mathbb{E}_q[\log(q(\theta;\phi))]$

On définit la fonction ELBO:

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\theta)}[\log(p(x,\theta))] + H(q(\theta;\phi))$$

C'est une fonction que nous pouvons évaluer car elle utilise des termes connus. Ainsi le problème devient:

$$\phi^* = \underset{\phi}{\text{argmax}}\mathcal{L}(\phi)$$

$$\text{tel que } \text{supp}(q(\theta;\phi)) \subseteq \text{supp}(p(\theta|\mathbf{x}))$$

ADVI

$$T : \text{supp}(p(\theta)) \to \mathbb{R}^K \text{est définie par } \xi = T(\theta)$$

On peut alors réécrire la densité jointe

$$p(x, \xi) = p(x, T^{-1}(\xi)) \cdot |\det J_{T^{-1}}(\xi)|$$

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\theta)}[\log(p(x, \theta))] + H(q(\theta; \phi))$$

$$\downarrow$$

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\xi;\phi)}[\log(p(x, \xi))] + H(q(\xi; \phi))$$

$$\downarrow$$

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\xi;\phi)}[\log(p(x, T^{-1}(\xi))) + \log(|\det J_{T^{-1}}(\xi)|)] + H(q(\xi; \phi))$$

ADVI

Nous avons beaucoup de choix d'approximation variationnelle $q(\xi;\phi)$ dans R^K. Les auteurs ont présenté deux options:

1. mean-field Gaussian:
   $q(\xi;\phi)$=Normal$(\xi;\mu,$ diag $(\sigma^2)\ )$=$\prod_{k=1}^{K}$ Normal$(\xi_k;\mu_k,\ \sigma_k^2\ )$, où le vecteur $\phi = (\mu_1,..,\mu_K,\sigma_1^2,...,\sigma_K^2)$ regroupe la moyenne et la variance de chaque facteur gaussien. On pose $\omega = \log(\sigma)$
2. full-rank Gaussian:
   $q(\xi;\phi)$=Normal$(\xi;\mu,\sum)$, où le vecteur $\phi = (\mu,\sum)$ regroupe la moyenne et la matrice de covariance. Pour garantir que $\sum$ reste toujours semi-définie positive, nous reparamétrons la matrice de covariance en utilisant une factorisation de Cholesky, $\sum = LL^T$. Le full-rank Gaussian généralise la mean-field Gaussian.

ADVI

Les auteurs développent un algorithme de montée de gradient stochastique qui utilise :

- l'intégration MC pour approximer les espérances non calculables, $q(\xi;\phi)$
- la différenciation automatique pour calculer les gradients.

terme non calculable

Nous ne pouvons pas utiliser directement la différenciation automatique sur l'ELBO:

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\xi;\phi)}[\log(p(x, T^{-1}(\xi))) + \log(|\det J_{T^{-1}}(\xi)|)] + H(q(\xi;\phi))$$

à cause de la présence du terme d'espérance non calculable. Cependant, nous pouvons différencier automatiquement les fonctions à l'intérieur de cette espérance.

ADVI

Nous utilisons une dernière transformation: la standarisation elliptique $\eta = S_\phi(\xi)$ qui convertit l'approximation variationnelle gaussienne $q(\xi;\phi)$ en une gaussienne centrée réduite $q(\eta)$. Cette nouvelle variable η utilise θ et ξ, ce qui permet :

1. calculer les derivées partielles par la règle de la chaîne
2. se ramener à une espérance plus facile à calculer.

Ainsi le calcul des gradients est possible. On obtient alors:

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathcal{N}(\eta;0,I)}[\log p(x, T^{-1}(S_\phi^{-1}(\eta))) + \log(|\det J_{T^{-1}}(S_\phi^{-1}(\eta))|)]$$

$$+H(q(\xi;\phi))$$

Il ne nous reste alors qu'à déterminer les paramètres de la distribution variationelle $q(\eta) = \mathcal{N}(\mu, \omega)$.

Contexte théorique: PPCA

Principe

On utilise une méthode appelée PPCA (Analyse en Composantes Principales Probabiliste) pour la réduction de dimension. Le modèle considère que les données sont générées selon la relation $x = wz + \sigma$, où w et z sont des paramètres latents et σ est l'écart-type du bruit.

La distribution des données x suit une distribution normale $\mathcal{N}(wz, \sigma)$. La PPCA utilise l'algorithme EM, un ensemble d'observations x et un ensemble de paramètres $\theta = (z, w, \sigma)$. L'objectif est d'estimer la distribution postérieure $p(\theta|x)$.
Cette distribution postérieure permet de déterminer les valeurs optimales des paramètres θ, ce qui facilite la réduction de dimension et la reconstruction des observations.

Contexte théorique: PPCA

Modélisation de la vraisemblance

On rappelle qu'on a $p(x, \theta) = p(x|\theta) \times p(\theta)$. On dispose également d'un jeu de données $\mathbf{x} = (x_i)_{0 \leq i \leq N}$ avec $x_i \in \mathbb{R}^D$ pour tout i. On veut faire une projection sur un sous-espace de dimension M $\leq D.Enfin, ondéfinit\theta$ le vecteur de paramètres du modèle tel que $\theta = (w, z, \sigma, \alpha)$.

On fait alors une série d'hypothèse permettant l'estimation in fine des paramètres du modèle:

- $x|\theta \sim \mathcal{N}(w^T z, \sigma)$
- $x|w, z, \sigma \sim \mathcal{N}(x; wz, \sigma \mathbf{I})$
- $w|\alpha \sim \mathcal{N}(w; \mathbf{0}, \sigma diag(\alpha)$
- $\alpha \sim InvGamma(\alpha; 1, 1)$

Ici, α désigne un vecteur de taille M qui va décider quelles composantes retenir lors de la PPCA.

Présentation du jeu de données

- *Taxi Service Trajectory - Prediction Challenge* 2015
- Trajectoires de taxis effectuées à Porto, Portugal.
- Coordonnées enregistrées toutes les 15 secondes
- $1^{ère}$ paire de coordonnées: point de départ du trajet
- Dernière paire: destination

- Description:
  - 442 taxis (Porto, Portugal)
  - TRIP_ID: identifiant unique pour chaque trajet
  - POLYLINE: liste de coordonnées GPS (chaîne de caractères)

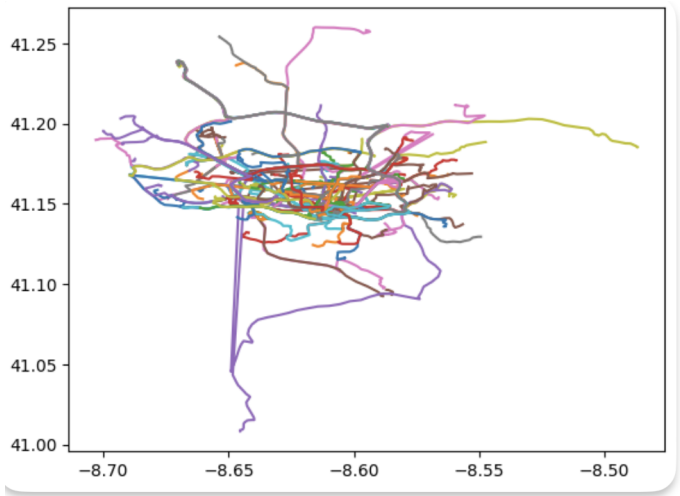
Objectif: Regrouper les trajectoires

Étape 1: Interpolation

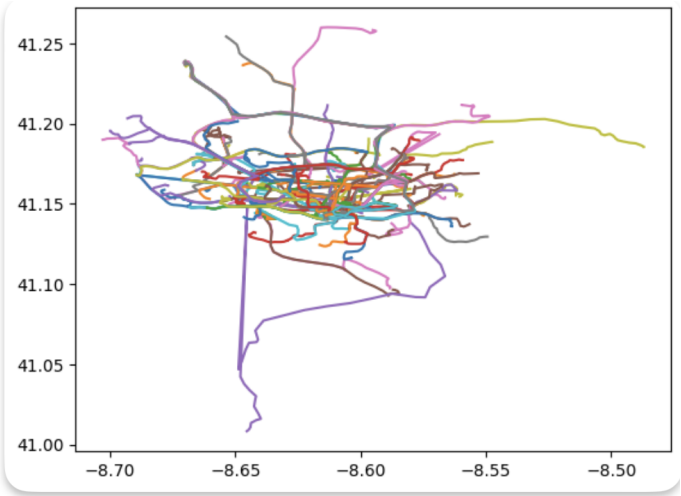
- Longueur des trajets dépend de la durée

→ Biais dans clusters introduit par la durée ($\neq$ Trajectoire)

✓ Solution: Interpolation
  - Normaliser longueur des trajets
  - Rend le clustering indépendant de la durée du trajet
  - Interpolation à 50 coordonnées (trajet moyen $\approx$ 13 minutes)



A) Avant interpolation



B) Après interpolation

Étape 2: Réduction de dimension

- **Objectif:** identifier une représentation des données de dimension inférieure (structures cachées)
- **Outils:**
  - PPCA: Déterminer une distribution q pour approcher $p(\theta|x)$ et optimiser θ
  - ADVI: déterminer les paramètres de la distribution variationnelle $q(\mu, \sigma^2 = e^{\omega})$

.

Étape 2: Réduction de dimension

Algorithm 1 Automatic Differentiation Variational Inference (ADVI)

Input Jeu de données $x = x_{1:N}$, modèle $p(x, \theta)$.
Configurer le compteur d'itérations $i = 1$.
Initialiser $\mu^{(1)} = 0$.
Initialiser $\omega^{(1)} = 0$ (mean-field) or $L^{(1)} = I$ (full-rank).
Déterminer η par recherche sur des valeurs finies.
while *Le changement dans l'ELBO est supérieur à un certain seuil* **do**
  Tirer M échantillons $\eta_m \sim \mathbb{N}(0, I)$.
  Approximer $\nabla_\mu \mathcal{L}$ par intégration MC.
  Approximer $\nabla_\omega \mathcal{L}$ or $\nabla_L \mathcal{L}$ par intégration MC.
  Calculer le step-size (i).
  Mise à jour de $\mu^{(i+1)} = \mu^{(i)} + \text{diag}(\rho^{(i)})\nabla_\mu \mathcal{L}$.
  Mise à jour de $\omega^{(i+1)} = \omega^{(i)} + \text{diag}(\rho^{(i)})\nabla_\omega \mathcal{L}$.
  or $L^{(i+1)} = L^{(i)} + \text{diag}(\rho^{(i)})\nabla_L \mathcal{L}$
  Incrémentation du compteur d'itérations
end while
Retourne $\mu^* = \mu^{(i)}$
Retourne $\omega^{=\omega^{(i)}}$ or $L^{=L^{(i)}}$

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta))\nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)$$

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta))\nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)\eta^T \text{diag}(e^\omega) + 1$$

Étape 2: Réduction de dimension

Objectif: Estimer les espérances par Monte Carlo

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}\left[\nabla_\theta \log(p(x,\theta))\nabla_\xi T^{-1}(\xi)\right] + \nabla_\xi \log(|J_{T^{-1}}|)$$

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}\left[\nabla_\theta \log(p(x,\theta))\nabla_\xi T^{-1}(\xi)\right] + \nabla_\xi \log(|J_{T^{-1}}|)\eta^T \text{diag}(e^\omega) + 1$$

Algorithm 1 Automatic Differentiation Variational Inference (ADVI)

Input Jeu de données $x = x_{1:N}$, modèle $p(x, \theta)$.
Configurer le compteur d'itérations $i = 1$.
Initialiser $\mu^{(1)} = 0$.
Initialiser $\omega^{(1)} = 0$ (mean-field) or $L^{(1)} = I$ (full-rank).
Déterminer η par recherche sur des valeurs finies.
while *Le changement dans l'ELBO est supérieur à un certain seuil* **do**
  Tirer M échantillons $\eta_m \sim \mathbb{N}(0, I)$.
  Approximer $\nabla_\mu \mathcal{L}$ par intégration MC.
  Approximer $\nabla_\omega \mathcal{L}$ or $\nabla_L \mathcal{L}$ par intégration MC.
  Calculer le step-size (i).
  Mise à jour de $\mu^{(i+1)} = \mu^{(i)} + \text{diag}(\rho^{(i)})\nabla_\mu \mathcal{L}$.
  Mise à jour de $\omega^{(i+1)} = \omega^{(i)} + \text{diag}(\rho^{(i)})\nabla_\omega \mathcal{L}$.
  or $L^{(i+1)} = L^{(i)} + \text{diag}(\rho^{(i)})\nabla_L \mathcal{L}$
  Incrémentation du compteur d'itérations
end while
Retourne $\mu^* = \mu^{(i)}$
Retourne $\omega^{=\omega^{(i)}}$ or $L^{=L^{(i)}}$

Démarche:

1. On tire M échantillons de $f(x, \theta_m)$
2. $\mathbb{E}_{\mathcal{N}(0,I)}[\text{f(x,}\theta\text{)}] \approx \frac{1}{M}\sum_{i=1}^{M} f(x, \theta_m)$

Étape 2: Réduction de dimension

Calcul de $log(p(x, \theta))$

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta)) \nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)$$

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta)) \nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)\eta^T \text{diag}(e^\omega) + 1$$

1. Tirer $\eta_i \sim \mathcal{N}(0, I)$
2. Standardisation elliptique: $\xi_i = S_\phi(\eta_i) = \frac{(\eta - \mu)}{\omega}$
3. Extraction de $\theta_i = T^{-1}(\xi_i)$
4. Extraction des échantillons des priors: z_i, w_i et σ_i
5. Calcul de $log(p(x, \theta)) = log(p(z_i)) + log(w_i) + log(\sigma_i) + log(p(x|\theta_i))$

Étape 2: Réduction de dimension

Calcul des gradients par différentiation automatique

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta)) \nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)$$

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta)) \nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)\eta^T \mathrm{diag}(e^\omega) + 1$$

Étape 2: Réduction de dimension

Calcul des gradients par différentiation automatique

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta)) \nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)$$

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(0,I)}[\nabla_\theta \log(p(x,\theta)) \nabla_\xi T^{-1}(\xi)] + \nabla_\xi \log(|J_{T^{-1}}|)\eta^T \mathrm{diag}(e^\omega) + 1$$

Il ne reste qu'un dernier calcul!

Étape 2: Réduction de dimension

Algorithm 1 Automatic Differentiation Variational Inference (ADVI)

Input Jeu de données $x = x_{1:N}$, modèle $p(x, \theta)$.
Configurer le compteur d'itérations $i = 1$.
Initialiser $\mu^{(1)} = 0$.
Initialiser $\omega^{(1)} = 0$ (mean-field) or $L^{(1)} = I$ (full-rank).
Déterminer η par recherche sur des valeurs finies.
while *Le changement dans l'ELBO est supérieur à un certain seuil* **do**
  Tirer M échantillons $\eta_m \sim \mathbb{N}(0, I)$.
  Approximer $\nabla_\mu \mathcal{L}$ par intégration MC.
  Approximer $\nabla_\omega \mathcal{L}$ or $\nabla_L \mathcal{L}$ par intégration MC.
  Calculer le step-size (i).
  Mise à jour de $\mu^{(i+1)} = \mu^{(i)} + \text{diag}(\rho^{(i)})\nabla_\mu \mathcal{L}$.
  Mise à jour de $\omega^{(i+1)} = \omega^{(i)} + \text{diag}(\rho^{(i)})\nabla_\omega \mathcal{L}$.
  or $L^{(i+1)} = L^{(i)} + \text{diag}(\rho^{(i)})\nabla_L \mathcal{L}$
  Incrémentation du compteur d'itérations
end while
Retourne $\mu^* = \mu^{(i)}$
Retourne $\omega^{=\omega^{(i)}}$ or $L^{=L^{(i)}}$

$$\rho_k^{(i)} = \eta \times i^{-1/2+\epsilon} \times (\tau + \sqrt{s_k^{(i)}})$$

$$s_k^{(i)} = \alpha g_k^{2(i)} + (1 - \alpha) s_k^{(i-1)}$$

Étape 2: Réduction de dimension

- On met à jour $\mu^{(i+1)}$ et $\omega^{(i+1)}$

$$\mu^{(i+1)} = \mu^{(i)} + diag(\rho^{(i)})\nabla_{\mu}\mathcal{L}$$

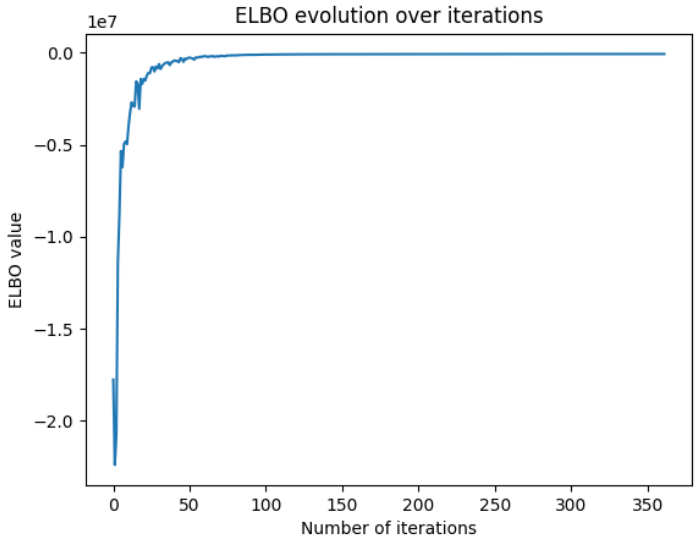
$$\omega^{(i+1)} = \omega^{(i)} + diag(\rho^{(i)})\nabla_{\omega}\mathcal{L}$$

Étape 2: Réduction de dimension

- On met à jour $\mu^{(i+1)}$ et $\omega^{(i+1)}$

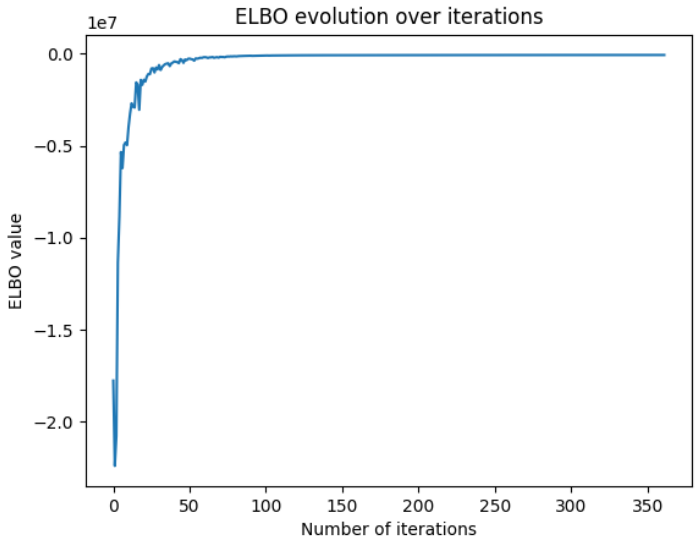
$$\mu^{(i+1)} = \mu^{(i)} + diag(\rho^{(i)})\nabla_{\mu}\mathcal{L}$$

$$\omega^{(i+1)} = \omega^{(i)} + diag(\rho^{(i)})\nabla_{\omega}\mathcal{L}$$



- Processus répété jusqu'à convergence de l'ELBO

Étape 2: Réduction de dimension



- On met à jour $\mu^{(i+1)}$ et $\omega^{(i+1)}$

$$\mu^{(i+1)} = \mu^{(i)} + diag(\rho^{(i)})\nabla_\mu \mathcal{L}$$

$$\omega^{(i+1)} = \omega^{(i)} + diag(\rho^{(i)})\nabla_\omega \mathcal{L}$$

- Processus répété jusqu'à convergence de l'ELBO

- Projection du dataset (x) dans l'espace de plus petite dimension:

1. On tire $\eta \sim q(\mu^*, e^{\omega^*}) = \mathcal{N}(\mu^*, e^{\omega^*})$
2. On extrait $\theta = T^{-1}(\eta)$ puis z, w, σ
3. Par la PPCA on a: $x \sim wz + \sigma$
4. Projection dans le sous-espace: $z = (w^T w)^{-1}(w^T(x - \sigma))$

Étape 3: Clustering

Le modèle de mélange outil fondamental pour le regroupement de données

- Attribuer les observations à des clusters
- Identification de structures sous-jacentes

- Appartenance au cluster C_k: estimation de $p(z \in C_k|\theta)$
- BayesianGausianMixture de sklearn

Étape 3: Clustering

Le modèle de mélange outil fondamental pour le regroupement de données

- Attribuer les observations à des clusters
- Identification de structures sous-jacentes

- Appartenance au cluster C_k: estimation de $p(z \in C_k|\theta)$
- BayesianGausianMixture de sklearn

Figure: Groupement des trajectoires 2 et 30 clusters [Extrait de l'article]

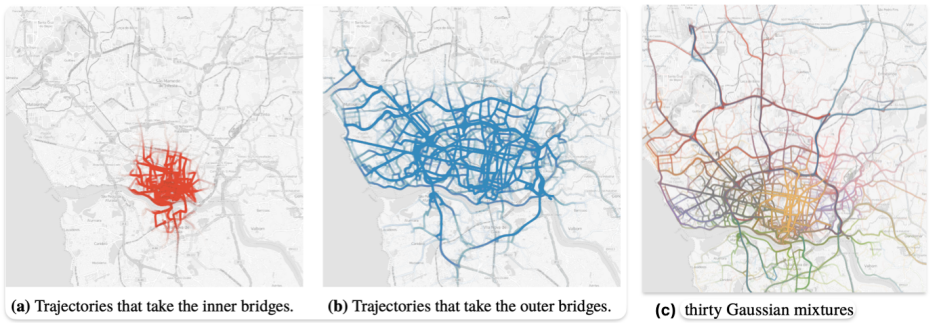
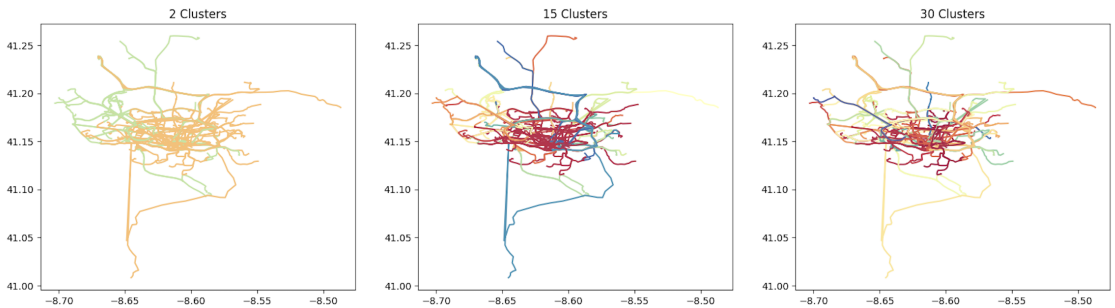


Figure: Groupement des trajectoires pour différents nombres de clusters



Conclusion

En conclusion

L'implémentation de l'algorithme ADVI nous a effectivement permis de réaliser un clustering sur un problème sans contraintes. Il est montré dans l'article que cet algorithme peut être appliqué à bien d'autres types de problèmes, de par son utilisation de l'inférence variationnelle.

Cependant, lors de notre lecture, et analyse de l'article, nous avons rencontré quelques difficultés:

- Comprendre Stan,
- Problème de reproductibilité: il manque des paramètres dans l'article,
- Survol de la partie application: l'article ne présente pas les transformations réalisées dans chaque cas de figure,
- Computation: nous avons dû réduire de beaucoup les dimensions du jeu de données par manque de ressources computationnelles,
- L'algorithme est assez sensible à la valeur du learning rate.