Prétraitement et analyse de données

Introduction à l'apprentissage automatique – GIF-4101 / GIF-7005

Professeur : Christian Gagné

Semaine 12

UNIVERSITÉ LAVAL

12.1 Prétraitement de données

Importance du prétraitement

- Algorithmes d'apprentissage sont sensibles aux valeurs d'entrées
  - Échelles des variables doivent être comparables
    - Variables d'échelles plus grandes dominantes dans mesures de similarité (ex. noyau gaussien) et distance (ex. euclidienne, manhattan)
    - Valeurs d'entrées élevées provoquent saturation de neurones sigmoïdes
  - Variables peuvent parfois être manquantes
    - Capteur défectueux, oublis/manques dans collecte, mesures ajoutées en cours de route
  - Dimensionnalité élevée
    - Sensibilité des algorithmes à la dimensionnalité
    - Redondance dans les mesures
- Prétaitement des données essentiel dans la pratique
  - Rarement accès à des données bien formatées et complètes, prêtes à être utilisées telles quelles
  - Important de comprendre la nature des données pour bien les traiter

Ajustement d'échelle

- Ajustement d'échelle des variables
  - Approche courante : ramener l'échelle des variables dans $[0, 1]$
  - Effectuer ajustements sur chaque variable indépendamment

$$x_i' = \frac{x_i - x_i^{min}}{x_i^{max} - x_i^{min}}, \quad i = 1, \ldots, D$$

où :

$$x_i^{max} = \max_{t=1,\ldots,N} x_i^t, \quad i = 1, \ldots, D$$

$$x_i^{min} = \min_{t=1,\ldots,N} x_i^t, \quad i = 1, \ldots, D$$

- Valeurs d'ajustements calculées sur un certain jeu de données
  - Nouvelle donnée pourrait avoir valeur de variable X_i à l'extérieur du domaine $[x_i^{min}, x_i^{max}]$
- Approche simple qui fait souvent un travail raisonnable

Standardisation

- Standardisation : ramener la distribution de chaque variable à une loi normale centrée-réduite, $x'_i \sim \mathcal{N}(0,1)$
  - Centrer la moyenne à zéro et ajuster pour un écart-type unitaire

$$x'_i = \frac{x_i - \mu_i}{\sigma_i}, \quad i = 1, \ldots, D$$

  - Moins sensible aux valeurs aberrantes qu'un ajustement d'échelle
- Traitement des variables indépendamment
  - Ne retire **pas** la covariance entre les variables, $\mathbf{\Sigma} \neq \mathbf{I}$
  - Transformation blanchissante (présentées plus tard aujourd'hui) permet d'obtenir données suivant une loi normale unitaire, $\mathbf{x}' \sim \mathcal{N}_D(0, \mathbf{I})$

Imputation

- Que faire si des valeurs de variables sont manquantes ?
  - Retirer les données avec valeurs manquantes
    - Perte de données pour l'apprentissage
    - Biais possible dans les données retirées
  - Marquer les variables manquantes pour l'algorithme d'apprentissage
    - Certains algorithmes d'apprentissage peuvent gérer les variables manquantes
  - Assigner une valeur par défaut aux variables manquantes (typiquement zéro)
  - Sélectionnner au hasard dans les autres données et assigner sa valeur à la variable manquante
  - Assigner valeur moyenne de la variable, $x_i' = \bar{x}_i$
    - Réduit la variance mesurée de la variable dans le jeu de données

Régression pour l'imputation

- Remplacement de variables manquantes peut dénaturer les données
  - Comment assigner une valeur plausible aux valeurs manquantes ?
- Utiliser l'apprentissage supervisé pour remplir valeurs manquantes
  - Pour chaque variable, apprendre modèle de régression pour imputer valeurs manquantes

$$x'_i = \mathrm{f}([x_1 \ \ldots \ x_{i-1}\, x_{i+1}, \ldots, x_D]^\top | \theta_i)$$

  - Les cibles r^t utilisées pour apprendre paramétrisation θ_i correspondent aux valeurs x_i pour les données où elles ne sont pas manquantes
- Valeurs plus fidèles aux données, mais peut encore réduire la variance comme régression va capturer valeurs les plus probables

12.2 Sélection de caractéristiques

Réduction de la dimensionnalité

- Réduction de la dimensionnalité
  - Passer d'un espace à D dimensions vers un espace à K dimensions, où $K < D$

$$X_1, \ldots, X_D \mapsto X'_1, \ldots, X'_K$$

- Approches possibles
  - Sélection de caractéristiques : choisir K variables parmi les D variables possibles

$$X_1, \ldots, X_D \mapsto X_{v_1}, \ldots, X_{v_K}$$
$$v_i \in \{1, \ldots, D\} \mid v_i \neq v_j, \forall j \leq i$$

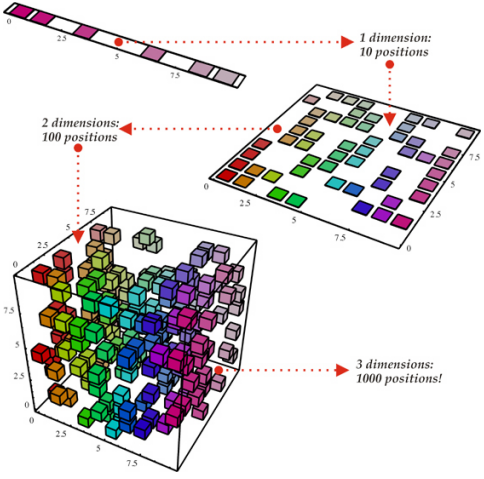
  - Extraction de caractéristiques : générer K variables comme des transformations des D variables d'origine

$$X_1, \ldots, X_D \mapsto \mathrm{f}_1(X_1, \ldots, X_D), \ldots, \mathrm{f}_K(X_1, \ldots, X_D)$$

Raisons pour réduire la dimensionnalité

- Malédiction de la dimensionnalité
  - Ajout d'une dimension augmente exponentiellement l'espace mathématique
  - 100 points équidistants de 0,01 en une dimension $\Rightarrow 10^{20}$ points nécessaires en 10 dimensions pour conserver la même densité
  - Grande dimensionnalité : complexité élevée en calculs et en mémoire
- Épargner des coûts de mesures
- Plus un modèle est simple, moins il y a de variances
- Plus facile d'expliquer avec moins de variables : extraction de connaissances
- Visualiser les données : analyse de résultats

Malédiction de la dimensionnalité



Source : Y. Bengio, http://www.iro.umontreal.ca/~bengioy/yoshua_en/research_files/CurseDimensionality.jpg, accédé le 2 octobre 2016.

Sélection de caractéristiques

- Objectif : trouver un sous-ensemble de K variables parmi $\{X_1, \ldots, X_D\}$, tout en préservant les performances
- Nombre de sous-ensembles possibles : $\binom{D}{K}$

$$\binom{10}{5} = 252, \quad \binom{50}{10} \approx 10^{10}, \quad \binom{100}{20} \approx 10^{20}$$

- Heuristique : *l'art d'inventer, de faire des découvertes*
  - Algorithme qui fournit rapidement (en temps polynomial) une solution réalisable, pas nécessairement optimale
  - Par opposition à un algorithme exact qui trouve une solution optimale

Évaluations des sous-ensembles de caractéristiques

- Approche filtre (*filter*)
  - Calculer la performance sans nouvel entraînement, avec une mesure indirecte (*proxy*)
  - Peu exigeant en calcul, mais résultats mitigés
- Approche enveloppe (*wrapper*)
  - Pour chaque ensemble de caractéristiques candidat, entraîner un nouveau classifieur
  - Évaluation de l'erreur empirique (entraînement, validation, validation croisée, etc.)
  - Beaucoup plus coûteux en calcul
- Approche embarquée (*embedded*) : sélection de caractéristiques intégrée à l'apprentissage du modèle

Sélection univariée

- Sélectionner selon mesures de performance des caractéristiques individuelles
  - Approche de base : sélectionner caractéristiques dont variance excède un seuil
    - Suppose que la variance décrit bien l'utilité de chaque caractéristique pour le classement
    - Bon pour filtrer caractéristiques de variance très faible ou nulle (éviter matrices de covariance singulières)
- Sélection selon d'autres critères
  - Corrélation entre caractéristiques (conserver ensemble de variables décorrélées)
  - Information mutuelle entre la caractéristique et la valeur cible

$$I(i) = \int_{X_i} \int_r p(X_i, r) \log \frac{p(X_i, r)}{p(X_i)\, p(r)} dr\, dX_i$$

  - Effet sur l'erreur empirique, avec imputation des variables non sélectionnées

Sélection avant séquentielle

- Construire graduellement l'ensemble de caractéristiques, en ajoutant la variable la plus prometteuse
  1. Démarrer avec un ensemble de caractéristiques vide
  2. Ajouter la caractéristique améliorant le plus (selon un certain critère) l'ensemble de caractéristiques
  3. Répéter étape 2 tant que le critère d'arrêt n'est pas atteint
- Algorithme vorace : prendre itérativement des décisions locales
  - Ne tient pas compte de relations complexes entre les variables
  - Exemple :
    - Variables X_a, X_b et X_c prises individuellement ou en paires $\Rightarrow$ faible gain
    - Les trois variables prises ensemble $\Rightarrow$ fort gain
- Complexité algorithmique $O(KD)$

Algorithme de sélection avant séquentielle

1. Initialiser l'algorithme :
   - Créer l'ensemble de caractéristiques sélectionnées :
   $$F^0 = \emptyset$$
   - Créer l'ensemble de caractéristiques non sélectionnées :
   $$G^0 = \{X_1, \ldots, X_D\}$$

2. Pour $t = 1, \ldots, D$, tant que le critère d'arrêt n'est pas atteint :
   2.1 Déterminer la caractéristique réduisant le plus l'erreur :
   $$X_j = \underset{X_i \in G^{t-1}}{\operatorname{argmin}} E(F^{t-1} + \{X_i\})$$
   2.2 Sélectionner cette caractéristique en l'ajoutant à F et la retirant de G :
   $$F^t = F^{t-1} + \{X_j\}, \quad G^t = G^{t-1} \backslash \{X_j\}$$

3. Retourner le sous-ensemble final F de caractéristiques sélectionnées

Critères d'arrêt

- Critères d'arrêt possibles
  - Arrêter lorsque K caractéristiques sont sélectionnées
  - Arrêter lorsque toutes les caractéristiques sont sélectionnées
    - Retourner l'ensemble de caractéristiques vu avec erreur empirique minimale
  - Arrêter lorsque réduction de l'erreur est inférieure à un seuil

$$E(F^t) - E(F^{t+1}) < \epsilon$$

Sélection arrière séquentielle

- Approche inverse : partir avec toutes les variables et retirer itérativement celles qui contribuent le moins

1. Créer l'ensemble de caractéristiques sélectionnées :

$$F^D = \{X_1, \ldots, X_D\}$$

2. Pour $t = D-1, D-2, \ldots, 1$, tant que le critère d'arrêt n'est pas atteint :
   2.1 Déterminer la caractéristique contribuant le moins :

$$X_j = \underset{X_i \in F^{t+1}}{\operatorname{argmin}} E(F^{t+1} \backslash \{X_i\})$$

   2.2 Retirer cette caractéristique de F :

$$F^t = F^{t+1} \backslash \{X_j\}$$

3. Retourner le sous-ensemble final F de caractéristiques sélectionnées

Autres approches pour sélection de caractéristiques

- Ajouter-*l*-retirer-*r*
  - Hybride entre les approches séquentielles avant et arrière, évite certains minimums locaux
- *Branch-and-bound*
  - Organiser les caractéristiques en arbres, selon leurs similarités
  - Réduction par coupe dans l'arbre pour éliminer les caractéristiques similaires
- Algorithme évolutionnaire multiobjectif
  - Optimisation stochastique à base de population, inspirée de l'évolution naturelle
  - Recherche globale : un individu = un sous-ensemble de caractéristiques
  - Optimisation selon deux objectifs simultanément : réduire l'erreur et réduire le nombre de caractéristiques choisies

12.3 Analyse en composantes principales

Extraction de caractéristiques

- Sélection de caractéristiques
  - Point fort : permet de retirer complètement des mesures
  - Point faible : parfois, plusieurs variables sont pauvres en information lorsque prises individuellement, mais riche en information lorsque prises collectivement
    - Exemple : reconnaissance d'objets à partir des pixels d'images
- Extraction de caractéristiques
  - Projection d'un espace à D dimensions vers un espace à K dimensions
  - Point fort : permet de compresser l'information vers un espace de dimensionnalité réduite
  - Point faible : exige de prendre toutes les D mesures originales

Rappel : transformations linéaires

- Translation dans un espace

$$\mathbf{y} = \mathbf{x} + \mathbf{u}$$

- Transformation linéaire selon matrice $\mathbf{A}$ de taille $K \times D$

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

  - Rotation dans un espace (exemple en 2D)

$$\mathbf{A} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- Formulation générale

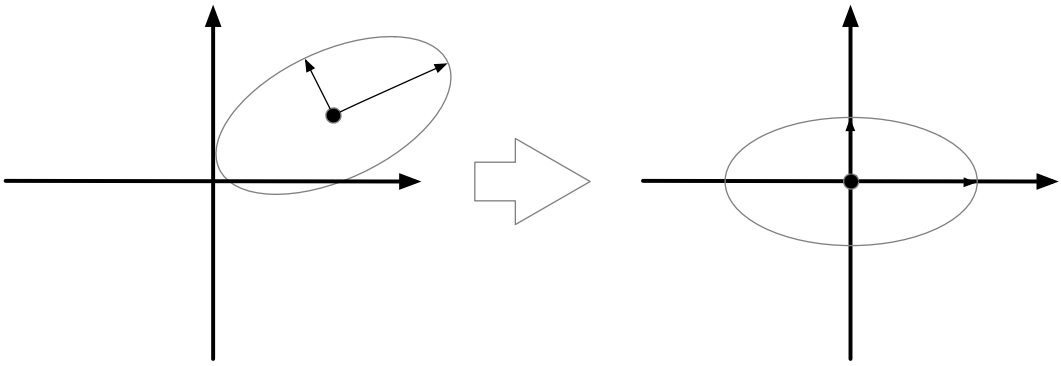
$$\mathbf{y} = \mathbf{A}(\mathbf{x} + \mathbf{u})$$

Analyse en composantes principales

- Analyse en composantes principales (ACP)
  - Projection linéaire dans un espace à K dimensions, avec une perte minimale d'information
  - Variance = information
  - Revient à extraire des vecteurs dans les directions de variances maximales
  - Non supervisée : n'utilise que les mesures, pas les étiquettes de classe
- 1ère composante principale : direction de variance maximale
- 2e composante principale : direction de variance maximale orthogonale à la première composante
- Transformation linéaire, centrée sur le vecteur moyen

$$\mathbf{z} = \mathbf{W}^\top (\mathbf{x} - \boldsymbol{\mu})$$

Illustration de l'ACP



12.4 Dérivation de l'ACP

Multiplicateurs de Lagrange

- Méthode de résolution de problèmes d'optimisation sous contraintes
  - Exemple : maximiser $f(\mathbf{x})$ sous contraintes que $g(\mathbf{x}) = 0$
  - Il existe un paramètre $\lambda \neq 0$ de sorte que

$$\nabla f + \lambda \nabla g = 0$$

  - Équation correspondante avec multiplicateur de Lagrange

$$L(\mathbf{x},\lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

- Maximum obtenu en trouvant $\nabla L(\mathbf{x},\lambda) = 0$
  - Si on est intéressé uniquement au $\mathbf{x}$, on peut éliminer λ sans devoir l'évaluer

Exemple avec le multiplicateur de Lagrange

- Maximiser $f(x_1,x_2) = 1 - x_1^2 - x_2^2$ sujet à la contrainte $g(x_1,x_2) = x_1 + x_2 - 1 = 0$
- Formulation avec multiplicateur de Lagrange

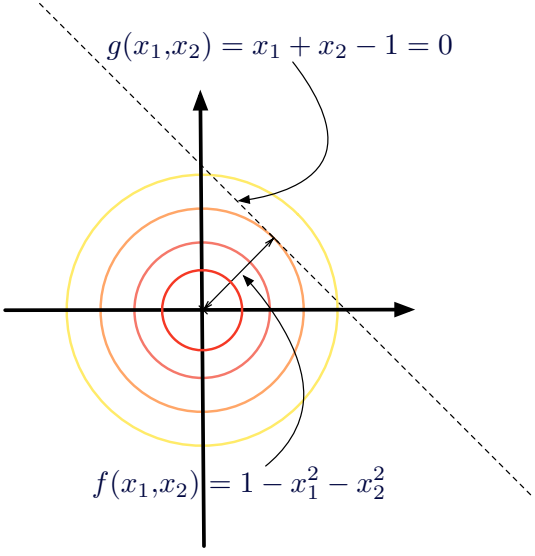
$$L(x_1,x_2,\lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$$

- Résolution de $\nabla L(x_1,x_2,\lambda) = 0$

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= -2x_1 + \lambda = 0 \\ \frac{\partial L}{\partial x_2} &= -2x_2 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= x_1 + x_2 - 1 = 0 \end{aligned}$$

- Solution au système d'équations : $x_1 = 0{,}5$, $x_2 = 0{,}5$ et $\lambda = 1$

Exemple avec le multiplicateur de Lagrange



Dérivation de l'ACP

- Première composante $\mathbf{w}_1$: direction de la variance principale

$$z_1 = \mathbf{w}_1^\top \mathbf{x}$$

- Seule la direction est importante, $\|\mathbf{w}_1\| = 1$
- Si $\mathrm{Cov}(\mathbf{x}) = \boldsymbol{\Sigma}$ alors $\mathrm{Var}(z_1) = \mathbf{w}_1^\top \boldsymbol{\Sigma} \mathbf{w}_1$

$$\begin{aligned}
\mathbb{E}[\mathbf{w}^\top \mathbf{x}] &= \mathbf{w}^\top \mathbb{E}[\mathbf{x}] = \mathbf{w}^\top \boldsymbol{\mu} \\
\mathrm{Var}(\mathbf{w}^\top \mathbf{x}) &= \mathbb{E}\left[(\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \boldsymbol{\mu})^2\right] \\
&= \mathbb{E}\left[(\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \boldsymbol{\mu})(\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \boldsymbol{\mu})^\top\right] \\
&= \mathbb{E}\left[\mathbf{w}^\top (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{w}\right] \\
&= \mathbf{w}^\top \mathbb{E}\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top\right] \mathbf{w} \\
&= \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w}
\end{aligned}$$

Première composante principale

- On recherche le vecteur $\mathbf{w}_1$ qui maximise $\mathrm{Var}(z_1)$, avec contrainte que $\mathbf{w}_1^\top \mathbf{w}_1 = 1$
- Résolution par méthode de Lagrange

$$\begin{aligned} L(\mathbf{w}_1, \alpha) &= \mathbf{w}_1^\top \mathbf{\Sigma} \mathbf{w}_1 - \alpha \left(\mathbf{w}_1^\top \mathbf{w}_1 - 1 \right) \\ \frac{\partial L(\mathbf{w}_1, \alpha)}{\partial \mathbf{w}_1} &= 2\mathbf{\Sigma}\mathbf{w}_1 - 2\alpha \mathbf{w}_1 = 0 \\ \mathbf{\Sigma}\mathbf{w}_1 &= \alpha \mathbf{w}_1 \end{aligned}$$

- Par définition, $\mathbf{\Sigma}\mathbf{w}_1 = \alpha \mathbf{w}_1$ est vrai lorsque $\mathbf{w}_1$ est un vecteur propre de $\mathbf{\Sigma}$ et que α est la valeur propre associée
- On choisi le vecteur propre avec la valeur propre la plus grande, $\alpha = \lambda_1$, étant donné que :

$$\mathrm{Var}(\mathbf{w}_1^\top \mathbf{x}) = \mathbf{w}_1^\top \mathbf{\Sigma} \mathbf{w}_1 = \alpha \mathbf{w}_1^\top \mathbf{w}_1 = \alpha$$

Deuxième composante principale

- Vecteur $\mathbf{w}_2$ maximise $\mathrm{Var}(z_2)$
  - Contrainte 1 : $\mathbf{w}_2$ est unitaire, $\mathbf{w}_2^\top \mathbf{w}_2 = 1$
  - Contrainte 2 : $\mathbf{w}_2$ est orthogonal à $\mathbf{w}_1$, $\mathbf{w}_2^\top \mathbf{w}_1 = 0$
- Résolution par méthode de Lagrange

$$\begin{aligned} L(\mathbf{w}_1,\mathbf{w}_2,\alpha,\beta) &= \mathbf{w}_2^\top \boldsymbol{\Sigma} \mathbf{w}_2 - \alpha \left(\mathbf{w}_2^\top \mathbf{w}_2 - 1 \right) - \beta(\mathbf{w}_2^\top \mathbf{w}_1 - 0) \\ \frac{\partial L(\mathbf{w}_1,\mathbf{w}_2,\alpha,\beta)}{\partial \mathbf{w}_2} &= 2\boldsymbol{\Sigma}\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0 \\ \mathbf{w}_1^\top \frac{\partial L(\mathbf{w}_1,\mathbf{w}_2,\alpha,\beta)}{\partial \mathbf{w}_2} &= 2\mathbf{w}_1^\top \boldsymbol{\Sigma}\mathbf{w}_2 - 2\alpha\mathbf{w}_1^\top\mathbf{w}_2 - \beta\mathbf{w}_1^\top\mathbf{w}_1 = 0 \end{aligned}$$

- Étant donné que $\boldsymbol{\Sigma}\mathbf{w}_1 = \lambda_1 \mathbf{w}_1$, alors :

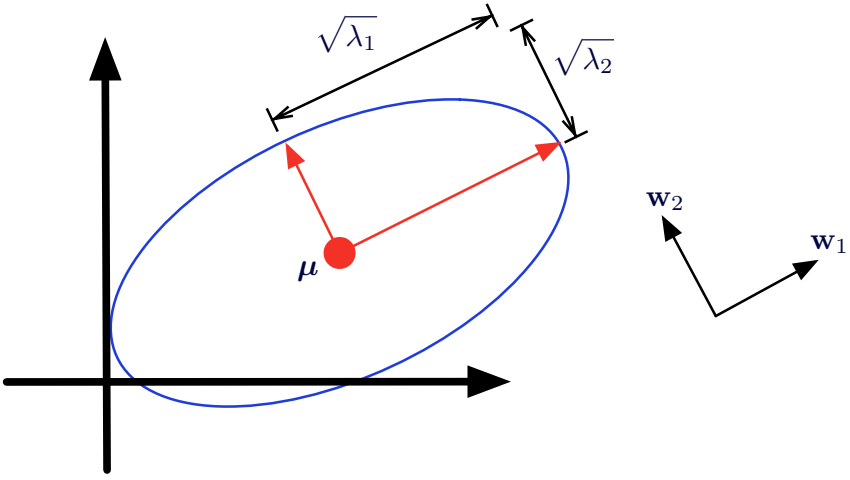
$$\begin{aligned} \mathbf{w}_1^\top \boldsymbol{\Sigma}\mathbf{w}_2 = \mathbf{w}_2^\top \boldsymbol{\Sigma}\mathbf{w}_1 = \lambda_1 \mathbf{w}_2^\top \mathbf{w}_1 &= 0 \\ 2\mathbf{w}_1^\top \boldsymbol{\Sigma}\mathbf{w}_2 - 2\alpha\mathbf{w}_1^\top\mathbf{w}_2 - \beta\mathbf{w}_1^\top\mathbf{w}_1 = -\beta\mathbf{w}_1^\top\mathbf{w}_1 = 0 &\Rightarrow \beta = 0 \end{aligned}$$

- On simplifie donc $2\boldsymbol{\Sigma}\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0$

Deuxième composante principale

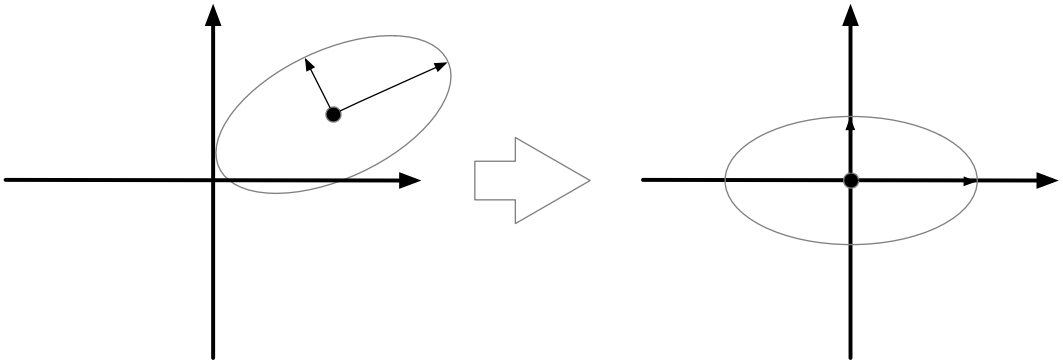
- $\boldsymbol{\Sigma}\mathbf{w}_2 = \alpha\mathbf{w}_2$ implique que $\mathbf{w}_2$ est également un vecteur propre de $\boldsymbol{\Sigma}$
  - Comme on veut maximiser $\mathrm{Var}(\mathbf{w}_2^\top \mathbf{x})$, on choisi le vecteur propre associé à la deuxième plus grande valeur propre, $\alpha = \lambda_2$
- On procède de la même façon pour les autres dimensions, en choisissant comme $\mathbf{w}_i$ les vecteurs propres, en ordre décroissant de valeurs propres associées
- Matrice de rotation $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_K]$ contient donc les $K \leq D$ premiers vecteurs propres (avec plus grandes valeurs propres)
- Propriétés supplémentaires
  - Comme $\boldsymbol{\Sigma}$ est symétrique, les vecteurs propres sont orthogonaux
  - Comme $\mathbf{w}_i$ sont unitaires, ils forment une base orthonormale
  - Si $\boldsymbol{\Sigma}$ est définie positive ($\mathbf{x}^\top\boldsymbol{\Sigma}\mathbf{x} > 0$, $\forall \mathbf{x} \neq 0$), toutes les valeurs propres sont non nulles, $\lambda_i \neq 0$, $\forall \lambda_i$
  - Sinon, le rang de $\boldsymbol{\Sigma}$ donne le nombre de valeurs propres non nulles

Vecteurs/valeurs propres et ACP



ACP comme transformation linéaire

$$\mathbf{z} = \mathbf{W}^\top(\mathbf{x} - \mathbf{m})$$



12.5 Dérivation alternative de l'ACP

Dérivation alternative

- Dérivation alternative de l'ACP
  - Recherche d'une transformation $\mathbf{z} = \mathbf{W}^\top \mathbf{x}$, où variables de $\mathbf{z}$ ne sont pas corrélées
  - Revient à chercher $\mathbf{W}$ afin que $\mathrm{Cov}(\mathbf{z}) = \mathbf{D}'$ soit diagonale
- Supposons $\mathbf{C}$, matrice $D \times D$, où colonne $\mathbf{c}_i$ est i-ème vecteur propre de $\mathbf{S}$, l'estimateur de $\mathbf{\Sigma}$
  - Donc $\mathbf{C}\mathbf{C}^\top = \mathbf{C}^\top\mathbf{C} = \mathbf{I}$

$$
\begin{aligned}
\mathbf{S} &= \mathbf{S}\mathbf{C}\mathbf{C}^\top \\
&= \mathbf{S}[\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_D]\mathbf{C}^\top \\
&= [\mathbf{S}\mathbf{c}_1\ \mathbf{S}\mathbf{c}_2\ \cdots\ \mathbf{S}\mathbf{c}_D]\mathbf{C}^\top \\
&= [\lambda_1\mathbf{c}_1\ \lambda_2\mathbf{c}_2\ \cdots\ \lambda_D\mathbf{c}_D]\mathbf{C}^\top \\
&= \lambda_1\mathbf{c}_1\mathbf{c}_1^\top + \lambda_2\mathbf{c}_2\mathbf{c}_2^\top + \cdots + \lambda_D\mathbf{c}_D\mathbf{c}_D^\top \\
&= \mathbf{C}\mathbf{D}\mathbf{C}^\top
\end{aligned}
$$

- Matrice $\mathbf{D}$ est diagonale, avec valeurs propres $\lambda_1, \lambda_2, \ldots, \lambda_D$

Décomposition spectrale

- $\mathbf{CDC}^\top$ est la décomposition spectrale de $\mathbf{S}$
- Comme $\mathbf{C}$ est orthogonale et $\mathbf{CC}^\top = \mathbf{C}^\top\mathbf{C} = \mathbf{I}$

$$\begin{aligned} \mathbf{S} &= \mathbf{CDC}^\top \\ \mathbf{C}^\top\mathbf{SC} &= \mathbf{C}^\top\mathbf{CDC}^\top\mathbf{C} \\ \mathbf{C}^\top\mathbf{SC} &= \mathbf{D} \end{aligned}$$

- On sait que $\mathrm{Cov}(\mathbf{z}) = \mathbf{W}^\top\mathbf{SW}$ et qu'on veut $\mathrm{Cov}(\mathbf{z})$ diagonale
  - On pose donc que $\mathbf{W} = \mathbf{C}$

12.6 Illustrations de l'ACP

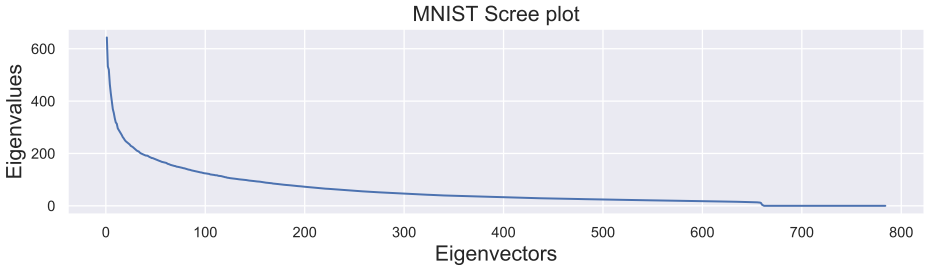
Proportion de la variance

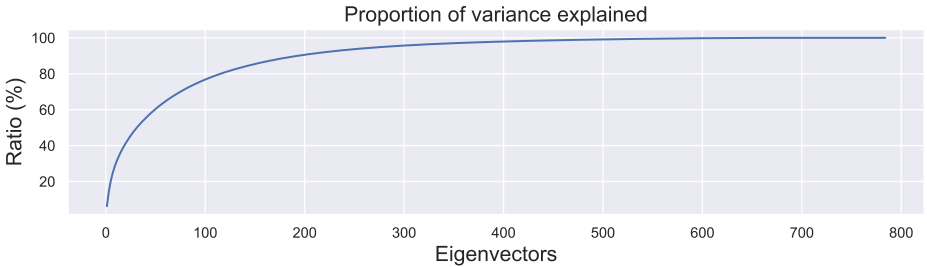
- Valeur propre λ_i indique la contribution de la composante associée à la variance
- Proportion de la variance expliquée par les K composantes principales

$$\mathrm{PdV} = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_K}{\lambda_1 + \lambda_2 + \cdots + \lambda_K + \cdots + \lambda_D}$$

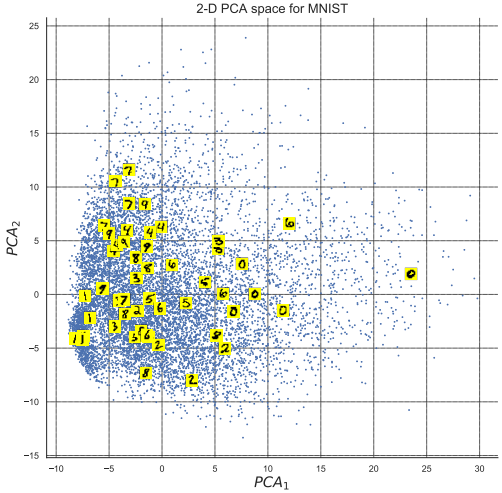
- Forte corrélation entre les variables $\Rightarrow$ peu de composantes avec valeurs propres élevées
- Tracé en éboulis : tracé du tri décroissant des valeurs propres

Tracé en éboulis

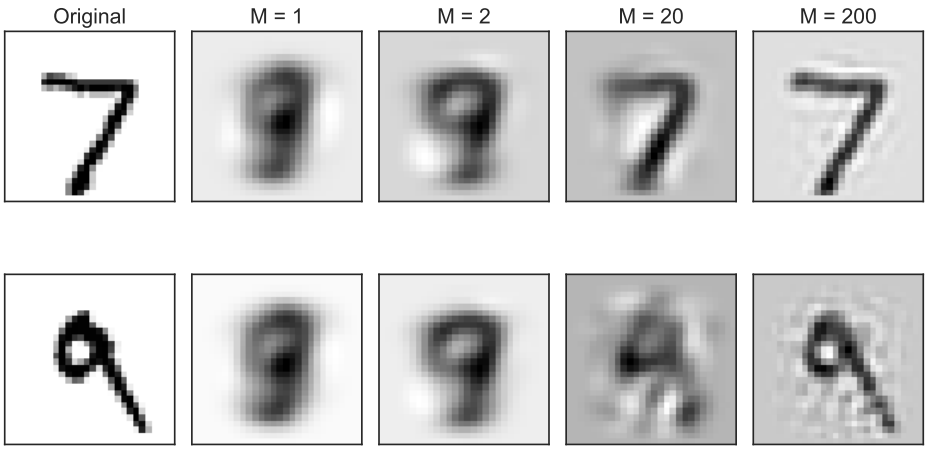




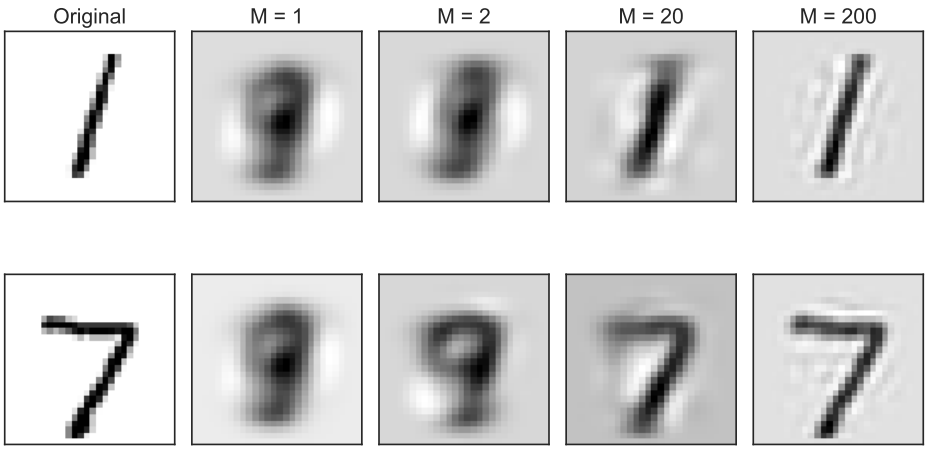
Exemple avec ACP



Reconstruction de caractères : 7 et 9



Reconstruction de caractères : 1 et 7



Caractéristiques de l'ACP

- ACP explique la variance de jeux de données
  - Cependant sensible aux données aberrantes, qui influencent grandement la variance
- Très intéressante pour visualiser des données
- Pour des dimensionnalités élevées (D grand), les calculs sur $\mathbf{S}$ peuvent être lourds ($O(D^2)$)
  - Existe méthodes pour réduire les calculs à un ordre $O(KD)$
- Perte de la signification des variables
  - Construction de variables artificielles correspondant à une combinaison linéaire des variables d'origines

Erreur de reconstruction

- Reconstruction des données
  - Projection dans l'espace de z

$$\mathbf{z}^t = \mathbf{W}^\top(\mathbf{x}^t - \boldsymbol{\mu})$$

  - Comme $\mathbf{W}$ est orthogonal, $\mathbf{W}\mathbf{W}^\top = \mathbf{I}$

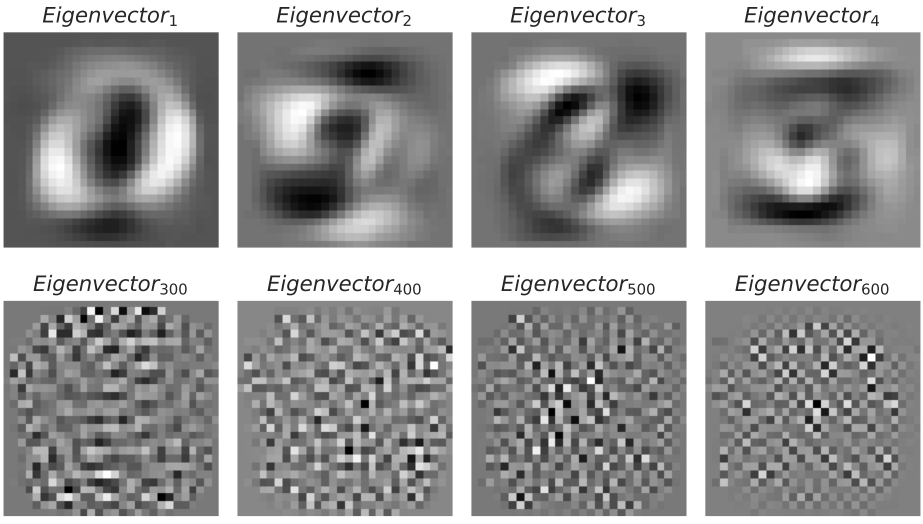
$$\begin{aligned} \mathbf{W}\mathbf{z}^t &= \mathbf{W}\mathbf{W}^\top(\mathbf{x}^t - \boldsymbol{\mu}) \\ \hat{\mathbf{x}}^t &= \mathbf{W}\mathbf{z}^t + \boldsymbol{\mu} \end{aligned}$$

- ACP minimise l'erreur de reconstruction

$$\mathrm{err}_{\mathrm{recon}} = \sum_t \|\hat{\mathbf{x}}^t - \mathbf{x}^t\|^2$$

- Erreur de reconstruction dépend directement du nombre de composantes K utilisées

Eigendigits



12.7 Transformation blanchissante

Transformation blanchissante

- Transformation blanchissante : centrer la moyenne des données sur l'origine, retirer toutes covariances et rendre la variance unitaire

$$\mathbf{x} \sim \mathcal{N}_D(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \overset{\text{blanchir}}{\mapsto} \mathbf{z} \sim \mathcal{N}_D(0, \mathbf{I})$$

  - Transformation linéaire

$$\mathbf{z} = \boldsymbol{\Sigma}^{-0,5}(\mathbf{x} - \boldsymbol{\mu})$$

- Lien fort avec distance de Mahalanobis

$$D_M(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$$

  - Distance de Mahalanobis correspond à distance euclidienne au carré dans un espace blanchi

- Comment calculer $\boldsymbol{\Sigma}^{-0,5}$?

Décomposition spectrale

- $\mathbf{CDC}^\top$ est la décomposition spectrale de $\mathbf{\Sigma}$
- Comme $\mathbf{C}$ est orthogonal et $\mathbf{CC}^\top = \mathbf{C}^\top\mathbf{C} = \mathbf{I}$

$$\begin{aligned} \mathbf{\Sigma} &= \mathbf{CDC}^\top \\ \mathbf{C}^\top\mathbf{\Sigma C} &= \mathbf{C}^\top\mathbf{CDC}^\top\mathbf{C} \\ \mathbf{C}^\top\mathbf{\Sigma C} &= \mathbf{D} \end{aligned}$$

- On sait que $\mathrm{Cov}(\mathbf{z}) = \mathbf{W}^\top\mathbf{\Sigma W}$ et qu'on veut $\mathrm{Cov}(\mathbf{z})$ diagonale
  - On pose donc que $\mathbf{W} = \mathbf{C}$

Décomposition de la matrice de covariance

- Décomposition de la matrice de covariance

$$\mathbf{\Sigma} = \mathbf{W}\mathbf{D}\mathbf{W}^\top$$

- Vecteurs propres de la matrice de covariance

$$\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_D]$$

- Valeurs propres de la matrice de covariance

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_D \end{bmatrix}$$

Racine carrée de la matrice de covariance

- $\mathbf{W}$ est orthogonale, donc $\mathbf{W}^{-1} = \mathbf{W}^\top$
- Développement de $\mathbf{\Sigma}^{0,5}$

$$\begin{aligned} \mathbf{\Sigma} &= \mathbf{W}\mathbf{D}\mathbf{W}^\top = \mathbf{W}\mathbf{D}^{0,5}\mathbf{D}^{0,5}\mathbf{W}^\top \\ &= (\mathbf{W}\mathbf{D}^{0,5}\mathbf{W}^\top)(\mathbf{W}\mathbf{D}^{0,5}\mathbf{W}^\top) = \mathbf{\Sigma}^{0,5}\mathbf{\Sigma}^{0,5} \\ \mathbf{\Sigma}^{-0,5} &= (\mathbf{W}\mathbf{D}^{0,5}\mathbf{W}^\top)^{-1} = \mathbf{W}\mathbf{D}^{-0,5}\mathbf{W}^\top \end{aligned}$$

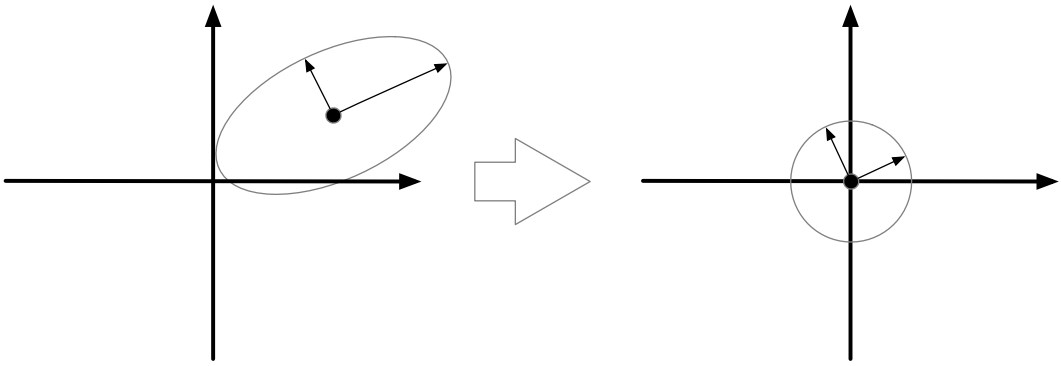
- Matrice $\mathbf{D}$ est diagonale, donc

$$\mathbf{D}^{-0,5} = \begin{bmatrix} \lambda_1^{-0,5} & 0 & \cdots & 0 \\ 0 & \lambda_2^{-0,5} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_D^{-0,5} \end{bmatrix}$$

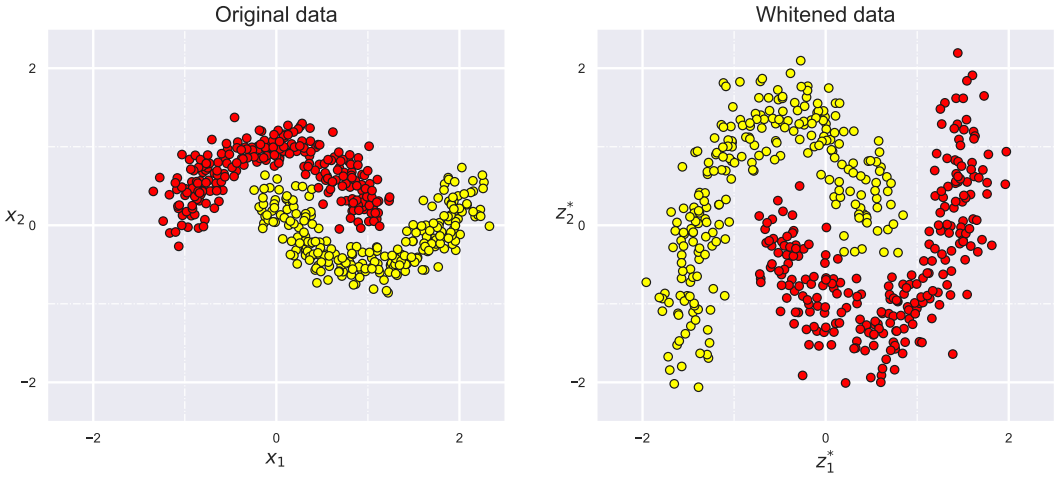
Récapitulatif

$$
\begin{aligned}
\mathbf{x} &\sim \mathcal{N}_D(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
\mathbf{z} &= \boldsymbol{\Sigma}^{-0,5}(\mathbf{x} - \boldsymbol{\mu}) \\
&= \mathbf{W}\mathbf{D}^{-0,5}\mathbf{W}^{\top}(\mathbf{x} - \boldsymbol{\mu}) \\
\text{où } \mathbf{W} &= [\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_D] \\
\text{et } \mathbf{D}^{-0,5} &= \begin{bmatrix} \lambda_1^{-0,5} & 0 & \cdots & 0 \\ 0 & \lambda_2^{-0,5} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_D^{-0,5} \end{bmatrix} \\
\mathbf{z} &\sim \mathcal{N}_D(0, \mathbf{I})
\end{aligned}
$$

Illustration d'une transformation blanchissante



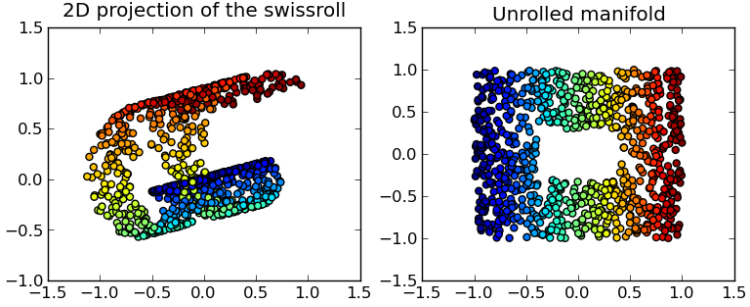
Exemple d'une transformation blanchissante



12.8 Apprentissage de variété

Apprentissage de variété

- Hypothèse d'une présence de variété (*manifold*) : données reposent sur espace non linéaire embarqué dans une espace de plus haute dimension
  - L'apprentissage de la variété vise à extraire cet espace
  - Méthodes non linéaires de réduction de la dimensionnalité
- Exemple du roulé suisse



Par Olivier Grisel, CC-BY 3.0, https://commons.wikimedia.org/wiki/File:Lle_hlle_swissroll.png.

Positionnement multidimensionnel

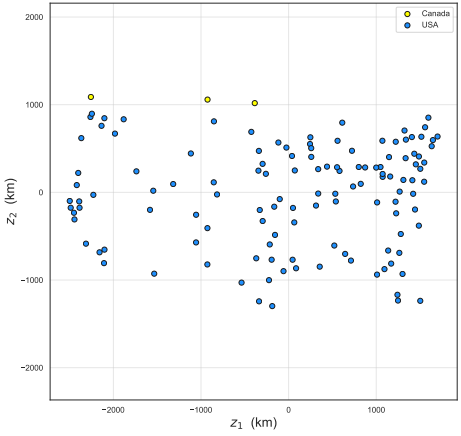
- Positionnement multidimensionnel (*multidimensional scaling*, MDS)
  - Trouver projection vers un espace de plus basse dimensionnalité préservant autant que possible les valeurs de distance $\|\mathbf{x}^i,\mathbf{x}^j\|$ entre toutes les paires de données du jeu $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^N$
- Méthode de Sammon : déterminer projection non linéaire $\mathrm{g}(\mathbf{x}|\theta)$ qui minimise

$$E(\theta|\mathcal{X}) = \sum_{t=1,\ldots,N} \sum_{\substack{s=1,\ldots,N \\ s\neq t}} \frac{(\|\mathrm{g}(\mathbf{x}^t|\theta) - \mathrm{g}(\mathbf{x}^s|\theta)\| - \|\mathbf{x}^t - \mathbf{x}^s\|)^2}{\|\mathbf{x}^t - \mathbf{x}^s\|^2}$$

  - $\theta^* = \text{argmin}_\theta E(\theta|\mathcal{X})$
  - $\mathrm{g}(\mathbf{x}|\theta)$ peut être une régression polynomiale, une régression à noyau, un réseau de neurones, etc.
  - Mesure de distance $\|\cdot\|$ arbitraire, n'a pas à être distance euclidienne

Positionnement multidimensionnel

- Positionner 128 villes nord-américaines seulement à partir des distances routières entre elles



t-SNE (t-distributed Stochastic Neighbour Embedding) 1/2

- Déterminer projection de chaque donnée en basse dimensionnalité en préservant le voisinage de l'espace d'origine
  - En pratique, utile pour visualiser données dans espace 2D ou 3D
- Déterminer probabilité d'être voisins entre les paires du jeu $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^N$ dans l'espace d'origine
  - Probabilité $p_{j|i}$ de sélectionner $\mathbf{x}^j$ comme voisin de $\mathbf{x}^i$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}^i - \mathbf{x}^j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}^i - \mathbf{x}^k\|^2/2\sigma_i^2)}$$

  - Probabilité $p_{i,j} = \frac{p_{i|j}+p_{j|i}}{2N}$ que $\mathbf{x}^j$ soit sélectionné comme voisin de $\mathbf{x}^i$ selon une loi normale centrée sur $\mathbf{x}^i$ ($p_{i,i} = 0$)
  - σ_i^2 est ajusté localement pour chaque donnée (méthode de bissection)

t-SNE (t-distributed Stochastic Neighbour Embedding) 2/2

- Déterminer probabilité d'être voisin entre paires du jeu dans espace de basse dimensionnalité
  - $\mathbf{z}^t$ est la projection de $\mathbf{x}^t$ dans l'espace basse dimensionnalité
  - Probabilité $q_{i,j}$ supposant une loi de Student

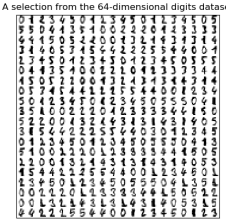
$$q_{i,j} = \frac{(1 - \|\mathbf{z}^i - \mathbf{z}^j\|^2)^{-1}}{\sum_{\substack{k=1,\dots,N \\ k \neq i}} (1 - \|\mathbf{z}^i - \mathbf{z}^k\|^2)^{-1}}$$

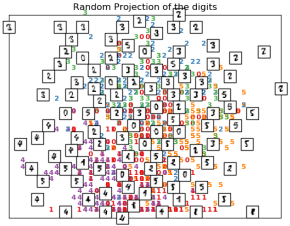
- Apprendre projections $\mathbf{z} = \mathrm{g}(\mathbf{x}|\theta)$ des points en basse dimensionnalité afin de minimiser divergence entre ces probabilités

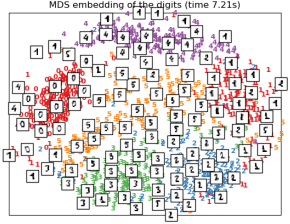
$$E(\theta|\mathcal{X}) = KL(P\|Q) = \sum_{t=1,\dots,N} \sum_{\substack{k=1,\dots,N \\ k \neq t}} p_{t,k} \log \frac{p_{t,k}}{q_{t,k}|\theta}$$

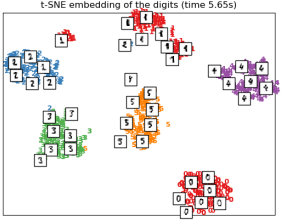
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E(\theta|\mathcal{X})$$

Comparaison d'apprentissage de variétés









Source : `https://scikit-learn.org/stable/auto_examples/manifold/plot_lle_digits.html`, accédé le 26 novembre 2018, code sous licence BSD 3.

12.9 Prétraitement et analyse de données avec scikit-learn

Scikit-learn : ajustements, standardisation et imputation

- Ajustement d'échelle et standardisation
  - `preprocessing.MinMaxScaler` : ajuster l'échelle selon valeurs minimales/maximales
  - `preprocessing.scale` : standardisation pour que variables suivent loi normale centrée-réduite
- Imputation
  - `impute.SimpleImputer` : imputer valeurs à une valeur fixe pour chaque variable
    - `strategy` : stratégie utilisée pour l'imputation simple, soit valeur moyenne (`mean`), valeur médiane (`median`), valeur plus fréquente (`most_frequent`) ou une constante (`constant`)
  - `impute.MissingIndicator` : obtenir masque indiquant variables manquantes d'un jeu de données

Scikit-learn : sélection de caractéristiques

- Sélection univariée
  - `feature_selection.VarianceThreshold` : sélectionner caractéristiques avec variance supérieure à un seuil
  - `feature_selection.SelectKBest` (`SelectPercentile`) : conserve les K meilleures (percentile supérieur) caractéristiques selon une mesure de performance
    - `chi2` : test χ^2 entre caractéristiques
    - `f_classif` : test ANOVA entre caractéristiques
    - `mutual_info_classif` : critère d'information mutuelle
- `feature_selection.RFE` : sélection arrière selon coefficients du modèle
  - `estimator` (objet) : modèle d'apprentissage utilisé pour la sélection
  - `n_features_to_select` (int) : nombre total de caractéristiques à sélectionner
  - `step` (int ou float)
    - Si ≥ 1, nombre de caractéristiques retirées à chaque itération
    - Si [0,1), ratio du nombre de caractéristiques retirées à chaque itération
- `feature_selection.SelectFromModel` : sélection à partir d'un modèle (ex. selon les coefficients)

Scikit-learn : analyse en composantes principales

- `decomposition.PCA` : analyse en composantes principales
  - Paramètres
    - `n_components` (int) : nombre de composantes à conserver, par défaut $K = \min(N, D)$
    - `whiten` (bool) : normalise par les vecteurs propres, effectuant ainsi une transformation blanchissante
  - Attributs
    - `components_` (array) : vecteurs des composantes principales (taille $K \times D$)
    - `explained_variance_` (array) : variance expliquée par chaque composante (vecteur de taille K)
    - `explained_variance_ratio_` (array) : proportion de la variance expliquée par chaque composante (vecteur de taille K)

Scikit-learn : apprentissage de variété

- `manifold.MDS` : positionnement multidimensionnel
  - `n_components` (int) : dimensionnalité de l'espace destination
  - `metric` (bool) : métrique ou non
  - `dissimilarity` : mesure de distance, soit `euclidean` (défaut) ou `precomputed`
- `manifold.TSNE` : t-SNE
  - `n_components` (int) : dimensionnalité de l'espace destination
  - `perplexity` (float) : lié au nombre de voisins utilisé (défaut : 30)
- Autres algorithmes d'apprentissage de variété non linéaires
  - `manifold.Isomap` : algorithme Isomap
  - `manifold.LocallyLinearEmbedding` : algorithme LLE
  - `manifold.SpectralEmbedding` : algorithme Laplacian eigenmaps