IA probabiliste et bioinspirée
Reconnaissance des formes
Apprentissage machine

GUIDE DE L'ÉTUDIANTE ET DE L'ÉTUDIANT
S8GIA – APP2

Module: Intelligence artificielle
Hiver 2024

Département de génie électrique et de génie informatique
Faculté de génie
Université de Sherbrooke

Historique des modifications

Date	Responsables	Description
Automne 2023	JBM	Code de départ en classes. Nouvelles AP. Nouvelle structure labos et procéduraux.
Automne 2022	JBM	Importé dans nouveau gabarit le matériel existant Jean Rouat et coll. des années et APs antérieures (GEI790, GEI792) avec modifications pour passer aux nouvelles AP.

Auteur: JBM et coll.
Version: 0.4 (9 février 2024 10:50)

Ce document est réalisé avec l'aide de LaTeX et de la classe `gegi-app-guide`.

TABLE DES MATIÈRES

1 ACTIVITÉS PÉDAGOGIQUES ET COMPÉTENCES

GEI890 – Préparation de données pour systèmes intelligents

1. Analyser un jeu de données et sélectionner des représentations appropriées pour une application spécifique et une technique d'intelligence artificielle donnée.
2. Appliquer des techniques de préparation de données formelles.

Description officielle : https://www.usherbrooke.ca/admission/fiches-cours/GEI890/

GEI895 – Conception de systèmes intelligents

1. Choisir une technique de l'intelligence artificielle en fonction de spécifications descriptives pour une application donnée.
2. Concevoir des systèmes intelligents utilisant des techniques appropriées de l'intelligence artificielle.
3. Mettre en oeuvre et valider les systèmes intelligents conçus avec les outils appropriés

Description officielle : https://www.usherbrooke.ca/admission/fiches-cours/GEI895/

Ces activités remplacent GEI790, GEI791 et GEI792.

2 SYNTHÈSE DE L'ÉVALUATION

Évaluation	GEI890-1	GEI890-2	GEI895-1	GEI895-2	GEI895-3
Défense d'APP	60	60	40	75	40
Évaluation sommative théorique	45	45	65	135	65
Évaluation finale théorique	15	15	45	10	15
Évaluation finale pratique	30	30		80	30
Total	150	150	150	300	150

TABLEAU 2.1 Synthèse de l'évaluation de l'unité

Voici les cotes utilisées pour les deux activités pédagogiques concernées.

Note(%)	<50	50	54	58	62	66	70	74	78	82	86	90
Cote	E	D	D+	C-	C	C+	B-	B	B+	A-	A	A+
Niveau	N0	N1	N1	N1	N2	N2	N2	N3	N3	N3	N4	N4
Libellé	Insuffisant	Passable (seuil)			Bien			Très bien (cible)			Excellent	

3 QUALITÉS DE L'INGÉNIEUR

Les qualités de l'ingénieur visées et évaluées par cette unité d'APP sont données dans le tableau un peu plus bas. D'autres qualités peuvent être présentes sans être visées ou évaluées dans cette unité. Pour une description détaillée des qualités et leur provenance, consultez le lien suivant: qualités et BCAPG

Qualité	Libellé	Touchée	Évaluée
Q01	Connaissances en génie	✓	✓
Q02	Analyse de problèmes	✓	✓
Q03	Investigation		
Q04	Conception	✓	✓
Q05	Utilisation d'outils d'ingénierie	✓	✓
Q06	Travail individuel et en équipe		
Q07	Communication	✓	
Q08	Professionnalisme		
Q09	Impact du génie sur la société et l'environnement		
Q10	Déontologie et équité		
Q11	Économie et gestion de projets		
Q12	Apprentissage continu	✓	

4 ÉNONCÉ DE LA PROBLÉMATIQUE

Analyse et classification d'images

Vous oeuvrez dans le domaine de la vision intelligente, avec comme mandat de classifier l'environnement d'un véhicule autonome pour le système de conduite de celui-ci. Vous cherchez à comparer différentes techniques de reconnaissance statistique des formes, leur performance pour cette application de vision, leurs avantages et inconvénients. Vous défendrez une synthèse de vos travaux sous forme d'une présentation orale théorique et pratique.

Les images à classer comprennent trois catégories d'environnements à reconnaître : des plages, des environnements urbains ("villes") et des forêts. Comme pour toute application d'apprentissage machine, vous devrez d'abord développer une représentation du problème, prétraiter cette représentation, puis appliquer les techniques de classification pertinentes et en critiquer la performance.

En apprentissage machine, le choix de la représentation est une étape cruciale qui limite souvent la discrimination atteingnable entre les classes. Deux critères servent généralement à juger de la qualité de la représentation : sa dimensionalité et la séparabilité des classes dans l'espace mathématique résultant. Dans notre cas, on propose les recommandations et directives suivantes pour la représentation :

- Investiguer si une normalisation ou des calculs peuvent diminuer la sensibilité aux variabilités non discriminantes entre les images, avec l'objectif que des images similaires (e.g. même sujet en plein soleil, au crépuscule, etc.) aboutissent à une représentation proche.
- Évaluer la pertinence de créer des sous-classes pour rendre les clusters convexes dans l'espace latent et ainsi améliorer la séparabilité pour les algorithmes étudiés.
- Pour limiter la dimensionalité, on n'utilisera pas les pixels bruts (i.e. l'image elle-même en tout ou en partie) comme représentation et donc comme entrée(s) des classificateurs, mais plutôt des quantités qui tiennent sur quelques nombres entiers ou réels et calculées à partir de l'image. À titre d'exemples, quelques pistes à investiguer :
  - Nombre de pixels de telle ou telle intensité ou couleur dans l'image, incluant une conversion des espaces de couleur (HSV, Lab, XYZ, CMYK, etc.) si bénéfique à la séparabilité.
  - Quantité représentant la distribution spatiale des couleurs (e.g. nombre de pixels bleus dans le coin supérieur gauche).
  - Statistiques (écarts-types, médianes, moyennes, asymétries des distributions, etc.) des nombres obtenus par les calculs précédents.

— Puisque l'objectif de l'APP est de créer soi-même une représentation, on n'utilisera pas d'apprentissage profond, ni d'algorithmes déjà implémentés dans des librairies de vision comme OpenCV, mais il n'est pas exclu de reproduire des principes sous-jacents simples. À titre d'exemples, on pourrait détecter une forme standard par corrélation avec une droite, cercle, etc., faire du *edge detection* simple, ou déployer tout calcul mathématique ou statistique simple qui facilite la détection d'une caractéristique de l'image.

La prochaine étape du processus de conception d'un algorithme d'apprentissage machine, le prétraitement, aide à comprendre les paramètres présélectionnés comme représentation. Le but est de ne conserver que les dimensions les plus discriminantes pour le classificateur avec une analyse des composantes principales.

— Pour chaque classe ou sous-classe, on étudie différentes quantités statistiques de la représentation : moyenne, matrice de covariance, etc., et on les décompose en leurs composantes principales, i.e. leurs valeurs et vecteurs propres. On visualise, dans la mesure du possible, la dispersion et la répartition des points dans l'espace, quite à réduire temporairement le nombre de paramètres pour faciliter la visualisation.

— Cette analyse mène, pour l'ensemble des classes, au choix d'une opération qui décorrèle le plus possible l'espace mathématique de la représentation, pour en réduire le plus possible la dimension. En d'autres mots, on sélectionnera les quelques (max 10) paramètres les mieux décorrélés mais surtout les plus discriminants possible.

Le prétraitement se conclut habituellement par la visualisation de la représentation finale obtenue, et par le partitionnement des données en ensembles d'entraînement et de validation.

Enfin, les techniques d'intelligence artificielle ! On implémentera trois classificateurs, soit un bayésien, un non-paramétrique par quantification vectorielle, et un réseau de neurones.

— Pour le classificateur bayésien, on modélisera dans un premier temps les classes ou sous-classes au moyen d'un modèle gaussien, et par la suite d'une densité de probabilité arbitraire. Un tel classificateur minimise le risque de Bayes, i.e. maximise la vraisemblance a posteriori, pour chacun des modèles. On choisira des coûts appropriés, le cas échéant, selon l'impact anticipé d'une mauvaise décision dans la matrice de confusion.

— Pour le classificateur non paramétrique, on choisira un certain nombre de représentants de (sous-)classe à l'aide de l'algorithme des k-moyennes, puis on concevra un classificateur basé sur l'algorithme des k-plus proches voisins (k-PPV).

— Pour le classificateur par réseau de neurones, on choisira judicieusement les hyperparamètres (nombre de couches, nombre de neurones, fonction d'activation, normalisation des entrées/sorties, etc.) d'un réseau formel simple complètement connecté limité à

une centaine de neurones (maximum !), et ceux de l'entraînement supervisé adéquat (nombre d'époch, algorithme d'optimisation, taux d'apprentissage, loss, critère d'arrêt, etc.).

Pour chaque classificateur, il est d'usage de comprendre le mieux possible les frontières de décision obtenues dans la représentation choisie, d'évaluer la performance de chaque algorithme au moyen, au minimum, d'une matrice de confusion, et d'en comparer les performances. Un estimé de la complexité computationnelle de chacun est aussi souvent très pertinent.

Dans tout processus de conception d'apprentissage machine, notre compréhension des données et des algorithmes évolue souvent en cours de route. Il est habituel d'itérer, au besoin, sur le choix de la représentation et sur la conception des classificateurs, en justifiant les modifications apportées selon les comportements observés, plutôt que de procéder par essai et erreur. En particulier, on devra probablement peaufiner la représentation ou les hyperparamètres pour discriminer les (sous-)classes les plus en erreur. Il est assez facile, avec un choix minimal de la représentation et en utilisant les techniques proposées sans trop les peaufiner, d'atteindre un taux de bonne classification de 60-70%. Atteindre un taux de 80 à 90% est le niveau cible ici. Un taux supérieur à 90% démontre une compréhension approfondie des choix de la représentation et de la mise en oeuvre des différentes techniques de classification.

Cette étude comparera enfin la performance des classificateurs entre eux, leurs forces et faiblesses, et leur facilité d'application pour la conduite autonome, selon des métriques appropriées.

L'ensemble du processus de conception doit bien entendu respecter les règles de l'art et les bonnes pratiques en ingénierie. Il est fortement conseillé de déployer un mécanisme de gestion des révisions pour comparer aisément les résultats entre différents choix de conception. Pour les algorithmes non déterministes, le réseau de neurones pour ne pas le nommer, une stratégie de sauvegarde des meilleurs modèles et de métadonnées (hyperparamètres utilisés) associées à chacun est aussi indispensable.

5 CONNAISSANCES NOUVELLES

Connaissances déclaratives (quoi)

- Analyse de données
  - Représentation de l'information.
  - Extraction des caractéristiques discriminantes d'un jeu de données.
  - Nettoyage des données.
  - Normalisation et étiquetages.
- Classification probabiliste
  - Lois de probabilité gaussiennes à dimensions multiples.
  - Paramétrisation d'une loi de probabilité gaussienne.
  - Décorrélation de l'espace de représentation.
  - Classification Bayésienne.
  - Classification par les k plus proches voisins.
  - Classification par le k-moyennes.
  - Mesure de la similitude, notions de coût et d'erreur.
  - Apprentissage supervisé et non supervisé.
- Réseau de neurones
  - Fonction d'activation.
  - Poids et biais.
  - Réseaux formels multicouches.
  - Entraînement par propagation arrière.
  - Ensembles d'apprentissage, de validation et de test.

Connaissances procédurales (comment)

- Concevoir un système intelligent basé sur des techniques de l'intelligence artificielle.
- Mettre en oeuvre un système intelligent basé sur des techniques d'intelligence artificielle.
- Analyser un jeu de données pour en ressortir les caractéristiques principales.
- Appliquer des techniques de nettoyage et de normalisation des données.
- Utiliser les algorithmes de classification probabilistes.
- Utiliser les réseaux de neurones simples.
- Identifier l'emplacement de la connaissance dans le modèle intelligent.
- Valider les performances d'un système intelligent.

Connaissances conditionnelles (quand)

— Effectuer un prétraitement des données pour en extraire les caractéristiques.
— Effectuer un étiquetage des données.
— Choisir les représentations de données appropriées pour une application spécifique.
— Adapter les représentations de données pour une technique d'intelligence artificielle donnée.

6 GUIDE DE LECTURE

Tout le matériel proposé est disponible sous forme électronique sur le site web.

Révision et réactivation de connaissances antérieures

1. Jean Rouat, Extraits de Notes de cours, *Intelligence artificielle probabiliste*
   — Chapitre 2, sections 2.1, 2.2 et 2.5 (probabilités et statistiques)
   — Chapitre 4 jusqu'à 4.5 inclusivement (algèbre matricielle et probabilités discrètes)
2. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 2, sections 2.7 et 2.11

Lectures obligatoires

Tutorat d'ouverture : Introduction à la classification, à l'apprentissage machine et au traitement des données

1. Jean Rouat, Extraits de Notes de cours, *Intelligence artificielle probabiliste*
   — Introduction (mise en contexte)
2. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 5 jusqu'à 5.1 incl. (Machine learning basics)

Laboratoire 1 : Prétraitement et représentation

1. Jean Rouat, Extraits de Notes de cours, *Intelligence artificielle probabiliste*
   — Le reste du chapitre 2 (densités de probabilité)
   — Chapitre 8 (composantes principales) et le reste du chapitres 4 (réduction de dimension)
2. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 2, section 2.12
   — Chapitre 5, section 5.8.1

Procédural 1 : Classification par plus proches voisins

1. Extraits de Moreira *et al.*, *General Introduction to Data Analytics*
   — Chapitre 9, de 9.1 à 9.3.1 incl. (classification par les plus proches voisins)

— Chapitre 5 jusqu'à 5.3.2 incl. (clustering)

2. Jean Rouat, Extraits de Notes de cours, *Intelligence artificielle probabiliste*
   — Chapitres 7 (k-PPV) et 9 (k-moyennes)
3. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 5, section 5.8.2

Procédural 2 : Réseaux de neurones formels multicouches

Note : Pour les réseaux de neurones, vous n'avez pas besoin de vous attarder à d'autres mathématiques que celles que nous couvrirons au laboratoire et pendant le procédural. Cependant vous devez comprendre les principes sous-jacents et les conséquences des différents choix de conception, d'hyperparamètres et d'algorithmes. Adaptez votre lecture en conséquence.

Matériel d'introduction

1. Extraits de Haykin 3rd ed., *Neural Networks and Learning Machines*
   — Chapitre "Introduction"
2. Extraits de Moreira *et al.*, *General Introduction to Data Analytics*
   — Chapitre 10, section 10.2.1
3. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 6, section 6.6 (Historical notes)

Réseaux multicouches, lectures générales

1. Extraits de Haykin 3rd ed., *Neural Networks and Learning Machines*
   — Chapitre 4, sections 4.1 et 4.2, 4.5, 4.7, 4.11, 4.13, 4.15
2. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 6, jusqu'à 6.4 incl.

Apprentissage

1. Extraits de Haykin 3rd ed., *Neural Networks and Learning Machines*
   — Chapitre 4, sections 4.3 et 4.4, 4.8, 4.10, 4.13, 4.15
2. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Chapitre 4, sections 4.3 et 4.4
   — Chapitre 5, sections 5.2, 5.3 et 5.9

— Chapitre 6, section 6.5 (rétropropagation et différentiation)
— Chapitre 7, section 7.8 (early stopping)
— Chapitre 8, sections 8.3 à 8.6 incl.
— Chapitre 8, sections 8.1.3 (batch size) et 8.2 (problèmes d'apprentissage)

Autres tutoriels et synthèses spécifiques

1. Courte synthèse des différents comportements lors de l'apprentissage
2. Résumé de l'entraînement, validation et test

Procédural 3 : Classification Bayésienne

Jean Rouat, Extraits de Notes de cours, *Intelligence artificielle probabiliste*

— Chapitres 5 (classification bayésienne) et 6 (frontières explicites entre classes)

Lecture optionelle complémentaire

1. Extraits de Haykin 3rd ed., *Neural Networks and Learning Machines*
   — Le reste du chapitre 4
2. Extraits de Goodfellow *et al.*, *Deep Learning*
   — Le reste du chapitre 2
   — Le reste du chapitre 5
   — Le reste du chapitre 8

7 LOGICIELS ET MATÉRIEL

On utilisera Python 3.11 pour résoudre cet APP. Le code de départ des laboratoires et de la problématique se trouve sur le site web de l'APP. Le fichier `requirements.txt` indique les versions des librairies avec lesquelles le code a été développé. Nous recommandons l'environnement `PyCharm` ou `VSCode` puisqu'ils sont installés sur les ordinateurs de la Faculté et que vous y aurez accès à l'examen final.

Installez les librairies requises dans votre environnement virtuel au moyen de la commande

```
pip install -r requirements.txt
```

De plus, dans Python 3.11, il est recommandé de configurer le lancement automatique du déverminage (debugger) avec l'option `-Xfrozen_modules=off` pour accélérer le démarrage du script et désactiver plusieurs messages d'erreur liés à cette version.

Le code de départ du laboratoire indique quel trou est à compléter ou modifier dans quel exercice. Faites une recherche pour `TODO Labo` dans les différents fichiers. Différentes sections du code peuvent être executées ou au contraire déactivées, cherchez `if True` dans le fichier principal. De la même manière `TODO Problématique` suggère des changements pertinents à sa solution.

Il n'est pas obligatoire de solutionner la problématique avec le code fourni, mais notez que le code à trous fourni à l'examen final sera très similaire au code de départ.

8 SANTÉ ET SÉCURITÉ

8.1 Dispositions générales

Dans le cadre de la présente activité, vous êtes réputés avoir pris connaissance des politiques et directives concernant la santé et la sécurité. Ces documents sont disponibles sur les sites web de l'Université de Sherbrooke, de la Faculté de génie et du département. Les principaux sont mentionnés ici et sont disponibles dans la section *Santé et sécurité* du site web du département.

— Politique 2500-004: Politique de santé et sécurité en milieu de travail et d'études

— Directive 2600-042: Directive relative à la santé et à la sécurité en milieu de travail et d'études

— Sécurité en laboratoire et atelier au département de génie électrique et de génie informatique

8.2 Dispositions particulières

La solution de l'APP n'utilise que des éléments logiciels, et ne requiert donc aucune considération particulière de sécurité autre que celles liées à l'utilisation habituelle des laboratoires informatiques.

9 SOMMAIRE DES ACTIVITÉS

Cet APP de 3 crédits comportera 3 laboratoires et 3 procéduraux.

Une défense orale et pratique de la solution à la problématique remplace le rapport.

Un livrable formatif, assorti d'une consultation pour en discuter, facilite la compréhension progressive de la solution à la problématique.

10 PRODUCTIONS À REMETTRE

- Les productions se font par équipe de 3, sauf lorsque indiqué autrement.
- L'identification des membres des équipes doit être faite sur la page web de l'unité avant le premier laboratoire.
- La date limite pour le dépôt électronique est 1h avant le début de la première défense, le jour de la défense. Les retards seront pénalisés.
- Les productions soumises à l'évaluation doivent être originales pour chaque équipe, sinon l'évaluation sera pénalisée en cas de non-respect de cette consigne.
- L'évaluation formative ne sera pas explicitement évaluée, mais par souci d'équité lors de la consultation, une pénalité dégressive sera appliquée à la note de la défense si rien n'est déposé comme livrable formatif, ou si le dépôt ne témoigne pas d'un effort suffisant dans la résolution de la problématique, dans les limites de la compréhension qu'a chacun des concepts de l'APP.

10.1 Schéma de concept

Le schéma de concept est une production individuelle optionnelle en vue de la deuxième rencontre de tutorat. Le schéma de concepts à faire lors de l'étude personnelle cible la question suivante :

> *Qu'est-ce qu'un classificateur et comment le met-on en oeuvre ?*

10.2 Défense de la solution

10.2.1 Dépôt électronique

Au plus tard 1h avant le début de la défense, i.e. avant la défense de la première équipe à l'horaire, tout code informatique produit ou modifié pendant l'APP ainsi que la présentation servant à la défense doivent être déposés sur le site habituel du département. **Tout retard non motivé sera pénalisé.** La date de dépôt sur le serveur est la seule date certifiant la remise du travail à temps.

- **Ne rien envoyer par courriel.**
- Rédigez un "readme" qui guide le correcteur et résume la structure de votre dépôt et de votre code.
- Le code doit être structuré de façon à ce que les évaluateurs **puissent le faire fonctionner** sans devoir configurer de 'path' particulier et sans devoir déplacer de fichiers. Les fichiers de données adéquats doivent donc être présents aux endroits adéquats.

10.2.2 Défense de la solution

Il n'y a pas de rapport à produire. Vous défendrez votre solution à la problématique lors d'une présentation orale. Chaque équipe disposera d'un maximum de 25 minutes pour présenter sa méthodologie, la justifier, et discuter des résultats. Il est attendu que chaque étudiante ou étudiant résume (15 secondes ou moins) à l'équipe professorale à tour de rôle sa contribution à la solution. **La qualité de la présentation (e.g. lisibilité, figures avec légendes, etc.) est explicitement évaluée.**

L'équipe présentera dans une autre séance de 20 minutes le code informatique final et fera la démonstration de son fonctionnement à l'auxiliaire d'enseignement.

Les équipes sont convoquées à une heure précisée ultérieurement sur le site de l'APP. La présence de tous est obligatoire. Vérifiez l'heure de passage de votre équipe.

10.2.3 Deuxième séance de tutorat

Afin d'alimenter la discussion et les échanges au cours du tutorat 2, assurez-vous d'avoir sous la main sous une forme facilement partageable les tableaux et les graphiques de votre solution à la problématique.

10.3 Informations et consignes pour la défense de la solution

Cette section liste quelques recommandations pour la structure et le contenu de la présentation. On devrait y retrouver les éléments suivants.

10.3.1 Justification

A-t-on déjà mentionné que la qualité des justifications de vos choix est l'indicateur le mieux corrélé à la qualité de votre compréhension ? **Toujours justifier l'approche et les choix en fonctions des objectifs, de l'application visée et des données particulières.** Le pourquoi est plus important pour la défense que le quoi et le comment, à moins que ces derniers ne sortent de l'ordinaire. **Évitez de reproduire des évidences, des concepts généraux, des éléments contenus dans les références ou les notes de cours, etc.** Présentez plutôt comment vous avez modifié et appliqué tout ça dans votre solution et pourquoi.

10.3.2 Description des travaux réalisés

Vous pourriez présenter :

— des schémas blocs ;

— des figures ;

— les équations pertinentes qui ne sont pas une répétition de théorie généralement connue.

Présentez du texte, des équations et des graphiques qui se complètent judicieusement. Définir les variables lorsque nécessaire.

10.3.3 Conditions expérimentales exactes précises pour chaque résultat

On devrait être en mesure de reproduire vos résultats ! **Décrivez les conditions précises (hyperparamètres, etc.)** dans lesquelles chaque résultat est obtenu. Donnez les valeurs des paramètres utilisés, sous forme d'un tableau à côté de la figure, par exemple.

10.3.4 Résultats

Les figures et les tableaux doivent être assortis d'une légende et des interprétations importantes.

10.3.5 Discussion

Le deuxième indicateur le plus corrélé à votre compréhension. Expliquez pourquoi vous obtenez ces résultats en fonction des différents choix, discutez de la signification de ces résultats en regard des objectifs du problème. **Encore une fois, discuter et commenter abondamment est préférable à énumérer des faits ou des procédures.** Décrivez les aspects importants de vos figures et expliquez leur signification.

10.3.6 Commentaires, recommandations et conclusion

Relever les points important de votre discussion, formuler des recommendations sur le travail réalisé, son potentiel et les applications futures.

10.3.7 Références

— Tout plagiat [1] (une attention particulière sera apportée au code source fourni et aux résultats) sera sanctionné. En cas de plagiat, la note du problème sera divisée par le nombre de personnes et d'équipes impliquées. Les points appropriés iront à chatGPT, le cas échéant.

— Ce qui précède n'a pas pour but de décourager la communication et la discussion entre les équipes, mais bien de prévenir la réutilisation sans citation de matériel et de solutions "fortement inspirées" d'autres sources ou d'exemples disponibles en ligne.

1. Toute copie non autorisée de matériel qui n'est pas de notoriété publique sans citer les sources d'information.

— En effet, vous pouvez utiliser à votre guise toute documentation autre que celle fournie ou suggérée. Citez simplement vos sources d'inspiration avec une référence appropriée, en particulier les exemples et codes trouvés sur le web.
— Évidemment, certains résultats et observations techniques seront forcément communs à l'ensemble de la classe.

10.4 Livrable formatif

10.4.1 Objectifs du livrable formatif

Le livrable formatif vise, d'une part, à faire progresser la compréhension de la problématique de manière à ce que les concepts importants soient appliqués et révisés à un rythme qui garantit l'assimilation des connaissances à temps pour livrer une solution. D'autre part, il vise à discuter constructivement et à itérer si nécessaire au moment le plus opportun de différents éléments de solution qui influent grandement sur la performance finale.

10.4.2 Dépôt électronique

Le livrable formatif doit être déposé électroniquement selon les directives indiquées sur le site web au moment opportun **une heure avant la consultation qui l'accompagne**. Une absence de dépôt, ou une progression insuffisante dans la résolution de la problématique, pourront conduire à une pénalité dégressive appliquée à la défense.

10.4.3 Format des livrables

Il n'y a pas de format imposé pour le dépôt, tout format permettant de montrer le progrès réalisé et de justifier les choix est adéquat. Par exemple, une équipe pourrait déposer un journal de bord de conception, un mini-rapport, une mini-présentation, des fichiers excel, du code, divers figures montrant des résultats intermédiaires, etc.

10.4.4 Consultation formative

Le contenu et les objectifs du livrable seront discuté le cas échéant en grand groupe lors d'une consultation, où les équipes pourront présenter si elles le désirent leur progrès et poser des questions par rapport à leur compréhension. L'équipe professorale orientera aussi au besoin les discussions formatives sur des éléments essentiels qui semblent avoir ou non été assimilés ou incorporés adéquatement dans les progrès discutés.

10.4.5 Livrable formatif 1 – Élaboration de la représentation et choix préliminaires de conception des 3 classificateurs

Le livrable vise à montrer et expliquer les choix préliminaires de la représentation des images. Le livrable devrait énumérer les quantités mathématiques retenues à cette date, les visualiser et discuter de leur potentiel discriminant pour la classsification subséquente. Le livrable devrait aussi montrer la préconception, i.e. les choix préliminaires des différents paramètres configurables des 3 classsificateurs, justifiés en fonction de la représentation et de la sépara-

bilité anticipée. Le livrable devrait montrer pourquoi les choix d'hyperparamètres semblent appropriés en fonction de la représentation choisie. Optionnellement, le livrable pourrait démontrer une première version de code fonctionnel, i.e. qui obtient un premier résultat préliminaire de classification.

11 ÉVALUATIONS

11.1 Défense et livrables associés

L'évaluation de la défense portera sur les compétences figurant dans la description des activités pédagogiques. Ces compétences ainsi que la pondération de chacune d'entre elles dans l'évaluation de la défense sont indiquées au tableau 11.1. L'équipe professorale évalue votre compétence à implémenter un processus de conception complet et rigoureux des techniques couvertes. Une simple mise en oeuvre (simplement coder et exécuter une technique) est considéré en-dessous du niveau cible ici.

Élément évalué	GEI890-1	GEI890-2	GEI895-1	GEI895-2	GEI895-3
Représentation	25	5	5	5	5
Prétraitement	5	10	5	5	5
Classificateur bayésien	5	10	5	15	5
Classificateur k-moy, k-PPV	5	10	5	15	5
Classificateur RNA	5	10	5	15	5
Discussion et justifications	10	10	15	15	10
Code	5	5		5	5
Total	60	60	40	75	40

TABLEAU 11.1 Sommaire de l'évaluation de la défense

La qualité de la communication, de la présentation et du code sont expressément évalués (10% dégressif dans chaque élément). La qualité insuffisante du livrable formatif pourrait aussi conduire à 10% dégressif dans les éléments visés.

11.2 Évaluation sommative

L'évaluation sommative théorique est un examen écrit qui porte sur tous les éléments de compétences de l'unité, y compris l'interprétation de résultats pratiques de mise en oeuvre des classificateurs proposés. L'évaluation sommative ne comporte pas formellement d'éléments pratique (code à produire ou modifier). C'est un examen sans documentation permise.

11.3 Évaluation finale

L'évaluation finale est un examen écrit et pratique qui porte sur tous les éléments de compétences de l'unité, où vous devrez mettre en oeuvre l'un ou l'autre ou l'ensemble des classificateurs proposés dans l'unité à partir d'un code à compléter semblable au code de départ des laboratoires. Vous aurez accès à l'ensemble du matériel de lecture et de référence sous forme électronique, et **aux logiciels installés sur les ordinateurs de la faculté.**

12 POLITIQUES ET RÈGLEMENTS

Dans le cadre de la présente activité, vous êtes réputés avoir pris connaissance des politiques, règlements et normes d'agrément ci-dessous:

Règlements de l'Université de Sherbrooke

- Règlement des études

Règlements facultaires

- Règlement facultaire d'évaluation des apprentissages / Programmes de baccalauréat
- Règlement facultaire sur la reconnaissance des acquis

Normes d'agrément

- Processus d'agrément et qualités du BCAPG
- Ingénieurs Canada – Àpropos de l'agrément

Enfin, si vous êtes en situation de handicap, assurez-vous d'avoir communiqué avec le *Programme d'intégration des étudiantes et étudiants en situation de handicap* à l'adresse: prog.integration@usherbrooke.ca.

13 INTÉGRITÉ, PLAGIAT ET AUTRES DÉLITS

Dans le cadre de la présente activité, vous êtes réputés avoir pris connaissance de la page Intégrité intellectuelle des Services à la vie étudiante.

14 PRATIQUE EN LABORATOIRE 1

Buts de l'activité

Le but de cette activité est de se familiariser avec ...

- l'analyse statistique élémentaire de distributions de classes
- l'analyse des composantes principales d'une distribution
- la décorrélation de l'espace mathématique d'une distribution
- des pistes de base pour investiguer la représentation de la problématique

Notes

- Consultez les conseils à la section "Matériel et logiciels".
- Solutionnez par vous-mêmes les exercices préparatoires si vos connaissances antérieures en statistiques requièrent une révision.

14.1 EXERCICES PRÉPARATOIRES

L1.P1 Génération et estimation de distributions statistiques

Objectifs :

- Être capable de manipuler les estimateurs élémentaires ;
- Comprendre l'importance de la taille de l'échantillonnage et son influence.

Réaliser la séquence de traitements donnée ci-dessous et commentez vos obervations. Vous aurez à programmer certains estimateurs et à observer leur dépendance vis-à-vis du nombre de points échantillonnés.

1. Générer de façon aléatoire les échantillons d'une distribution normale (i.e. gaussienne) spécifiée par :

$$n = 2 \text{ (dimensions)}, N = 20 \text{ (nombre de vecteurs)}$$
$$m = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Solution Utiliser `numpy.random.multivariate_normal()`.

2. Visualisez les points ainsi générés.

3. Calculer la moyenne échantillonnée $\widehat{m}$ et la matrice de covariance échantillonnée $\widehat{\Sigma}$ à partir des points générés.

   Solution Attention, peu importe la librairie, il faut s'assurer d'utiliser **les estimateurs non biaisés pour un échantillon** :

$$m_X \cong \hat{m}_X = \frac{1}{N}\sum_{k=1}^{N} X_k \tag{14.1}$$

$$\Sigma_X \cong \hat{\Sigma}_X = \frac{1}{N-1}\sum_{k=1}^{N}(X_k - \hat{m}_X)(X_k - \hat{m}_X)^T \tag{14.2}$$

4. Répéter les opérations 1. à 3. ci-dessus 10 fois en stockant à chaque fois les valeurs de $\widehat{m}$ et de $\widehat{\Sigma}$.

5. Calculer ensuite la moyenne et l'écart type des $\widehat{m}$ et $\widehat{\Sigma}$, i.e. la moyenne et l'écart-type $m_{\widehat{m}}$ et $\sigma_{\widehat{m}}$ de la moyenne $\widehat{m}$ de chaque essai et les mêmes quantités $m_{\widehat{\Sigma}}$ et $\sigma_{\widehat{\Sigma}}$ pour la matrice de covariance $\widehat{\Sigma}$ de chaque essai.

6. Est-ce que $\widehat{m}$ et $\widehat{\Sigma}$ fluctuent beaucoup d'un essai à l'autre ? Commentez vos résultats : est-ce que les estimateurs sont bons et est-ce que la taille de l'échantillonnage a une influence sur l'estimation ?

   Solution Plus les écarts-types $\sigma_{\widehat{m}}$ et $\sigma_{\widehat{\Sigma}}$ calculés en 5. sur les valeurs de $\widehat{m}$ et de $\widehat{\Sigma}$ sont grands, plus cela signifie que la variabilité est grande d'un échantillon à l'autre.

7. Essayez des tailles d'échantillon différentes. Refaire les manipulations précédentes cette fois-ci pour N =10, 50, 100 et 500.

8. Tracer sur un graphe les écart-types $\sigma_{\widehat{m}}$ et $\sigma_{\widehat{\Sigma}}$ en fonction de la taille N =10, 20 50, 100 et 500. Placer en abscisse les valeurs de N et en ordonnée celles de $\sigma_{\widehat{m}}$ et $\sigma_{\widehat{\Sigma}}$. Comme la matrice de covariance est de dimension 4, il faut donc faire 4 graphiques pour observer l'évolution de la qualité de l'estimation des éléments de la matrice de covariance.

9. Examiner l'influence de N (taille de l'échantillonnage) sur les erreurs (variances) d'estimation de $\widehat{m}$ et de $\widehat{\Sigma}$. Quelles sont vos observations ?

   Solution La variabilité devrait diminuer avec la taille de l'échantillon, et la moyenne des moyennes devrait se rapprocher de la moyenne voulue à mesure que la taille de l'échantillon augmente.

L1.P2 Calculs de vecteurs propres et valeurs propres

a) Calculer à la main les vecteurs propres et valeurs propres de $\Sigma = \begin{bmatrix} 21 & -8 \\ -8 & 9 \end{bmatrix}$

Solution

Calculs des valeurs propres : solutionner l'équation caractéristique $|\Sigma - \lambda I| = 0$

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 25 \\ 5 \end{bmatrix}$$

Calcul des vecteurs propres : solutionner pour chaque λ l'équation $(\Sigma - \lambda_x I)\,\vec{e_x} = 0$ (système de 2 équations à 2 inconnues). On rapporte habituellement le vecteur propre normalisé (longueur 1) dont la première coordonnée est positive.

$$\vec{e_1} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{-1}{\sqrt{5}} \end{bmatrix} \qquad \vec{e_2} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix}$$

b) Calculer à la main les vecteurs propres et valeurs propres des densités de probabilités gaussiennes du problème P3.E1.

Solution

$$\begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ \frac{1}{2} \end{bmatrix} \qquad \begin{bmatrix} \lambda_{12} \\ \lambda_{22} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ \frac{1}{2} \end{bmatrix}$$

$$\overrightarrow{e_{11}} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \overrightarrow{e_{21}} = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad \overrightarrow{e_{12}} = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \overrightarrow{e_{22}} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

L1.P3 Calcul du déterminant et de l'inverse d'une matrice

Calculer à la main le déterminant et l'inverse des matrices de covariance du problème P3.E1.

Solution

$$|\Sigma_1| = |\Sigma_2| = 3/4$$

$$\Sigma_1^{-1} = \begin{bmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{bmatrix} \qquad \Sigma_2^{-1} = \begin{bmatrix} 4/3 & 2/3 \\ 2/3 & 4/3 \end{bmatrix}$$

14.2 EXERCICES

L1.E1 Processus de conception en apprentissage machine et code de départ

Objectifs :

- Comprendre les grandes étapes du processus de conception en apprentissage machine et en classification.
- Comprendre la structure du code de départ fourni pour l'APP.
- Comprendre le (pseudo)code nécessaire à la visualisation et à l'extraction des statistiques de la représentation.

---

1. Faites un schéma de concept du processus de conception et d'utilisation d'un classificateur.
2. Constatez la structure du code de départ fourni pour le laboratoire (`labo3Classes.py`). Analyser rapidement quelles classes implémentent les classificateurs Bayésien, PPV et RN couverts dans l'APP, quelles librairies sont utilisées et à quel effet, quels wrappers sont disponibles pour vous simplifier la vie, quelles étapes du processus de conception sont effectuées où dans le code de départ, quelles fonctions de support (e.g. visualisation, opérations fréquentes, etc.) sont disponibles et ce qu'elles font.
3. Ébauchez le pseudocode de la première section de `labo_APP2()` où l'on visualise les 3 classes qui serviront dans tous les laboratoires, et on calcule leurs statistiques de base.

L1.E2 Décorrélation et réduction de dimension de la représentation

Objectif : Comprendre l'opération de décorrélation et les conséquences d'une réduction de dimension sur la discrimination d'un classificateur.

La matrice de covariance Σ pour une classe arbitraire C à trois dimensions est donnée par

$$\Sigma = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

1. Dans un script de votre cru, obtenez dans Python les valeurs et vecteurs propres de la classe. Quelles dimensions sont corrélées ?

Solution

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 7 \end{bmatrix}$$

$$\vec{e_1} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} \qquad \vec{e_2} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} \qquad \vec{e_3} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2. Quelle est la différence entre indépendance et non corrélation ?

   Solution Deux vecteurs non corrélés ne sont pas forcément indépendants. La corrélation est liée au moment d'ordre 2 des lois de probabilité (covariance). Elle est associée à l'intercorrélation entre deux lois de probabilités. Si l'intercorrélation est nulle, les variables sont non liées. Si tous les moments d'ordre supérieur à deux sont aussi nuls, il y a indépendance. L'indépendance est plus contraignante que la non corrélation. Comme les lois gaussiennes ont tous leurs moments supéreurs à 2 nuls, la non corrélation entraîne dans ce cas aussi l'indépendance.

3. Visualiser sur un dessin (papier ou Onenote) 3D les valeurs et vecteurs propres obtenues, et la distribution des points si on assume que la classe a une densité de probabilité gaussienne.

4. Pourrait-on simplement enlever la dimension 2 de la représentation ?

5. Comment se transforme la représentation lorsqu'on représente un vecteur x de la classe C par l'approximation

$$\tilde{x} = c_1 e_1 \tag{14.3}$$

   où e_1 est le vecteur propre correspondant à la plus grande valeur propre et c_1 correspond aux coordonnées du vecteur projeté sur e_1.

   Solution Utiliser `helpers.analysis.project_onto_new_basis()`.

6. Même question si x est plutôt projeté sur les autres vecteurs propres

$$\tilde{x} = c_2 e_2 + c_3 e_3 \tag{14.4}$$

   avec e_i les deux autres vecteurs propres et les c_i correspondant aux coordonnées du vecteur projeté sur chacun d'entre eux.

$$c_i = X^T e_i \tag{14.5}$$

   Calculez la nouvelle matrice de covariance, vecteurs et valeurs propres.

L1.E3 Visualisation de la représentation et décorrélation d'une classe seule puis d'un système de classes

Objectif : Comprendre la distribution de données à classifier, effectuer une décorrélation des données d'un système de classes

---

On souhaite éventuellement résoudre un problème de reconnaissance à trois classes, C1, C2 et C3, dans les fichiers et le code de départ fournis `labo3Classes.py`. Au préalable, nous analysons les données à classer. L'espace d'entrée (i.e. la représentation) est codé sur un vecteur de dimension deux. On dispose d'un échantillonnage de 1000 vecteurs par classe. Compléter le code (chercher les endroits marqués `TODO` dans le code) pour :

1. Visualiser la distribution des points pour les 3 classes.
2. En supposant que la distribution des vecteurs est normale pour chacune des classes, corriger le code pour estimer les vecteurs moyennes et les matrices de covariances associées à chaque classe.

   Solution

   Utiliser `numpy.mean`, `numpy.cov` et `numpy.linalg.eig`. Les classes ont été générées à l'aide des distributions gaussiennes suivantes :

$$m_1 = \begin{bmatrix} 1 \\ 6 \end{bmatrix} \qquad \Sigma_1 = \begin{bmatrix} 4 & 4 \\ 4 & 10 \end{bmatrix}$$

$$m_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$m_3 = \begin{bmatrix} -3 \\ 6 \end{bmatrix} \qquad \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

3. Vérifiez que les coordonnées sont liées (i.e. corrélées) pour la classe C_1. Calculez les variances σ_1^2 et σ_2^2 associées à chacune des dimensions d'entrée pour cette classe.

   Solution $\rho = \frac{2}{\sqrt{10}}$, $\sigma_1^2 = 4$ et $\sigma_2^2 = 10$

4. Calculez les valeurs propres λ_1 et λ_2 et vecteurs propres $\vec{e_1}$ et $\vec{e_2}$ de la matrice de covariance de C_1.

   Solution $\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 12 \\ 2 \end{bmatrix} \qquad \vec{e_1} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} \qquad \vec{e_2} = \begin{bmatrix} \frac{-2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix}$

5. Créez la matrice E qui sera formée des vecteurs propres de C_1 ($\vec{e_1}$ et $\vec{e_2}$)

$$E = \begin{bmatrix} e_{11} & e_{21} \\ e_{12} & e_{22} \end{bmatrix}$$

puis décorrélez les données de la classe par la rotation définie par

$$Y = EX$$

où X est le vecteur formés des données initiales de la classe (corrélées) et Y le nouveau vecteur. Pour valider l'effet de cette rotation, vérifiez que la nouvelle matrice de covariance Σ_Y calculée à partir des données transformées Y s'approche d'une matrice diagonale.

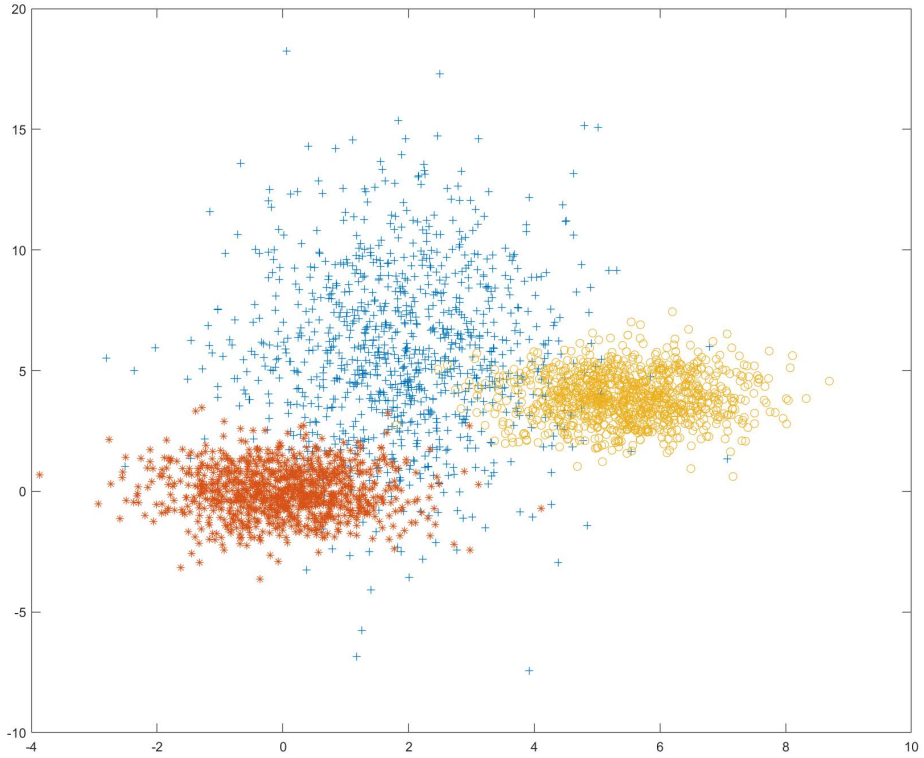
Solution

** Dans Python le format des données est tel qu'on peut utiliser directement la matrice fournie par `eig` comme matrice E comme argument à `helpers.analysis.project_onto_new_basis()`.

6. Est-ce que la décorrélation proposée serait applicable à l'ensemble des classes ? Justifiez. Visualisez le système décorrélé.

Solution

Oui, parce cette décorrélation, qui est en fait une rotation, n'affectera pas les deux autres classes qui ont des distributions déjà décorrélées. En d'autres mots, un modèle gaussien où les variances sont identiques restera décorrélé peu importe la rotation qu'on lui impose. On peut voir ce résultat graphiquement, puisque les distributions dans ce cas-ci sont "rondes" et que le rayon du cercle restera le même peut importe l'angle selon lequel où le regarde. Attention, la rotation dépend de l'ordre des vecteurs propres.



7. Quelle(s) dimension(s) de la représentation du problème est-il souhaitable de conserver ? Avant ou après décorrélation ?

L1.E4 Choix de la représentation

Objectif : Comprendre l'importance d'un bon choix de la représentation pour la classification d'images au moyen d'un exemple dans l'espace des couleurs.

Utilisez le code de départ fourni pour la problématique `problematique.py` pour répondre aux questions suivantes.

1. Visualisez quelques images de chaque classe.
2. Comment une image RGB est-elle stockée en mémoire ?
3. Observez l'histogramme de couleur d'une image. Parmi les images chargées, quelles caractéristiques vous semblent de bons candidats discriminants pour la représentation ?

4. Est-ce que les histogrammes dépendent de la luminosité de l'image, du contraste, etc. ? Comment rendre la représentation robuste à ces conditions ?
5. Est-ce qu'un espace de couleur différent facilite la discrimination des différentes images ? Utilisez par exemple `scikit-image.color` pour les différentes conversions.
6. Compléter le code et la fonction `ImageCollection.generateRGBHistograms()` pour calculer par exemple la moyenne de chaque canal R, G et B pour chaque image.
7. Répéter pour une autre métrique de votre choix.
8. Étudier si les quelques métriques obtenues sont corrélées, discriminantes, etc.

15 PRATIQUE PROCÉDURALE 1

But de l'activité

Le but de cette activité est de se familiariser avec ...

- la notion de frontière de classification
- la classification par distance au(x) plus proche(s) voisin(s)
- l'apprentissage non-supervisé de représentant(s) de classe pour un classificateur k-PPV

15.1 EXERCICES

P1.E1 Classification par les plus proches voisins et par le barycentre

Objectif : Manipuler les concepts de base de la classification non paramétrique.

On considère les 5 points du plan muni de la distance euclidienne

$$D(X,Y) = \sqrt{\sum_{i=1}^{M}(x_i - y_i)^2}$$

et définis par :

$x_1 = (2,2)$; $x_2 = (2,-2)$; $x_3 = (-2,-2)$; $x_4 = (-2,2)$ et $x_5 = (0,0)$.

x_1 , x_2 et x_3 appartiennent à la classe C_1 et x_4 , x_5 appartiennent à la classe C_2.

1. Tracer les frontières des zones de décision suivant la règle du 1-PPV.
2. Déterminer et reporter sur le graphique les barycentres m_1 et m_2 de chaque classe C_1, C_2 ; en déduire et tracer les frontières des zones de décision au sens de la règle du barycentre, i.e. considérer que chaque classe est caractérisée par un seul vecteur représentant qui est le barycentre.
3. Commenter les résultats : quelle est la technique qui permet la meilleure classification (1 PPV ou les barycentres) ?

P1.E2 Algorithme des k-moyennes

Objectif : Comprendre l'apprentissage non supervisé des k-moyennes (k-means).

---

1. Dessinez de façon aléatoire quelques points 2D d'une classe.
2. Décidez du nombre de représentants de classe k que l'algorithme utilisera.
3. Simulez l'algorithme en plaçant à chaque itération les frontières et les barycentres (chapitres 7 et 9 des notes de cours et figure 9.4).
4. Quel critère choisissez-vous pour arrêter le déroulement de l'algorithme ?

P1.E3 Évaluation de la complexité de l'algorithme des k-PPV

Objectif : Étudier la complexité d'un algorithme de classification non paramétrique.

On souhaite implémenter la méthode des plus proches voisins. On suppose :

- Espace (représentation) de dimension M, nombre de classes égal à L, nombre de points disponibles à l'apprentissage égal à N et k est le nombre de plus proches voisins à examiner (k-PPV).
- La distance euclidienne est utilisée.
- On trie par dichotomie (on prend les k plus petites distances parmi les $N-1$ distances calculées (un tri par dichotomie est récursif et nécessite $N \log_2 N$ comparaisons pour trier N valeurs, au lieu de N^2).
- On utilise un critère de décision à la majorité relative.

1. Estimer la complexité de l'algorithme (nombre de multiplications, d'additions, de comparaisons, ...).

   Solution

   $N * [M(soustr + multi) + (M-1)add. + 1\surd] + N \log_2(N)$ comparaisons

2. Évaluer le gain obtenu en remplaçant la distance euclidienne par la distance du **Sup** *(Tchébyscheff)*, qui consiste à ne conserver que le terme dominant dans la somme des différences sur les coordonnées : $D(X,Y) = Max(|x_i - y_i|)_{i=1,...,M}$.

16 PRATIQUE PROCÉDURALE 2

Note : apportez votre laptop, un des numéros requiert l'accès au web.

But de l'activité

Le but du procédural est de se familiariser avec ...

- les réseaux de neurones formels
- l'apprentissage supervisé par rétropropagation de l'erreur
- le choix des hyperparamètres en fonction de la représentation

16.1 EXERCICES

P2.E1 Prédiction d'un RN multicouches et apprentissage par rétropropagation de l'erreur

Objectifs :

- Comprendre la structure d'un réseau de neurones multicouches.
- Comprendre un mécanisme simple d'apprentissage supervisé.

Soit un réseau de neurones composé de deux couches. La première couche contient 2 neurones d'entrée, l_{11} et l_{12}, et la couche de sortie, un seul neurone l_{21}. La représentation x est 2D. La fonction d'activation de l_1 est sigmoïdale $\frac{1}{1+e^{-z}}$, ou z est le noeud interne au neurone qui précède la fonction d'activation, tandis que celle en sortie est un reLu. Les poids du réseau sont listés dans le tableau suivant :

Neurone	w_1	w_2	b
l_{11}	0.5	-0.1	0.75
l_{12}	-0.8	-0.4	0.3
l_{21}	1.2	-0.8	0.1

1. Quelle sera la sortie prédite y_{21A} pour $x_A = \{1, 2\}$?
   Solution $y_{11A} = 0.741$, $y_{12A} = 0.214$ et $y_{21A} = 0.818$

2. On désire entraîner le réseau avec une descente de gradient simple de la forme

$$\Delta w_k = -lr \cdot \frac{\partial e}{\partial w_k} \cdot e \tag{16.1}$$

où e est l'erreur de prédiction ($loss$) et lr le taux d'apprentissage. Quels seront les nouveaux poids si la sortie souhaitée pour x_A était $y^*_{21A} = 0.5$ et le taux d'apprentissage est de 0.1 ?

Solution $\frac{\partial e}{\partial z_{11}} = 0.192$, $\frac{\partial e}{\partial z_{12}} = 0.168$

Neurone	w_1	w_2	b
l_{11}	0.493	-0.106	0.744
l_{12}	-0.805	-0.411	0.305
l_{21}	1.176	-0.807	0.068

3. Calculez la nouvelle valeur prédite avec ces nouveaux poids.

Solution $y_{11A} = 0.736$, $y_{12A} = 0.211$ et $y_{21A} = 0.767$

P2.E2 Entraînement et convergence d'un réseau de neurones

Objectif : pouvoir diagnostiquer les comportements observés lors de l'apprentissage d'un RN.

---

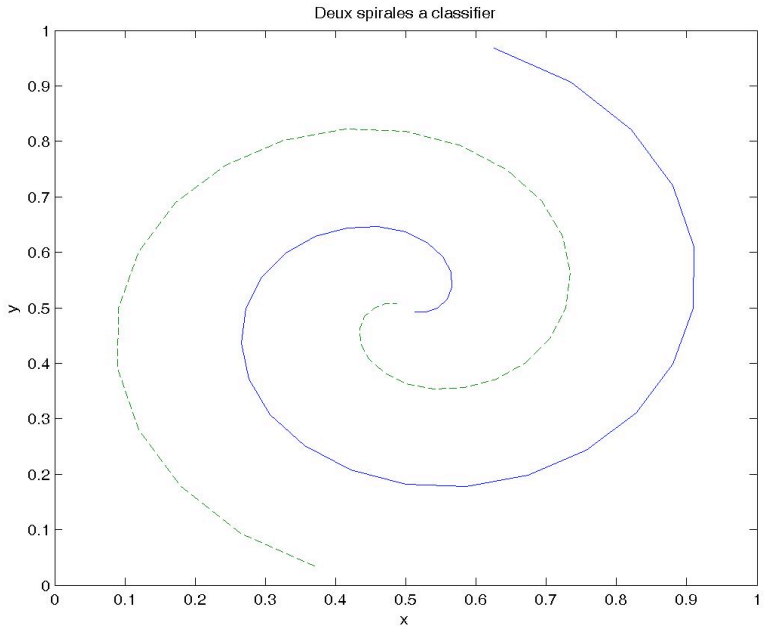
1. Pour une fonction d'activation sigmoïdale, où la dérivée est-elle maximale ? nulle ? Quel est l'impact sur l'apprentissage si le nombre d'epochs est grand ? À quoi sert la normalisation des données dans ce contexte ?
2. Quels sont les avantages et inconvénients des fonctions d'activation usuelles selon l'application (classification, régression, profondeur du réseau, etc.) ?
3. Quel impact sur l'apprentissage ont le rythme d'apprentissage et/ou le momentum ? Commenter le rythme d'apprentissage proposé au numéro précédent. Que se serait-il passé pour un rythme de 2 ? 0.2 ?
4. Nommez et discutez des avantages et inconvénients des algorithmes d'apprentissage usuels en regard du type de données, de l'application, etc. ?
5. Quels sont les loss usuels ? Quelles autres métriques usuelles sont pertinentes à monitorer pendant l'apprentissage ?

6. Pour l'apprentissage, on utilise habituellement un de trois modes : le mode "instantané" ou stochastique, où les poids sont mis à jour pour chacun des vecteurs d'entrée ; le mode "batch" où les poids sont mis à jour seulement après que l'erreur moyenne (loss moyen) pour un sous-ensemble des données ait été estimée, et un autre mode qui consiste à mettre à jour les poids une fois par epoch, à partir de l'erreur moyenne globale. Discutez des 3 techniques et de leurs avantages et inconvénients respectifs.
7. À quoi servent les données de validation ? de test ?
8. Comment diagnostique-t-on un surentraînement, sous-entraînement, un mauvais choix d'hyperparamètres, un mauvais choix de rythme d'apprentissage, un jeu de données inadéquat ? Dessinez les courbes caractéristiques du loss pour les sous-ensembles d'entraînement et de validation dans ces différents cas.

P2.E3 Choix des hyperparamètres en fonction de la représentation

Objectif : comprendre comment choisir les hyperparamètres en fonction du problème.

Dans une représentation 2D (x, y), on considère deux nuages ou classes de points disposés sur des spirales imbriquées l'une dans l'autre. On souhaite concevoir un RN qui classe automatiquement les points selon leur appartenance à une classe ou à une autre.



1. Où sont les frontières ? Est-ce un problème linéairement séparable ?
2. La représentation utilisée est-elle adéquate ? Facilite-elle la discrimination entre les classes ? Quelles autres quantités dérivées de l'espace actuel seraient bénéfiques pour faciliter l'apprentissage des frontières ?
3. Proposez une architecture de RN pour résoudre le problème de classification.
4. Proposez un choix d'hyperparamètres d'entraînement.
5. Testez vos choix sur Google Developper :happy
6. Quel est l'impact du bruit dans la représentation sur la quantité de données nécessaires pour assurer une densité de probabilité locale adéquate pour l'apprentissage (comparez la position des frontières avec ou sans bruit pour plusieurs jeux d'entraînement différents) ? sur le choix des hyperparamètres (batch size, learning rate, etc.) ?

17 PRATIQUE EN LABORATOIRE 2

Buts de l'activité

On vise à ...

- se familiariser avec keras (tensorflow)
- implémenter un classificateur RN simple
- comprendre les résultats d'apprentissage
- appliquer un classificateur à un cas réel

17.1 EXERCICES

L2.E1 Structure générale d'une librairie d'apprentissage machine

Objectif : comprendre la philosophie haut niveau derrière les APIs `keras` ou `scikit-learn`

Élaborez le pseudocode générique qui permet de classifier après avoir apris d'une représentation.

L2.E2 OU exclusif et RNA

Objectif : réaliser un classificateur élémentaire

Armés de votre pseudocode, examinez le script `xor.py` et répondez aux questions suivantes :

1. À quel endroit les données d'entraînement, i.e. la représentation dans ce cas-ci, sont-elles produites ?
2. Est-ce qu'on prétraitement supplémentaire est nécessaire ? (e.g. composantes principales, normalisation, etc.)
3. Combien de neurones et de couches ce réseau comprend-il ? Où l'architecture du réseau est-elle instanciée ?
4. Quelles sont les fonctions d'activation utilisées ? Quels sont les paramètres permettant de contrôler ces fonctions d'activation ?
5. Quel est le rôle de `model.compile()` dans `tensorflow` ?
6. Quelle méthode (méthode au sens logiciel, objets, méthodes, membres, etc.) entraîne le réseau ?

7. Combien d'epochs sont nécessaires pour atteindre la convergence de l'apprentissage ?
8. Est-ce que la configuration fournie permet la convergence et l'implémentation du XOR ? Justifier votre réponse et corriger le script au besoin.
9. Comment utiliseriez-vous le réseau entraîné pour calculer la sortie (prédiction) à de nouvelles entrées ?

L2.E3 Réseau de neurones pour classifier des fleurs

Objectif : concevoir un classificateur RN à partir d'une représentation existante

---

On veut classification automatiquement des fleurs, des iris pour être plus spécifique.

Chaque fleur est définie par 4 caractéristiques : longueur et largeur du sépale, longueur et largeur du pétale. Ces 4 quantités deviennent la représentation du problème. La base de données fournie dans l'archive du laboratoire étiquette aussi les fleurs en 3 classes ("versicolor, virginica et setose"). Ce dataset est courant dans les tutoriels d'apprentissage machine.

Examinez la base de données fournie (`iris.mat`) et le fichier à compléter `iris.py`. Armés de votre pseudocode, analysez-les, complétez le script et classifiez les données d'entraînement avec un taux d'erreur minimal.

1. Étudiez l'espace de la représentation. Est-ce que cette représentation est discriminante ? Est-ce que toutes les dimensions sont utiles ? Lesquelles des 4 dimensions serait-il utile de conserver à votre avis ? Faut-il décorréler ?
2. Justifiez le prétraitement (la normalisation) imposé à la représentation avant l'entraînement.
3. Testez plusieurs configurations (le nombre de couches cachées, de neurones par couches, les fonctions d'activation, etc.). Laquelle donne les meilleurs résultats ? Pourquoi ? Auriez-vous pu anticiper les résultats obtenus ?
4. Testez la généralisation de l'apprentissage en créant un ensemble de validation à partir de la base de données (e.g. les 15 derniers exemples par classe). Est-ce que les performances (taux d'erreur de classification) sont comparables entre (a) la validation et l'entraînement et entre (b) cette méthode et la précédente sans validation ? Qu'est-ce qui explique les différences ? Comment mitiger le problème ?
   Solution Utiliser `sklearn.model_selection.train_test_split`.

5. Le choix de la représentation change-t-il la performance finale du classificateur ? de l'entraînement ?

L2.E4 Classification par réseau de neurones

Objectif : résoudre un problème de classification au moyen d'un réseau de neurones

---

En utilisant les mêmes 3 classes qu'au laboratoire 1,

1. Convertissez les étiquettes de classe en un format qui permet d'utiliser un `loss` plus approprié que le `MSE` pour l'entraînement d'un classificateur. Étudiez comment fonctionne cette stratégie en construisant le pseudocode de l'encodage et du décodage de la prédiction plus loin dans le script, après l'entraînement.

   Solution Utiliser `sklearn.preprocessing.OneHotEncoder` en prévision d'un loss basé sur la `crossentropy`.

2. En vous basant sur les exemples précédents, complétez le code fourni pour déployer un classificateur par RN sur les 3 classes fournies.

3. Partitionnez les données en sous-ensemble d'entraînement et de validation.

4. Utilisez un `callback` pour visualiser la performance de l'entraînement d'une manière plus ergonomique que l'affichage par défaut, par exemple à chaque multiple de 25 epochs, et un autre pour stopper l'entraînement lorsque la généralisation se dégrade.

   Solution
   Voir le `callback` `print_every_N_epochs` fourni et `Keras.callbacks.EarlyStopping`.

5. Calculez et commentez la performance au moyen du taux de classification des données d'origine (fourni à chaque epoch par `tensorflow` via la métrique `accuracy`), et après l'entraînement avec la matrice de confusion.

6. Discutez des résultats en répétant l'entraînement plusieurs fois. Élaborez un choix d'hyperparamètres qui minimise la variabilité observée dans les résultats d'entraînement et qui maximimse simultanément la performance de classification. Comment peut-on anticiper la performance et le choix optimal d'hyperparamètres en analysant les données ?

18 PRATIQUE PROCÉDURALE 3

But de l'activité

Le but de cette activité est de se familiariser avec ...

- le risque de Bayes comme critère de classification
- le calcul de frontières explicites

18.1 EXERCICES

P3.E1 Frontière du critère de Bayes entre deux classes à distribution gaussienne

Objectif :

- Anticiper le comportement d'un classificateur à partir des statistiques descriptives des classes.
- Trouver par le calcul les frontières entre deux classes à loi de probabilité gaussienne.

On considère 2 des 3 classes du code de départ du laboratoire, modélisées par les probabilités a priori $P(C_1) = P(C_2) = 0.5$, et par une distribution gaussienne bidimensionnelle dans l'espace de représentation respectivement égales à $p(X|C_1)$ et $p(X|C_2)$, avec les paramètres suivants.

$$M_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \; M_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix};$$

$$\Sigma_1 = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}; \; \Sigma_2 = \begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix}$$

Décomposition en composantes principales et "visualisation des ellipses"

Au moyen des valeurs et vecteurs propres calculés pendant l'exercice 3 du laboratoire 1 (L1.E3), esquissez les ellipses représentant ces modèles de densité de probabilité gaussienne dans le plan.

1. Dessiner une ellipse associée à chaque distribution et qui est le lieu des points pour lesquels il y a équiprobabilité, en exploitant les propriétes liées aux axes principaux des ellipses.
2. Commenter les frontières attendues entre les classes, les zones de chevauchement des classes, les directions respectives des ellipses, etc.

Calcul des frontières

1. Trouvez à la main la frontière du critère de Bayes qui minimise le risque d'erreur entre ces classes. Quelle est cette frontière ? Tracez-la.

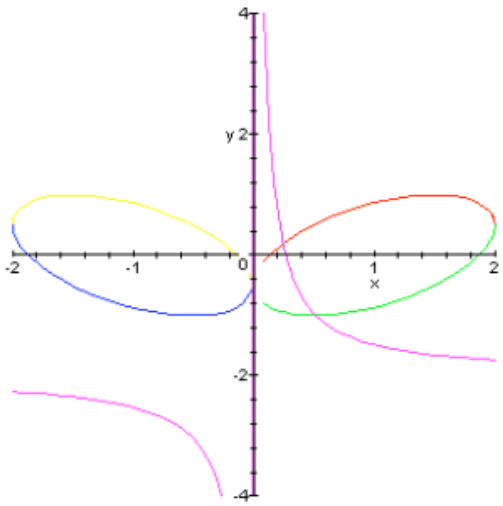
   Solution

   Frontière : $x = 0$

2. On ajoute au classificateur une matrice de coût non diagonale. Trouvez la nouvelle frontière du critère de Bayes qui minimise le coût $l_{11} = l_{22} = 0$ et $l_{12} = 2l_{21}$.
   Quelle est cette frontière ? Tracez-la.

   Solution

   Frontière : $y = \frac{0.52}{x} - 2$



P3.E2 Connaissances dans les techniques étudiées

Objectif : pourquoi "apprentissage machine" ?

Pour conclure les procéduraux de l'APP, discutez où se trouvent les connaissances "apprises" dans un classificateur bayésien ? PPV ? dans un réseau de neurones artificiels ? Comment ces connaissances sont-elles représentées ou encodées ? Dans chacun, quel est le mécanisme pour la mise en œuvre de cette « intelligence » ?

18.2 EXERCICES SUPPLÉMENTAIRES

P3.S1 Évaluation de la probabilité d'erreur pour une classification par erreur minimale

Faire l'exercice 6.4.2 des notes de cours portant sur l'application en télécommunication et pour lequel le corrigé est aussi donné dans les notes de cours.

19 PRATIQUE EN LABORATOIRE 3

Buts de l'activité

Le but de cette activité est d'implémenter ...

- un classificateur par risque de Bayes (distribution gaussienne ou non)
- un classificateur PPV et les k-moyennes

19.1 EXERCICES PRÉPARATOIRES

L3.P1 Calcul de frontières

Objectif : Calculer la frontière entre les classes du laboratoire

---

En supposant que le coût est le même pour toutes les mauvaises décisions, que le problème est défini dans un espace mathématique représenté par les paramètres (x, y), que la probabilité d'appartenir à une classe est la même pour toutes les classes, calculer la frontière analytique entre les classes fournies dans le code de départ et modélisées par les gaussiennes dont les propriétés sont

$$m_1 = \begin{bmatrix} 1 \\ 6 \end{bmatrix} \qquad \Sigma_1 = \begin{bmatrix} 4 & 4 \\ 4 & 10 \end{bmatrix}$$

$$m_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$m_3 = \begin{bmatrix} -3 \\ 6 \end{bmatrix} \qquad \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Solution

Pour obtenir les équations des frontières, il faut égaler les risques :

$$\begin{array}{rcl} R_i(X) & = 1 - P(\mathrm{C}_i|X) = 1 - \frac{p(X|\mathrm{C}_i)P(\mathrm{C}_i)}{P(X)} & \\ R_1(X) & = R_2(X) & = R_3(X) \\ p(X|\mathrm{C}_1)P(\mathrm{C}_1) & = p(X|\mathrm{C}_2)P(\mathrm{C}_2) & = p(X|\mathrm{C}_3)P(\mathrm{C}_3) \end{array}$$

Notez que dans le développement précédent, écrire $R_i(X) = 1 - P(\mathrm{C}_i|X)$ n'est valide que si les coûts sont tous unitaires.

Sachant que la densité de probabilité est

$$p(X|\mathrm{C}_i) = \frac{1}{(2\pi)^{M/2}|\Sigma|^{1/2}} e^{\frac{-d^2(X)}{2}}$$

et en prenant des paires de classes

$$\frac{p(X|\mathrm{C}_i)}{p(X|\mathrm{C}_j)} = \frac{P(\mathrm{C}_j)}{P(\mathrm{C}_i)}$$

on trouve après simplification

$$d_j^2(X) - d_i^2(X) = 2\ln\left[\frac{P(\mathrm{C}_j)}{P(\mathrm{C}_i)}\sqrt{\frac{|\Sigma_i|}{|\Sigma_j|}}\right]$$

où

$$d^2(X) = (X - m_x)^{\mathrm{T}}\Sigma^{-1}(X - m_x)$$

En développant le calcul matriciel

$$\begin{aligned} d^2(X) &= \begin{bmatrix} x - m_x & y - m_y \end{bmatrix} \begin{bmatrix} \Sigma_{11}^{-1} & \Sigma_{12}^{-1} \\ \Sigma_{12}^{-1} & \Sigma_{22}^{-1} \end{bmatrix} \begin{bmatrix} x - m_x \\ y - m_y \end{bmatrix} \\ &= \begin{bmatrix} x - m_x & y - m_y \end{bmatrix} \begin{bmatrix} \Sigma_{11}^{-1}(x - m_x) + \Sigma_{12}^{-1}(y - m_y) \\ \Sigma_{12}^{-1}(x - m_x) + \Sigma_{22}^{-1}(y - m_y) \end{bmatrix} \\ &= \Sigma_{11}^{-1}(x - m_x)^2 + 2\Sigma_{12}^{-1}(x - m_x)(y - m_y) + \Sigma_{22}^{-1}(y - m_y)^2 \end{aligned}$$

Notez que ce résultat est implémenté semi-analytiquement dans la fonction `helpers.classifiers.get_gaussian_borders()`.

Ensuite on trouve pour chaque classe

$$d_1^2(X) = \frac{5}{12}(x-1)^2 - \frac{1}{3}(x-1)(y-6) + \frac{1}{6}(y-6)^2$$
$$d_2^2(X) = x^2 + y^2$$
$$d_3^2(X) = (x+3)^2 + (y-6)^2$$

Frontière entre C_1 et C_2 :

$$d_2^2(X) - d_1^2(X) = \frac{7}{12}x^2 + \frac{5}{6}y^2 - \frac{7}{6}x - \frac{53}{12} + \frac{1}{3}xy + \frac{5}{3}y = \ln\frac{|\Sigma_1|}{|\Sigma_2|} = \ln(24)$$

Frontière entre C_2 et C_3 :

$$d_3^2(X) - d_2^2(X) = 6x + 45 - 12y = \ln 1 = 0$$

Frontière entre C_1 et C_3 :

$$d_1^2(X) - d_3^2(X) = -\frac{7}{12}x^2 - \frac{29}{6}x - \frac{487}{12} - \frac{1}{3}xy + \frac{31}{3}y - \frac{5}{6}y^2 = -\ln(24)$$

19.2 EXERCICES

L3.E1 Modèles gaussiens

Objectif : Construire et visualiser des modèles gaussiens de la représentation

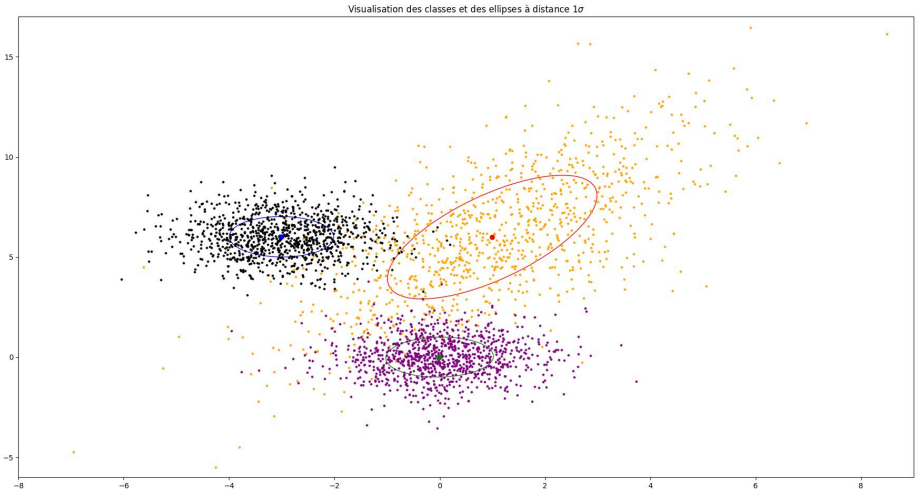
---

Analysez (pseudocode !) et complétez le code fourni (même code qu'aux laboratoires précédents) pour :

1. Obtenir les valeurs et vecteurs propres de ces classes, puis superposer sur le graphique les ellipses à 1σ de ces classes.

   Solution

   Étudier `ClassificationData.getStats()` et utilisez `matplotlib.patches.Ellipse`.

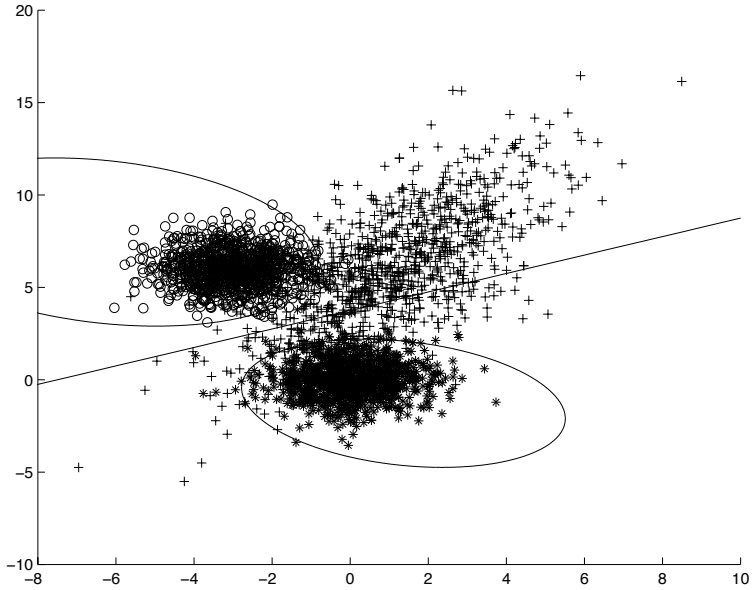


2. Intuitivement, déterminer où un classificateur optimal positionnerait la frontière de décision entre les classes ?

3. Superposer sur le graphique des classes les frontières calculées dans l'exercice préparatoire.

   Solution

   Utiliser `helpers.classifiers.get_gaussian_borders()` pour obtenir les coefficients, et tous les arguments de `helpers.analysis.view_classes` dans la méthode `ClassificationData.get_borders()`.



4. Laquelle des frontières est une "droite" ? Pourquoi ? Quelles frontières ne sont pas linéaires ? Pourquoi ?

L3.E2 Classification avec un modèle non paramétrique

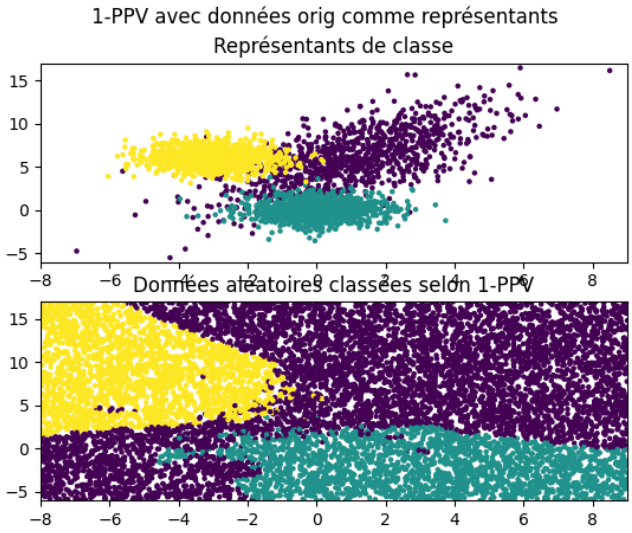
Objectif : mettre en oeuvre un classificateur k-PPV qui utilise ou non les k-moyennes.

En utilisant les mêmes 3 classes fournies, compléter votre pseudocode et le code pour...

1. En prenant les points originaux déjà classés comme représentants de classes, implémenter un 1-PPV.

   Solution

   Utiliser `sklearn.neighbors.KNeighborsClassifier`.

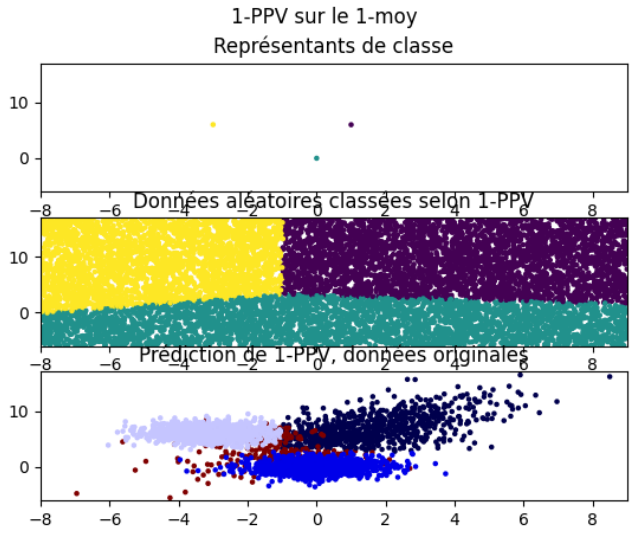


2. Comparer la performance d'un 5-PPV en comparaison du 1-PPV.

3. Générer 1 seul représentant pour chaque classe au moyen de l'algorithme des k-moyennes (1-moyenne) et comparer comment se comporte le 1-PPV qui utilise ces nouveaux représentants par rapport aux autres classificateurs ?

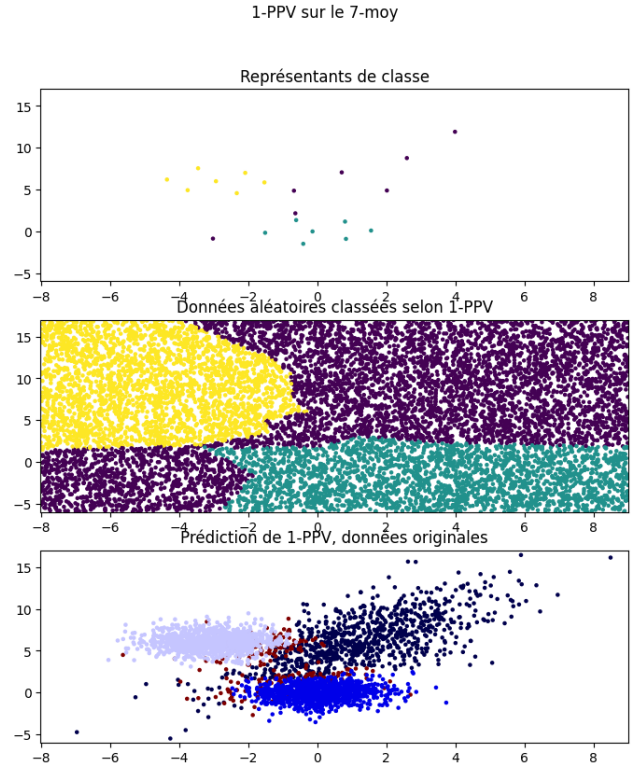
   Solution

   Utiliser `sklearn.cluster.KMeans`.



4. Déterminer si le résultat est différent pour un 7-moyenne suivi d'un 1-PPV ?

Solution



L3.E3 Classificateur Bayésien

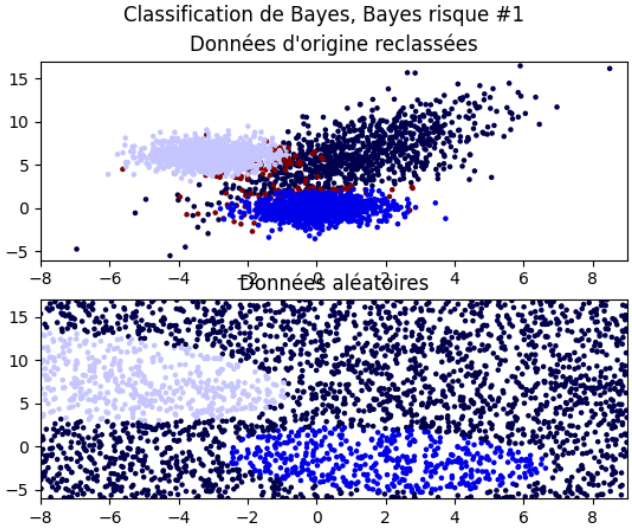
Objectif : se familiariser avec l'implémentation d'un classificateur bayésien

---

La librairie `scikit-learn` ne contient pas de fonction qui permette d'implémenter un classificateur bayésien pour des densités gaussiennes arbitraires non décorrélées. Il faut donc créer un classificateur de toutes pièces.

1. Élaborer le pseudocode d'un classificateur de Bayes où le risque est appliqué à des modèles de densité de probabilité pour chaque classe. Assumer pour l'instant des classes équiprobables à coût unitaire, comme l'exercice préparatoire.
2. Comparer avec la classe `BayesClassifier` et la compléter pour déployer le classificateur précédent sur les 3 classes fournies.

   Solution



L3.E4 Synthèse et comparaison

Objectif : Discuter des performances obtenues.

Comparez pour tous les classificateurs de l'APP le taux de classification des données originales, les matrices de confusion, la qualité des frontières de décision respectives, la complexité relative de l'entraînement et de la prédiction, ainsi que tout autre élément pertinent à la discussion.

19.3 EXERCICES SUPPLÉMENTAIRES

L3.S1 Classificateur Bayésien complet

Objectif : Modifier le classificateur pour implémenter des a priori et des coûts dans le calcul du risque

---

1. Modifier le pseudocode du classificateur de Bayes pour incorporer la prise en compte de classes non équiprobables et des coûts arbitraires.

2. Modifier la classe `BayesClassifier`.

L3.S2 Classificateur Bayésien à densité de probabilité arbitraire

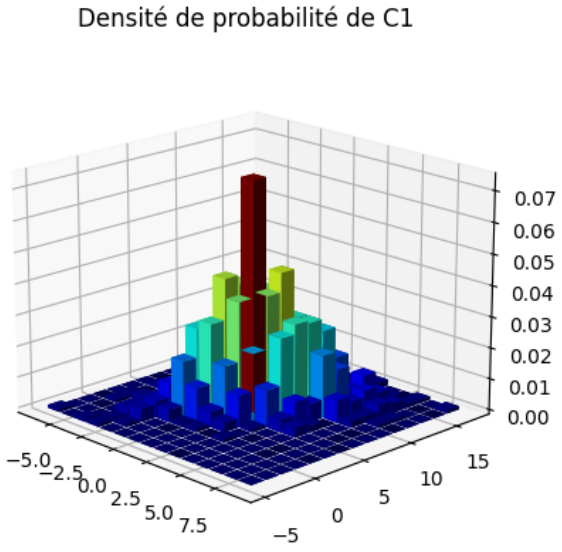
Objectif : mettre en oeuvre un classificateur bayésien à densité de probabilité arbitraire.

En vue de la résolution de la problématique, modifier le pseudocode et le code du classificateur bayésien pour utiliser un modèle de probabilité arbitraire plutôt que gaussien.

Complétez la classe `histProbDensity` pour :

1. Construire un modèle empirique de densité de probabilité pour chacune des classes, au lieu du modèle gaussien.

   Solution En 2D, utiliser pour l'entraînement `numpy.histogram2d`, voir en particulier la fonction `helpers.analysis.creer_hist2D()`, sinon écrire le pseudocode qui compte le nombre de données dans un voxel à plusieurs dimensions et l'implémenter. Exemple pour C_1 du laboratoire :



2. Compléter la méthode de prédiction pour calculer la probabilité d'appartenir à cette classe.
3. Instancier un classificateur `BayesClassifier` qui utilise ces densités arbitraires plutôt que celles gaussiennes.