IFT-6390 Fundamentals of Machine Learning
Professor: Ioannis Mitliagkas

Homework 1 - Theoretical part
Marc-Antoine Provost

Devoir 1 - Partie Théorique

1. **Probability warm-up: conditional probabilities and Bayes rule** [5 points]

   Rappels de probabilités: probabilité conditionnelle et règle de Bayes

   (a) Give the definition of the conditional probability of a discrete random variable X given a discrete random variable Y.
   Donnez la définition de la probabilité conditionnelle de la variable aléatoire discrète X sachant la variable aléatoire discrète Y

   ### Answer 1.a)

   $$\mathrm{P(A|B)} = \frac{P(A \cap B)}{P(B)}$$

   (b) Consider a biased coin with probability 2/3 of landing on heads and 1/3 on tails. This coin is tossed three times. What is the probability that exactly two heads occur (out of the three tosses) given that the first outcome was a head?
   Soit une pièce déséquilibrée dont la probabilité d'obtenir face est 2/3 et la probabilité d'obtenir pile est 1/3. Cette pièce est lancée à trois reprises. Quelle est la probabilité d'obtenir exactement deux faces (parmis les trois lancers), sachant que le premier lancer a fait face ?

Answer 1.b)

With the conditional probability formula and the fact the the events are independent, we have

$$P(A \cap B) = P(A)P(B)$$

$$\begin{aligned}
\frac{P(A \cap B)}{P(B)} &= \\
\frac{P(A)P(B)}{P(B)} &= \frac{\frac{4}{9} \cdot \frac{2}{3}}{\frac{2}{3}} \\
&= \frac{4}{9} \\
&= \frac{2}{3}
\end{aligned}$$

(c) Give two equivalent expressions of $P(X, Y)$:

(i) as a function of $P(X)$ and $P(Y|X)$

(ii) as a function of $P(Y)$ and $P(X|Y)$

Donnez deux expressions équivalentes de $P(X, Y)$:

(i) en fonction de $P(X)$ et $P(Y|X)$

(ii) en fonction de $P(Y)$ et $P(X|Y)$

Answer 1.ci)

$$P(X, Y) = P(Y|X)P(X)$$

Answer 1.cii)

$$P(X, Y) = P(Y)P(X|Y)$$

(d) Prove Bayes theorem:
Prouvez le théorème de Bayes:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}.$$

Answer 1d)

If we start from the conditional probability of two events X and Y, $P(X \cap Y)$, it is equal to $P(X)P(Y|X)$.
So we have,

$$P(X \cap Y) = P(X)P(Y|X)$$

Which is also equal to

$$P(X \cap Y) = P(Y)P(X|Y)$$

By equating the two, we get

$$P(X)P(Y|X) = P(Y)P(X|Y)$$

and thus

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

(e) A survey of certain Montreal students is done, where 55% of the surveyed students are affiliated with UdeM while the others are affiliated with McGill. A student is drawn randomly from this surveyed group.

Un sondage des étudiants Montréalais est fait, où 55% des élèves sondés sont affiliés à l'UdeM alors que les autres sont affiliés à McGill. Un étudiant est choisi aléatoirement parmis ce groupe.

i. What is the probability that the student is affiliated with McGill?
Quelle est la probabilité que l'étudiant soit affilié à McGill?

Answer 1.ei)

1 - 0.55 = 0.45, there is a 45% probability that the random student being drawn is from McGill.

ii. Now let's say that this student is bilingual, and you know that 80% of UdeM students are bilingual while 50% of McGill students are. Given this information, what is the probability

that this student is affiliated with McGill ?
Considérons maintenant que l'étudiant est bilingue, et que 80% des étudiants de l'UdeM sont bilingues alors que seulement 50% des étudiants de McGill le sont. Étant donné cette information, quelle est la probabilité que cet étudiant soit affilé à McGill ?

Answer 1.eii)

With X = being affiliated with McGill and Y = being bilingual, we have

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}$$

$$P(X \cap Y) = P(Y|X)P(X)$$

$$\begin{aligned} P(Y|X)P(X) &= 0.5 \cdot 0.45 \\ &= 0.225 \\ P(Y) &= (0.55 \cdot 0.8) + (0.45 \cdot 0.5) \\ &= 0.665 \\ P(X|Y) &= \frac{0.225}{0.665} \\ &\approx 0.338 \end{aligned}$$

2. **Bag of words and single topic model** [10 points]

Bag of words (sac de mots) et modèle de sujet unique

We consider a classification problem where we want to predict the topic of a document from a given corpus (collection of documents). The topic of each document can either be *sports* or *politics*. 2/3 of the documents in the corpus are about *sports* and 1/3 are about *politics*.

We will use a very simple model where we ignore the order of the words appearing in a document and we assume that words in a document are independent from one another given the topic of the document.

On s'intéresse à un problème de classification où l'on veut prédire le sujet d'un document d'un certain corpus (ensemble de documents). Le sujet de chaque document peut être soit *sport*, soit *politique*. 2/3 des documents du corpus sont sur le *sport*, et 1/3 sont sur la *politique*.

On va utiliser un modèle très simple où on ignore l'ordre des mots apparaissant dans le document et l'on considère que les mots dans un document sont indépendants les uns des autres, étant donné le sujet du document.

In addition, we will use very simple statistics of each document as features: the probabilities that a word chosen randomly in the document is either "goal", "kick", "congress", "vote", or any another word (denoted by *other*). We will call these five categories the vocabulary or dictionary for the documents: $V = \{"goal", "kick", "congress", "vote", other\}$.

De plus, nous allons utiliser des statistiques très simples des documents: les probabilités qu'un mot choisi au hasard dans un document soit "goal", "kick", "congress", "vote", ou n'importe quel autre mot (dénoté par *other*). Nous appelons ces cinq catégories le vocabulaire ou dictionnaire pour les documents: $V = \{"goal", "kick", "congress", "vote", other\}$.

Consider the following distributions over words in the vocabulary given a particular topic:

Soit les distributions suivantes des mots du vocabulaire, par sujet:

	$\mathbb{P}(\text{word} \mid \text{topic} = \textit{sports})$	$\mathbb{P}(\text{word} \mid \text{topic} = \textit{politics})$
word = "*goal*"	1/100	7/1000
word = "*kick*"	1/200	3/1000
word = "*congress*"	0	1/50
word = "*vote*"	5/1000	1/100
word = *other*	980/1000	960/1000

Table 1:

This table tells us for example that the probability that a word chosen at random in a document is "vote" is only 5/1000 if the topic of the document is *sport*, but it is 1/100 if the topic is *politics*.

Cette table nous dit par exemple que la probabilité qu'un mot choisi aléatoirement dans un document soit "vote" n'est que de 5/1000 si le sujet du document est le *sport*, mais est de 1/100 si le sujet est la *politique*.

(a) What is the probability that a random word in a document is "goal" given that the topic is *politics*?
Quelle est la probabilité qu'un mot aléatoire dans un document soit "goal" étant donné que le sujet est la *politique* ?

Answer 2.a)

P(word = *"goal"*|topic = *politics*) = 7/1000

(b) In expectation, how many times will the word "goal" appear in a document containing 200 words whose topic is *sports*?
Quelle est l'espérance du nombre de fois où le mot "goal" apparait dans un document de 200 mots dont le sujet est le *sport*?

Answer 2.b)

With the random variable X being the number of time the word "goal" appear in a document and the random variable Y being the topic, we have

$$E(X|Y) = \frac{1}{100} \cdot 200 = 2$$

(c) We draw randomly a document from the corpus. What is the probability that a random word of this document is "goal"?
On tire aléatoirement un document du corpus. Quelle est la probabilité qu'un mot aléatoire de ce document soit "goal"?

Answer 2.c)

With the random variable W being the word and the random variable T being the topic, we have

$$\begin{aligned}
P(W = ”goal”) = & \\
& P(W = ”goal|topic = sport) \cdot P(T = sport) \\
& + P(W = ”goal”|topic = politics) \cdot P(T = politics) \\
& = \frac{1}{100} \cdot \frac{2}{3} + \frac{7}{1000} \cdot \frac{1}{3} \\
& = \frac{27}{3000} \\
& = \frac{9}{1000}
\end{aligned}$$

(d) Suppose that we draw a random word from a document and this word is "kick". What is the probability that the topic of the document is *sports*?
Supposons que l'on tire aléatoirement un mot d'un document et

que ce mot est "kick". Quelle est la probabilité que le sujet du document soit le *sport*?

Answer 2.d)

Using Bayes' theorem, and with the random variable W being the word and the random variable T being the topic, we have

$P(T = sport|W = kick) = \frac{P(W=kick|T=sport)P(T=sport)}{P(W=kick)}$
$= \frac{\frac{1}{200}\cdot\frac{2}{3}}{\frac{1}{200}\cdot\frac{2}{3}+\frac{3}{1000}\cdot\frac{1}{3}}$

≈ 0.769

(e) Suppose that we randomly draw two words from a document and the first one is "kick". What is the probability that the second word is "goal"?
Supposons que l'on tire aléatoirement deux mots d'un document et que le premier soit "kick". Quelle est la probabilité que le second mot soit "goal"?

Answer 2.e)

With random variable W representing the first word and the random variable X representing the second word, we have
$P(X = "goal"|W = "kick") =$
$P(goal|sport) \cdot P(sport|kick) + P(goal|politics) \cdot P(politics|kick)$
Beforehand, we need to calculate the information that we don't have from the previous exercises, which is P(politics|kick) and is equal to
$P(politics|kick) = \frac{\frac{3}{1000}\cdot\frac{1}{3}}{\frac{1}{200}\cdot\frac{2}{3}+\frac{3}{1000}\cdot\frac{1}{3}}$
$= \frac{3}{13}$
With this information, we get
$P(X = "goal"|W = "kick") = \frac{1}{100} \cdot \frac{10}{13} + \frac{7}{1000} \cdot \frac{3}{13}$
$= \frac{121}{13000}$

(f) Going back to learning, suppose that you do not know the conditional probabilities given a topic or the probability of each topic (i.e. you don't have access to the information in table 1 or the topic distribution), but you have a dataset of N documents where each document is labeled with one of the topics *sports* and *politics*. How would you estimate the conditional probabilities (e.g., $\mathbb{P}(\text{word} = \text{"}\textit{goal}\text{"} \mid \text{topic} = \textit{politics})$) and topic probabilities (e.g., $\mathbb{P}(\text{topic} = \textit{politics})$) from this dataset?

Pour en revenir à l'apprentissage, supposons que l'on ne connaisse pas les probabilités conditionnelles étant donné chaque sujet ni les probabilités de chaque sujet (i.e. nous n'avons pas accès aux informations de la table 1 où aux proportions de chaque sujet), mais nous avons un jeu de données de N documents où chaque document est annoté avec un des sujet *sport* ou *politique*. Comment estimeriez vous les probabilitiés conditionelles (e.g., $\mathbb{P}(\text{mot} = \text{"}\textit{goal}\text{"} \mid \text{sujet} = \textit{politique})$) et les probabilités des sujets (e.g., $\mathbb{P}(\text{sujet} = \textit{politique})$) à partir de ce jeu de données ?

Answer 2.f)

To estimate the topic probabilities, I would do

$$\frac{\text{total number of documents about this particular topic}}{\text{total number of documents}}$$

And to estimate the conditional probabilities (e.g., P(word = "goal" | topic = politics) , I would do

$$\frac{\text{total number of particular word in this particular topic}}{\text{total number of words in this particular topic}}$$

3. **Maximum likelihood estimation** [5 points]

   Estimateur du maximum de vraisemblance

   Let $x \in \mathbb{R}$ be uniformly distributed in the interval $[0, \theta]$ where θ is a parameter. That is, the pdf of x is given by

$$f_\theta(x) = \begin{cases} 1/\theta & \text{if } 0 \leq \mathbf{x} \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

   Soit $x \in \mathbb{R}$ distribué uniformément dans l'interval $[0, \theta]$ où θ est un paramètre. C'est à dire que la fonction de densité de probabilité de x est donnée par :

$$f_\theta(x) = \begin{cases} 1/\theta & \text{si } 0 \leq \mathbf{x} \leq \theta \\ 0 & \text{sinon} \end{cases}$$

Suppose that n samples $D = \{x_1, \ldots, x_n\}$ are drawn independently according to $f_\theta(x)$.
Supposons que n points $D = \{x_1, \ldots, x_n\}$ sont tirés aléatoirement indépendemment selon $f_\theta(x)$.

(a) Let $f_\theta(x_1, x_2, \ldots, x_n)$ denote the joint pdf of n independent and identically distributed (i.i.d.) samples drawn according to $f_\theta(x)$. Express $f_\theta(x_1, x_2, \ldots, x_n)$ as a function of $f_\theta(x_1), f_\theta(x_2), \ldots, f_\theta(x_n)$
Soit $f_\theta(x_1, x_2, \ldots, x_n)$ la fonction de densité de probabilité jointe de n points indépendemment et identiquement distribué (i.i.d) selon $f_\theta(x)$. Exprimez $f_\theta(x_1, x_2, \ldots, x_n)$ en fonction de $f_\theta(x_1), f_\theta(x_2), \ldots, f_\theta(x_n)$

Answer 3.a)

$f(D|\theta) = f(x_1, x_2, ..., x_n|\theta)$

$= \prod_{i=1}^{n} f(x_i|\theta)$

(b) We define the maximum likelihood estimate by the value of θ which maximizes the likelihood of having generated the dataset D from the distribution $f_\theta(x)$. Formally,

$$\theta_{MLE} = \arg\max_{\theta \in \mathbb{R}} f_\theta(x_1, x_2, \ldots, x_n),$$

Show that the maximum likelihood estimate of θ is $max(x_1, \ldots, x_n)$

On définie l'estimateur du maximum de vraisemblance comme la valeur de θ qui maximise la vraisemblance de générer le jeu de donnée D à partir de la distribution $f_\theta(x)$. Formellement,

$$\theta_{MLE} = \arg\max_{\theta \in \mathbb{R}} f_\theta(x_1, x_2, \ldots, x_n),$$

Montrez que l'estimateur du maximum de vraisemblance de θ est $max(x_1, \ldots, x_n)$.

Answer 3.b)

The likelihood function is given by

$$L(\theta) = \prod_{i=1}^{n} = \frac{1}{\theta^n} I(X_1, ..., X_n \epsilon [0, \theta])$$

Where the indicator function equals to 1 if the logical value of its argument is true and 0 otherwise. From this, we can observe that $\frac{1}{\theta^n}$ decreases as θ increases, but that the indicator function returns 0 if $max(X_1, ..., X_n) > \theta$ which is the same as saying that the indicator function returns 0 if $\theta < max(X_1, ..., X_n)$. Hence, the likelihood function is maximized at $\hat{\theta} = max(X_1, ..., X_n)$

4. **Maximum likelihood estimation 2** [10 points]

   Estimateur de maximum de vraisemblance 2

   Consider the following probability density function:

   $$f_\theta(x) = 2\theta x e^{-\theta x^2}$$

   where θ is a parameter and x is positive real number.
   Soit la fonction de densité de probabilité suivante:

   $$f_\theta(x) = 2\theta x e^{-\theta x^2}$$

   où θ est un paramètre et x est un nombre réel positif.

   Using the same notation as in exercise 3, compute the maximum likelihood estimate of θ.
   (hint: you may simplify computations by proving that the maximizer of $f_\theta(x_1, x_2, \ldots, x_n)$ is also the maximizer of $log[f_\theta(x_1, x_2, \ldots, x_n)]$)
   En utilisant les mêmes notations que dans dans l'exercice 3, calculez l'estimateur du maximum de vraisemblance de θ.
   (indice: vous pouvez simplifier les calculs en prouvant que le θ maximisant $f_\theta(x_1, x_2, \ldots, x_n)$ corresponds aussi au maximum de $log[f_\theta(x_1, x_2, \ldots, x_n)]$)

Answer 4

$$\begin{aligned}
l(\theta) &= \sum_{i=1}^{n} log(f(x_i|\theta)) \\
&= \sum_{i=1}^{n} log(2\theta x e^{-\theta x_i^2}) \\
&= \sum_{i=1}^{n} log(2\theta x_i) + log(e^{-\theta x_i^2}) \\
&= \sum_{i=1}^{n} log(x_i) + log(2\theta) + log(e^{-\theta x_i^2)} \\
&= \sum_{i=1}^{n} log(x_i) + log(2\theta) - \theta x^2 log(e) \\
&= \sum_{i=1}^{n} log(x_i) + log(2\theta) - \theta x_i^2 \\
&= \sum_{i=1}^{n} log(x_i) + log(2\theta) + log(\theta) - \theta x_i^2
\end{aligned}$$

By taking the derivative with respect to θ, we have

$$\frac{d(\sum_{i=1}^{n} log(x_i) + log(2\theta) + log(\theta) - \theta x_i^2)}{d\theta}$$
$$= \frac{n}{\theta} - \sum_{i=1}^{n} x_i^2$$

To find the MLE, we equal the derivative to 0, which gives us $\hat{\theta} = \frac{n}{\sum_{i=1}^{n} x_i^2}$

5. ***k*-nearest neighbors** [10 points]

   k plus proches voisins

   Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n independent labelled samples drawn using the following sampling process:

   - the label of each $\mathbf{x}_i$ is drawn randomly with 50% probability for each of the two classes

- x_i is drawn uniformly in S^+ if its label is positive, and uniformly in S^- otherwise

Where S^+ and S^- are two **unit** hyperspheres whose centers are 10 units apart.
Soit $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ un ensemble de n points labelés indépendants, tirés aléatoirement suivant la procédure suivante:

- le label de chaque $\mathbf{x}_i$ est tiré aléatoirement avec une probabilité de 50% pour chacune des deux classes
- x_i est tiré uniformément dans S^+ si son label est positif, et uniformément dans S^- sinon

Où S^+ et S^- sont deux hypersphères **unitaires** dont les centres sont espacés de 10 unités.

(a) Show that if k is odd the average probability of error of the k-NN classifier is given by

$$P_n(e) = \frac{1}{2^n} \sum_{j=0}^{(k-1)/2} \binom{n}{j}.$$

Montrez que si k est impair, la probabilité d'erreur moyenne du classifieur des k plus proches voisins est donné par

$$P_n(e) = \frac{1}{2^n} \sum_{j=0}^{(k-1)/2} \binom{n}{j}.$$

Answer 5.a)

In the k-NN algorithm, an error occurs when our hyperspheres S^+ and S^- have less than (k - 1)/2 labelled samples within themselves. That is ;

P(error) =P(True label is S^+ ,while less than $\frac{k+1}{2}$ nearest neighbors are in S^+) + P(True label is S^-, while less than $\frac{k+1}{2}$ nearest neighbors are in S^-)

$$= P(\text{Less than } \frac{k+1}{2} \text{nearest neighbors are in } S^+|S^+)P(S^+)$$
$$+ P(\text{Less than } \frac{k+1}{2} \text{nearest neighbors are in } S^-|S^-)P(S^-)$$

Since x_i is drawn uniformly and $S+$ and $S-$ are two unit hyperspheres whose centers are 10 units apart, we can say that

$$\begin{aligned} P(\text{m nearest neighbors in} S^+|S^+) &= P(\text{m of n samples in } S^+|S^+) \\ &= \frac{\binom{n}{m}}{2^n} \end{aligned}$$

We can also say :

$$\begin{aligned} P(\text{m nearest neighbors in} S^-|S^-) &= P(\text{m of n samples in } S^-|S^-) \\ &= \frac{\binom{n}{m}}{2^n} \end{aligned}$$

Thus,

$$P_n(e) = \frac{1}{2^n} \sum_{j=0}^{(k-1)/2} \binom{n}{j}$$

(b) Show that in this case the single-nearest neighbor classifier ($k = 1$) has a lower error rate than the k-NN classifier for $k > 1$.
Montrez que dans ce cas le classifieur du plus proche voisin ($k = 1$) a un plus faible taux d'erreur que le classifieur des k plus proches voisins pour $k > 1$.

Answer 5.b)

If we take k = 1, the average probability of error of the k-NN classifier as seen in 5.a) gives us

$$\begin{aligned} P_n(e) &= \frac{1}{2^n} \sum_{j=0}^{0} \binom{n}{j} \\ &= \frac{1}{2^n} \end{aligned}$$

If we take the hyperparameter k > 1, the average probability of error of the k-NN classifier gives us

$$\begin{aligned} P_n(e) &= \frac{1}{2^n} \sum_{j=0}^{1} \binom{n}{j} \\ &> \frac{1}{2^n} \end{aligned}$$

We can then see that the single-nearest neighbor classifier (k = 1) has a lower error rate than the k-NN classifier for k > 1 .

(c) If k is allowed to increase with n but is restricted by $k \leq a\sqrt{n}$ (for some constant a), show that $P_n(e) \to 0$ as $n \to \infty$.
Si k peut augmenter avec n mais est limité par $k \leq a\sqrt{n}$ (pour une constante a), montrez que $P_n(e) \to 0$ lorsque $n \to \infty$.

Answer 5.c)

For j $\geq$ 0 and j $\leq \frac{k-1}{2}$,

$$\binom{n}{j} \leq \binom{n}{\frac{k-1}{2}}$$

Hence,

$$\frac{1}{2^n}\sum_{j=0}^{(k-1)/2}\binom{n}{j} \leq \frac{1}{2^n}\sum_{j=0}^{(k-1)/2}\binom{n}{\frac{k-1}{2}}$$
$$= \frac{1}{2^n}\frac{k+1}{2}\binom{n}{\frac{k-1}{2}}$$
$$= \frac{1}{2^n}\frac{k+1}{2}\frac{n!}{(\frac{k-1}{2})!(n-\frac{k-1}{2})!}$$
$$\leq \frac{1}{2^n}\frac{n!}{(n-\frac{k-1}{2})!}$$
$$\leq \frac{1}{2^n}\frac{n!}{(n-k)!}$$
$$= \frac{1}{2^n}\frac{n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(n-k+1)\cdot(n-k)\cdot(n-k-1)\cdot\ldots\cdot3\cdot2\cdot1}{(n-k)\cdot(n-k-1)\cdot(n-k-2)\cdot\ldots\cdot3\cdot2\cdot1}$$
$$= \frac{1}{2^n}\cdot n\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(n-k+1)$$
$$\leq \frac{1}{2^n}n^k$$
$$\leq \frac{1}{2^n}n^{a\sqrt{n}}$$
$$= (\frac{n^a}{2^{\sqrt{n}}})^{\sqrt{n}}$$

$\lim_{n\to\infty}(\frac{n^a}{2^{\sqrt{n}}}) = 0$

It can also be thought of intuitively, as the growing number of training data in a k-NN algorithm leads to fewer miss-classification, because of the profusion of labeled points near the test point x_i.

6. **Gaussian Mixture** [10 points] Mélange de Gaussiennes

Let $\mu_1, \mu_2 \in \mathbb{R}^2$, and let Σ_1, Σ_2 be two 2x2 positive definite matrices (i.e. symmetric with positive eigenvalues).
We now introduce the two following pdf over $\mathbb{R}^2$:
Soit $\mu_1, \mu_2 \in \mathbb{R}^2$, et soit Σ_1, Σ_2 deux matrices 2x2 positives définies (i.e. symmétriques avec des valeurs propres strictement positives).
On définie maintenant les deux fonctions de densités de probabilités suivantes sur $\mathbb{R}^2$:

$$f_{\mu_1,\Sigma_1}(\mathbf{x}) = \frac{1}{2\pi\sqrt{det(\Sigma_1)}} e^{-\frac{1}{2}(\mathbf{x}-\mu_1)^T \Sigma_1^{-1}(\mathbf{x}-\mu_1)}$$

$$f_{\mu_2,\Sigma_2}(\mathbf{x}) = \frac{1}{2\pi\sqrt{det(\Sigma_2)}} e^{-\frac{1}{2}(\mathbf{x}-\mu_2)^T \Sigma_2^{-1}(\mathbf{x}-\mu_2)}$$

These pdf correspond to the multivariate Gaussian distribution of mean μ_1 and covariance Σ_1, denoted $\mathcal{N}_2(\mu_1, \Sigma_1)$, and the multivariate Gaussian distribution of mean μ_2 and covariance Σ_2, denoted $\mathcal{N}_2(\mu_2, \Sigma_2)$.
Ces fonctions de densité de probabilités correspondent à la distribution gaussienne multivariée de centre μ_1 et covariance Σ_1, notée $\mathcal{N}_2(\mu_1, \Sigma_1)$, et à la gaussienne multivariée de centre μ_2 et covariance Σ_2, notée $\mathcal{N}_2(\mu_2, \Sigma_2)$.

We now toss a balanced coin Y, and draw a random variable X in $\mathbb{R}^2$, following this process : if the coin lands on tails ($Y = 0$) we draw X from $\mathcal{N}_2(\mu_1, \Sigma_1)$, and if the coin lands on heads ($Y = 1$) we draw X from $\mathcal{N}_2(\mu_2, \Sigma_2)$.
On lance maintenant une pièce équilibrée Y, et on tire une variable aléatoire X dans $\mathbb{R}^2$, en suivant le procédé suivant : si la pièce atterit sur pile ($Y = 0$), on tire X selon $\mathcal{N}_2(\mu_1, \Sigma_1)$, et si la pièce atterit sur face ($Y = 1$), on tire X selon $\mathcal{N}_2(\mu_2, \Sigma_2)$.

Calculate $\mathbb{P}(Y = 0|X = \mathbf{x})$, the probability that the coin landed on tails given $X = \mathbf{x} \in \mathbb{R}^2$, as a function of μ_1, μ_2, Σ_1, Σ_2, and $\mathbf{x}$. Show all the steps of the derivation.
Calculez $\mathbb{P}(Y = 0|X = \mathbf{x})$, la probabilité que la pièce atterisse sur pile

sachant $X = \mathbf{x} \in \mathbb{R}^2$, en fonction de μ_1, μ_2, Σ_1, Σ_2, et $\mathbf{x}$. Montrez toutes les étapes du calcul.

Answer 6

We can express our problem as : given the data point x, what is the probability it came from the first Gaussian, i.e. $\mathcal{N}_2(\mu_1, \Sigma_1)$.
This can also be expressed as :

$$p(z_{nk} = 1 | x_n)$$

In the above expression, z is a latent variable that takes two possible values; 1 when x came from the Gaussian k and 0 otherwise.
We can also say that the overall probability of observing a point that comes from the Gaussian k (Gaussian 1) is equal to the mixing coefficient of that Gaussian (denoted π).
This all translates to :

$$\pi_k = p(z_k = 1)$$

If we denote $\mathbf{z}$ to be the set of all possible latent variables z we have in our case :

$$\mathbf{z} = \{z_1, z_2\}$$

With z occurring independently we get :

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Hence, we can declare :

$$p(x_n | \mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(x_n | \mu_k, \Sigma_k)^{z_k}$$

From the product rule we can also state :

$$p(x_n, z) = p(x_n | z) p(z)$$

And from marginalization we get :

$$\begin{aligned} p(x_n) &= \sum_{k=1}^{K} p(x_n|z)p(z) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \end{aligned}$$

Finally, from Bayes' rule we have :

$$\begin{aligned} &p(z_k = 1|x_n) = \\ &\qquad = \frac{p(x_n|z_k = 1)p(z_k = 1)}{\sum_{j=1}^{K} p(x_n|z_j = 1)p(z_j = 1)} \\ &\qquad = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \\ &\qquad = \frac{\frac{1}{2\pi\sqrt{det(\Sigma_1)}} e^{-\frac{1}{2}(\mathbf{x}-\mu_1)^T \Sigma_1^{-1} (\mathbf{x}-\mu_1)} \cdot 0.5}{\frac{1}{2\pi\sqrt{det(\Sigma_2)}} e^{-\frac{1}{2}(\mathbf{x}-\mu_2)^T \Sigma_2^{-1} (\mathbf{x}-\mu_2)} \cdot 0.5 + \frac{1}{2\pi\sqrt{det(\Sigma_1)}} e^{-\frac{1}{2}(\mathbf{x}-\mu_1)^T \Sigma_1^{-1} (\mathbf{x}-\mu_1)} \cdot 0.5} \end{aligned}$$

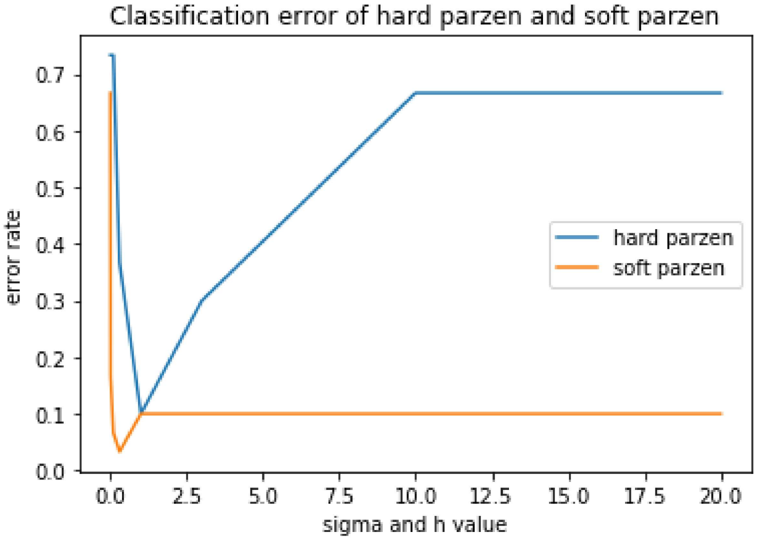
IFT 6390, Homework 1

Marc-Antoine Provost

Question 5

In [8]:

```
import numpy as np
import matplotlib.pyplot as plt
iris = np.loadtxt('iris.txt')

radius_hard = [0.001, 0.01, 0.1, 0.3, 1.0, 3.0, 10.0, 15.0, 20.0]
validation_error_hard = [0.7333333333333334, 0.7333333333333334, 0.7333333333333334, 0.3666666666666667, 0.09999999999999998, 0.30000000000000004, 0.6666666666666667, 0.6666666666666667, 0.6666666666666667]
validation_error_soft = [1.3333333333333335, 0.33333333333333326, 0.1333333333333333, 0.06666666666666665, 0.19999999999999996, 0.19999999999999996, 0.19999999999999996, 0.19999999999999996, 0.19999999999999996]

plt.plot(radius_hard, validation_error_hard, label = "hard parzen")
plt.plot(radius_hard, np.divide(validation_error_soft, 2), label = "soft parzen")
plt.xlabel('sigma and h value')
plt.ylabel('error rate')
plt.legend()
plt.title("Classification error of hard parzen and soft parzen")
plt.show()
```



As we increase the lenght in the k-NN algorithm with hard neighbourhood, we consider a larger portion of neighbors, thus the increase in the error rate. To illustrate this point, let us consider an example with a small lenght. In this case, our model will have very low bias, because it only considers a small window of points to predict the class of our new test point (but at the same time has high variance because we are sensitive to outliers). When selecting a wider lenght, all test points will belong to the same class; the majority class. Thus, increasing our hyperparameter h will result in higher bias, but smaller variance. I.e. we will have a larger classification error rate as we increase our hypeparameter.

The same can be said for the hyperparameter sigma in the k-NN algorithm with kernel density estimation. In this case, we are now taking a weighted vote instead of computing the vote of all neighbors in a specified lenght. With a small sigma, we give more weight to the points that are close to our test point, resulting in a smaller error rate. As sigma increase, the weight is more equally distributed, thus increasing the error rate.

Question 7

As the hyperparameter h increases, so does the running time for the method hard_parzen. By increasing our hyperparameter h, we include a larger number of points to compute the number of predictions and the error rate on.

When increasing the hyperparameter sigma, the running time seems to decrease a bit. This can be due to the fact that with a small sigma, more computation is needed to calculate different weights, while with a bigger sigma the weights are more or less similar.

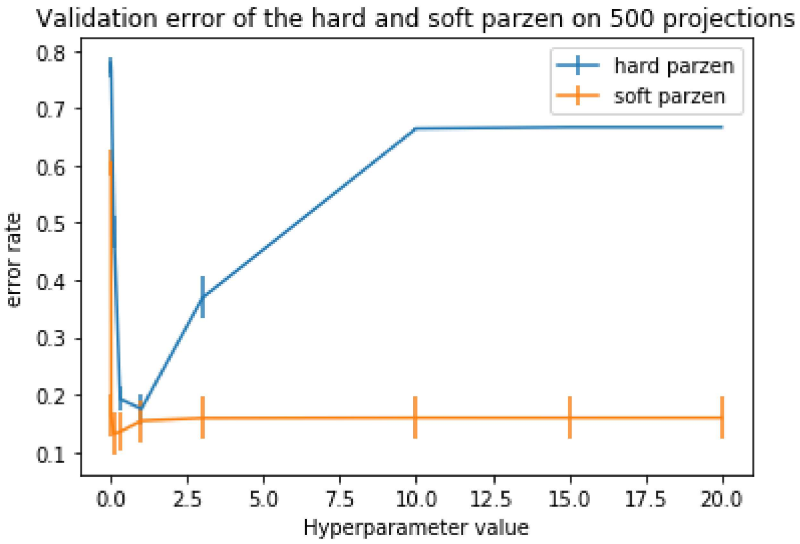
Question 9

In [ ]:

```
N = 500
splitted_data = split_dataset(iris)
training = splitted_data[0]
validation = splitted_data[1]
test = splitted_data[2]
x_train = splitted_data[0][:, 0:4]
y_train = splitted_data[0][:, 4]
x_val = splitted_data[1][:, 0:4]
y_val = splitted_data[1][:, 4]
validation_error_hard = np.zeros((500, 9))
validation_error_soft = np.zeros((500, 9))
hyperparameter = [0.001, 0.01, 0.1, 0.3, 1.0, 3.0, 10.0, 15.0, 20.0]

for i in range(N):
    random_matrix = np.random.normal(0, 1, (4, 2))
    modified_training = random_projections(x_train, random_matrix)
    modified_validation = random_projections(x_val, random_matrix)
    a = ErrorRate(modified_training, y_train, modified_validation, y_val)
    for (index, value) in enumerate(hyperparameter):
        validation_error_hard[i, index] = a.hard_parzen(value)

for i in range(N):
    random_matrix = np.random.normal(0, 1, (4, 2))
    modified_training = random_projections(x_train, random_matrix)
    modified_validation = random_projections(x_val, random_matrix)
    a = ErrorRate(modified_training, y_train, modified_validation, y_val)
    for (index, value) in enumerate(hyperparameter):
        validation_error_soft[i, index] = a.soft_parzen(value)

avg_val_hard = np.mean(validation_error_hard, axis=0)
avg_val_soft = np.mean(validation_error_soft, axis=0)
np.divide(avg_val_soft, 2)
np.std(validation_error_hard, axis = 0)
np.std(validation_error_soft, axis = 0)
```

```
In [32]: avg_val_soft = [0.60446667, 0.163, 0.13146667, 0.1354, 0.1546, 0.15906667, 0.1
         5966667, 0.1596, 0.15966667]
         avg_val_hard = [0.77413333, 0.76646667, 0.48446667, 0.19266667, 0.1752,0.36926
         667, 0.66426667, 0.66666667, 0.66666667]
         hyperparameter = [0.001, 0.01, 0.1, 0.3, 1.0, 3.0, 10.0, 15.0, 20.0]
         hard_std = [6.83747842e-02, 7.07786534e-02, 1.42870633e-01, 1.02724032e-01, 1.
         31823518e-01, 1.87153781e-01, 1.03732562e-02, 1.33226763e-15, 1.33226763e-15]
         error_hard = np.dot(hard_std, 0.2)
         soft_std = [0.11348813, 0.18681213, 0.17870666, 0.17186967, 0.18206597, 0.1861
         4838, 0.18669233, 0.18669233, 0.18669233]
         error_soft = np.dot(soft_std, 0.2)


         plt.errorbar(hyperparameter, avg_val_hard, yerr = error_hard, label= "hard par
         zen")
         plt.errorbar(hyperparameter, avg_val_soft, yerr = error_soft, label= "soft par
         zen")
         plt.xlabel("Hyperparameter value")
         plt.ylabel("error rate")
         plt.title("Validation error of the hard and soft parzen on 500 projections")
         plt.legend()
```

```
Out[32]: <matplotlib.legend.Legend at 0xcb49c10>
```



The results are quite similar as the ones from question 5. This can be explained by our dimensionality reduction technique (random projection). Indeed, we reduced our number of features and it still captured the essence of our initial four features. Dimensionality reduction helps avoid the curse of dimensionality and reduce time and storage space.