Faire apprendre à des machines : les plus proches voisins

Rémi Lajugie

L'actualité informatique de l'année 2016 a été marquée par deux événements importants.

Le 14 février 2016, pour la première fois depuis leur mise en service et plus de trois millions de kilomètres roulés, une voiture sans chauffeur de Google a eu son premier accrochage (sans gravité) avec un bus.

Le 15 mars 2016, un ordinateur a battu pour la première fois les plus grands champions au jeu de go, jeu pour lequel on pensait qu'il faudrait plus d'un siècle pour qu'un ordinateur réussisse à y jouer.

A leur manière, chacun de ces événements marque d'indéniables progrès dans ce que l'on a coutume d'appeler l'"intelligence artificielle" : c'est à dire que des programmes informatiques sont capables de résoudre des tâches apparemment très complexes et qui nécessitent, pour les êtres humains, un apprentissage souvent long. Dans le premier cas, les voitures automatiques ont réussi à rouler trois millions de kilomètres sur route sans le moindre accrochage, dans l'autre un ordinateur a battu le meilleur joueur de go au monde. Ces prouesses ont été rendues possibles par deux phénomènes. D'un côté, la loi de Moore qui régit l'informatique n'a cessé de s'appliquer : la puissance des ordinateurs croît de manière exponentielle [1] D'un autre côté, l'approche dite de l'apprentissage automatique, a offert un nouveau cadre pour l'intelligence artificielle. Cette approche vise à faire apprendre à des machines quels sont leurs meilleurs paramètres de fonctionnement. C'est à dire que, plutôt que de demander à un ingénieur ou un technicien d'ajuster les paramètres d'un programme, c'est un programme appelé "programme d'apprentissage" qui s'en charge.

Prenons l'exemple de la voiture sans chauffeur : on imagine fort bien qu'il y a un grand nombre de paramètres à ajuster pour la conduite autonome : distance de sécurité à respecter en fonction de la vitesse, réduction de la vitesse en fonction de la tenue de route, de la pression des pneus, passage des rapports de vitesse, manière de faire un créneau, moments opportuns pour allumer les phares etc.

Il est inimaginable qu'un être humain (ni même mille) ait pu prendre en compte de lui même toutes les situations possibles. Il faut la puissance de calcul d'un ordinateur pour pouvoir le faire.

Pour réaliser leurs voitures automatiques, Google ou Uber ont eu recours à de nombreuses méthodes dites "d'apprentissage automatique". Toutes reposent sur une idée semblable. Il s'agit de montrer à l'ordinateur des exemples de conduite parfaite et d'autres très mauvais ; il doit alors en déduire quels les paramètres qui font une bonne conduite.

De manière plus abstraite, l'apprentissage automatique vise à faire apprendre une règle de décision permettant de réagir convenablement lorsqu'une nouvelle situation se présente. Comme il s'agit de s'appuyer sur des exemples passés pour déterminer une conduite future, l'analogie avec l'apprentissage chez les humains est claire, et c'est de là que vient la dénomination "apprentissage artificiel" ("machine learning" en anglais).

La figure 1 représente schématiquement le principe de l'apprentissage : on considère une méthode (par exemple une méthode pour conduire) qui repose sur des paramètres. Puis, plutôt que de les ajuster un par un à la main, on va utiliser un algorithme d'apprentissage en lui montrant des exemples de ce qu'il faut faire et ne pas faire (par exemple une conduite prudente et une autre, très dangereuse) il va ajuster les paramètres de la méthode pour qu'elle produise un résultat conforme à celui que l'on attend.

Dans cette note, nous allons explorer le fonctionnement d'une des plus anciennes mais aussi de l'une des plus efficaces de ces méthodes d'apprentissage : la méthode des plus proches voisins. Cette méthode n'a pas été utilisée dans le cas de la voiture sans chauffeur. Néanmoins les concepts fondamentaux que nous allons dégager (sous et sur apprentissage) sont généraux et concernent toutes les méthodes d'apprentissage.

Notons que la méthode que nous allons étudier est très utilisée dans un domaine qui nous concerne tous : la recommandation de produits. En effet, qui n'a pas été étonné du caractère extrêmement ciblé des

1. Gordon Moore, fondateur d'Intel, avait énoncé une loi empirique, jamais démentie jusqu'en 2016 : la puissance de calcul des ordinateurs double tous les deux ans.

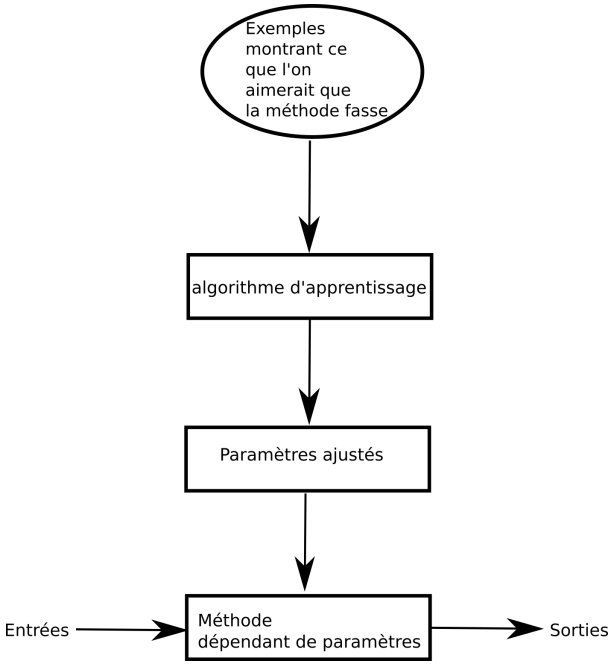

FIGURE 1 – Schéma décrivant la place d'une procédure d'apprentissage.

publicités sur Internet ? Le principe du ciblage des publicités est de récupérer des données sur les utilisateurs puis d'utiliser un programme d'apprentissage automatique pour prédire quels sont les produits les plus susceptibles d'intéresser le consommateur. Les programmes les plus efficaces en ce domaine utilisent assez souvent la méthode des plus proches voisins.

1 Le principe de la méthode

1.1 Problème considéré

Plutôt que de la présenter dans un cadre trop général, présentons la méthode dans un cas particulier à partir duquel il sera très facile d'extrapoler le cas général.

On considère deux phénomènes, chacun générant des points dans le plan. Le premier phénomène génère des points "bleus" et l'autre des points "rouges". La situation est représentée par la figure 2. En d'autres termes nous considérons que nos points peuvent avoir deux modalités (dans un contexte médical cela pourrait être "sain" ou "malade", dans un contexte de conduite autonome "conduite acceptable", "conduite dangereuse") et ont deux dimensions qui peuvent représenter diverses caractéristiques (la vitesse et la distance au véhicule qui précède dans le cas de la conduite automatique).

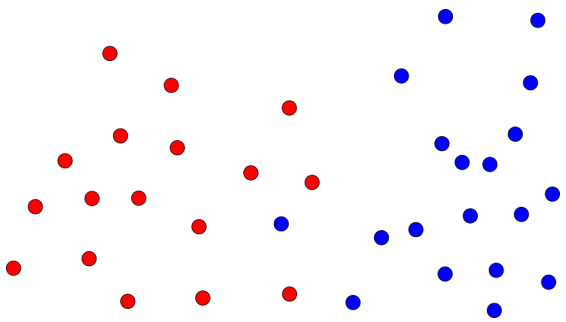

FIGURE 2 – Echantillon d'apprentissage en deux dimensions.

La façon dont les points sont générés points a peu d'importance.

Vocabulaire important :

Les coordonnées des points s'appellent les *descripteurs*. Le caractère bleu ou rouge du point s'appelle l'*étiquette* ou le *label*. On va supposer que l'on dispose d'un certain nombre de points pour lesquels

l'étiquette, bleue ou rouge est connue. Ces points forment **l'ensemble d'apprentissage**. C'est à partir d'eux qu'on va travailler.

Objectif. Le but de la méthode est, à partir de cet ensemble d'apprentissage de déterminer une règle permettant de dire quelle est l'étiquette de points inconnus (pour lesquels on connaît les coordonnées/descripteurs mais pas l'étiquette). En d'autres termes, si, comme dans la figure 4, un nouveau point est présenté, on veut pouvoir dire s'il faut le considérer comme étant un point bleu ou un point rouge.

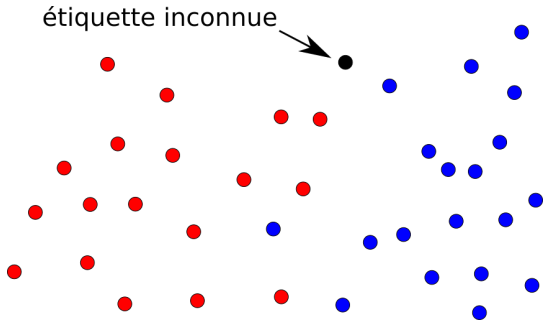

FIGURE 3 – Echantillon d'apprentissage en deux dimensions

FIGURE 4 – Le noir de la figure signifie que l'on ne connaît pas l'étiquette du point.

1.2 Le plus proche voisin

La règle de décision du plus proche voisin est fondée sur le principe suivant : à un nouveau point présenté, on va affecter la même étiquette que le point de l'ensemble d'apprentissage le plus proche du nouveau point. Si le nouveau point est proche d'un point bleu, il est raisonnable de penser que ce nouveau point est bleu.

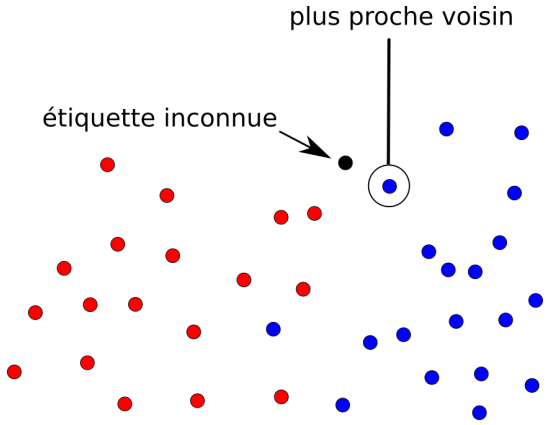

FIGURE 5 – Illustration du fonctionnement de la règle du plus proche voisin sur un exemple simple : on cherche le point le plus proche du nouveau point, c'est celui qui est entouré et on affecte au nouveau point l'étiquette du point le plus proche : ici le bleu.

Frontière de décision. On appelle *frontière de décision* associée à une règle de décision, la frontière entre la zone dans laquelle où l'étiquette attribuée est la bleue et celle où il s'agit de l'étiquette rouge.

1.3 Les k-plus proches voisins

La règle des k plus proches voisins est une extension de la règle du plus proche voisin. Cette fois, au lieu de ne considérer que le plus proche voisin du point dont on souhaite déterminer l'étiquette,

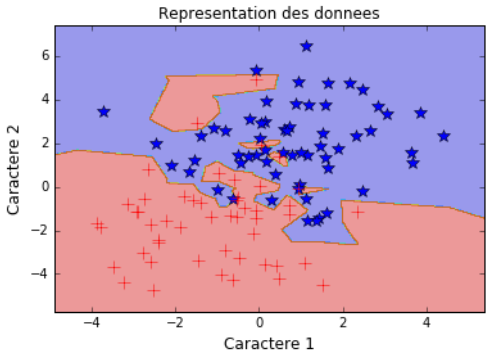

FIGURE 6 – Frontière de décision associée à la règle du plus proche voisin sur un exemple de données traitées à l'aide du code Python en annexe.

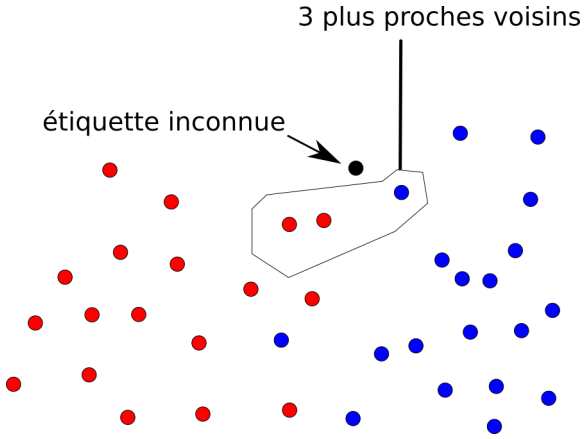

FIGURE 7 – Illustration du fonctionnement de la règle des trois plus proches voisins sur le même exemple que celui sur lequel on a travaillé plus haut. Il faut noter que la décision finale est d'affecter l'étiquette rouge au point, alors que précédemment on affectait l'étiquette bleue.

on regarde les k plus proches voisins et on affecte au nouveau point l'étiquette majoritaire parmi ses voisins [2].

Frontière de décision. Dans la figure 8, on a représenté un exemple de frontière de décision associée à la règle de décision des dix plus proches voisins sur un exemple. Il est intéressant de comparer la frontière de décision présentée dans cette figure avec celle de la règle du plus proche voisin dans la partie précédente. La règle de décision du plus proche voisin engendrait une frontière de décision beaucoup plus "compliquée" (en un sens que nous ne préciserons pas ici) que celle des 10 plus proches voisins.

Désormais une question se pose : quelle règle de plus proches voisins convient le mieux ? On a vu sur notre exemple que la frontière de décision associée à la règle du plus proche voisin semblait plus "compliquée" que celle associée à la règle des trois plus proches voisins. Est-ce que cette complexité est une bonne ou une mauvaise chose ? Ou encore est elle totalement neutre ? C'est ce que nous allons discuter dans la suite.

2 L'apprentissage du nombre de voisins

Dans le cadre des plus proches voisins, la procédure d'apprentissage à proprement parler consiste en l'ajustement du nombre de plus proches voisins pris en compte dans la règle des k-plus proches voisins.

Pour pouvoir comparer les règles des plus proches voisins, il faut définir un critère.

2. Il convient bien entendu de traiter de manière spécifique le cas où k est pair.

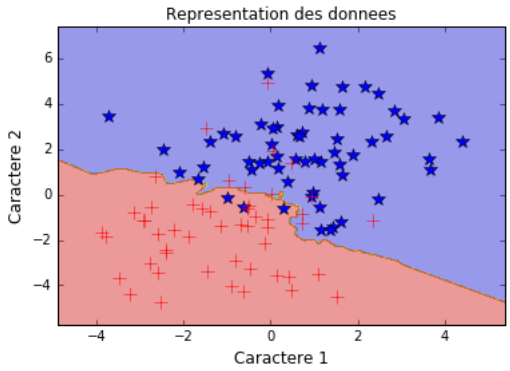

FIGURE 8 – Frontière de décision associée à la règle des dix plus proches voisins sur un exemple de données traitées à l'aide du code Python en annexe.

2.1 Sur-apprentissage et sous-apprentissage

Le premier critère qui vient à l'esprit est celui du nombre d'erreurs commises par rapport aux exemples d'apprentissage. C'est à dire que chacun de nos exemples de l'ensemble d'apprentissage est présenté de nouveau, on lui applique la règle des k-plus proches voisins, on regarde si l'étiquette affectée est la bonne ou pas.

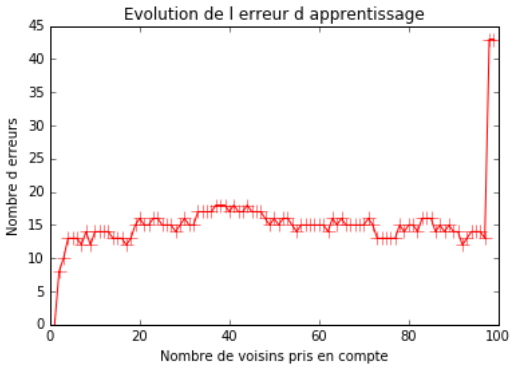

FIGURE 9 – Exemple de courbe d'erreur obtenu sur l'ensemble d'apprentissage utilisé dans les figures de la partie précédente (courbe obtenue à l'aide du programme Python en annexe).

Bien évidemment, c'est la règle du plus proche voisin qui donne les meilleurs résultats puisque si un point fait partie de l'ensemble d'apprentissage, son plus proche voisin est lui-même et donc il est tout naturel que la bonne étiquette va lui être attribuée. La prise en compte de plusieurs voisins conduit naturellement à faire plus d'erreurs notamment sur les points proches de la frontière de décision de la règle du plus proche voisin.

Cependant, la performance qui nous intéresse réellement n'est pas celle de la méthode appliquée aux points d'apprentissage (après tout, pour ceux là on connaît déjà la bonne étiquette) mais les performances de la méthode sur des points que l'on a jamais rencontré. Pour cela imaginons un autre ensemble de points dont on connaît l'étiquette mais qui n'a pas été utilisé comme ensemble d'apprentissage. Cet ensemble va nous servir de test et nous l'appellerons désormais l'*ensemble de test*.

Regardons la performance des règles des k-plus proches voisin sur cet ensemble. Un exemple de résultat obtenu est donné par la figure 10.

On constate que l'erreur sur cet échantillon de test ne se comporte pas comme l'erreur d'apprentissage. Lorsque le nombre de voisins pris en compte augmente, l'erreur augmente mais lorsque le nombre de voisins pris en compte diminue, l'erreur augmente aussi !

Que se passe-t-il ?

Dans le cas où le nombre de voisins augmente, la précision de la prédiction diminue [3], c'est le phénomène de **sous-apprentissage**, on n'utilise pas assez les informations des données.

3. Dans le cas limite où l'on considère autant de voisins qu'il y a de points dans l'échantillon d'apprentissage pour faire

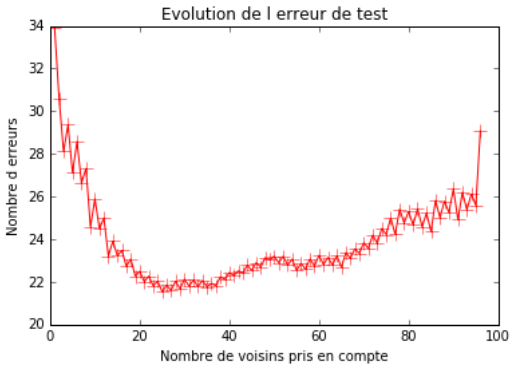

FIGURE 10 – Exemple de courbe d'erreur obtenu sur l'ensemble d'apprentissage utilisé dans les figures de la partie précédente (courbe obtenue à l'aide du programme Python en annexe).

Dans le cas où le nombre de voisins diminue, la performance diminue car la frontière de décision devient tellement compliquée qu'elle est extrêmement sensible à de petites perturbations (erreurs de mesures par exemple), en d'autres termes on colle trop aux données. C'est le phénomène de **sur-apprentissage** : on a tellement bien appris sa leçon (comme en apprenant par coeur sans comprendre) que l'on est incapable de généraliser lorsqu'une nouveauté se présente.

Il y a donc un *compromis* à trouver. Il faut une règle de décision conduisant à une frontière de décision ni trop complexe ni trop simple.

2.2 Sélection du nombre de voisins par validation

Une des procédures les plus efficaces pour choisir le bon nombre de voisins est la suivante : de notre ensemble de points dont on connaît les étiquettes, on fait deux groupes : d'un côté l'ensemble d'apprentissage et de l'autre l'ensemble de test que nous allons utiliser pour valider notre procédure.

Puis pour chaque nombre de voisin possible on calcule les performances de la règle des plus proches voisins sur l'ensemble de test. Au bout du compte, on choisira le nombre de voisins donnant le moins d'erreurs sur l'ensemble de test.

Cette procédure de sélection du nombre de plus proche voisin se fait automatiquement, simplement en recherchant le minimum de l'erreur de test. C'est ce genre de procédure que l'on appelle "procédure d'apprentissage automatique". Ces procédures cherchent à éviter les problèmes de sous et de sur apprentissage en choisissant des règles de décision qui généralisent bien à de nouvelles données. Ce principe, en apparence simple, est la brique fondamentale qui a conduit aux récents succès de l'intelligence artificielle que nous avons évoqué dans l'introduction de cette note. Toutes les méthodes cherchent à résoudre un problème d'optimisation : minimiser l'erreur de test.

3 Conclusion

Cette brève présentation, sans formalisme mathématique, de la méthode des plus proches voisins a permis de faire sentir le principe de base de toutes les méthodes d'apprentissage : à partir d'exemples d'entraînement, on apprend une règle destinée à être appliquée à de nouveaux points telle que cette règle puisse avoir de bonnes performances dans des situations nouvelles. Cependant, la recherche d'une règle de décision trop complexe mène au phénomène de sur-apprentissage contre lequel il faut développer une stratégie astucieuse de sélection de modèle, c'est l'"apprentissage" de la règle à proprement parler. Nous avons ici proposé de nous intéresser à la méthode dite de "validation". De nombreuses autres méthodes plus efficaces existent, chacune présentant divers avantages et inconvénients.

Reprenons un de nos exemples introductifs : les principes que nous avons étudiés ici sont sous-jacents aux récents algorithmes ayant débouché sur les voitures sans chauffeur. A partir d'exemples de conduites, et par une procédure de validation adéquate, Google ou Uber ont permis à leurs voitures

la prédiction, tous les points auront systématiquement la même étiquette que l'étiquette majoritaire. Par exemple si l'on a 60 points bleus et 50 rouges, la règle des 110 plus proches voisins va systématiquement conduire à donner l'étiquette bleue à un nouveau point.

d'ajuster automatiquement leur manière de conduire de sorte à ce que leurs voitures sans chauffeur fassent le moins d'erreur de conduite possible en situation réelle.

A l'heure actuelle ces méthodes sont appliquées également dans de nombreux autres domaines : citons par exemple le diagnostic médical qui est une question dans laquelle les ordinateurs sont devenus plus performants que les humains, ou encore la question de la reconnaissance automatique d'objets ou d'actions dans une vidéo. Il n'y a pas à douter que ces méthodes vont encore conduire dans les prochaines décennies à de nombreuses et spectaculaires applications.