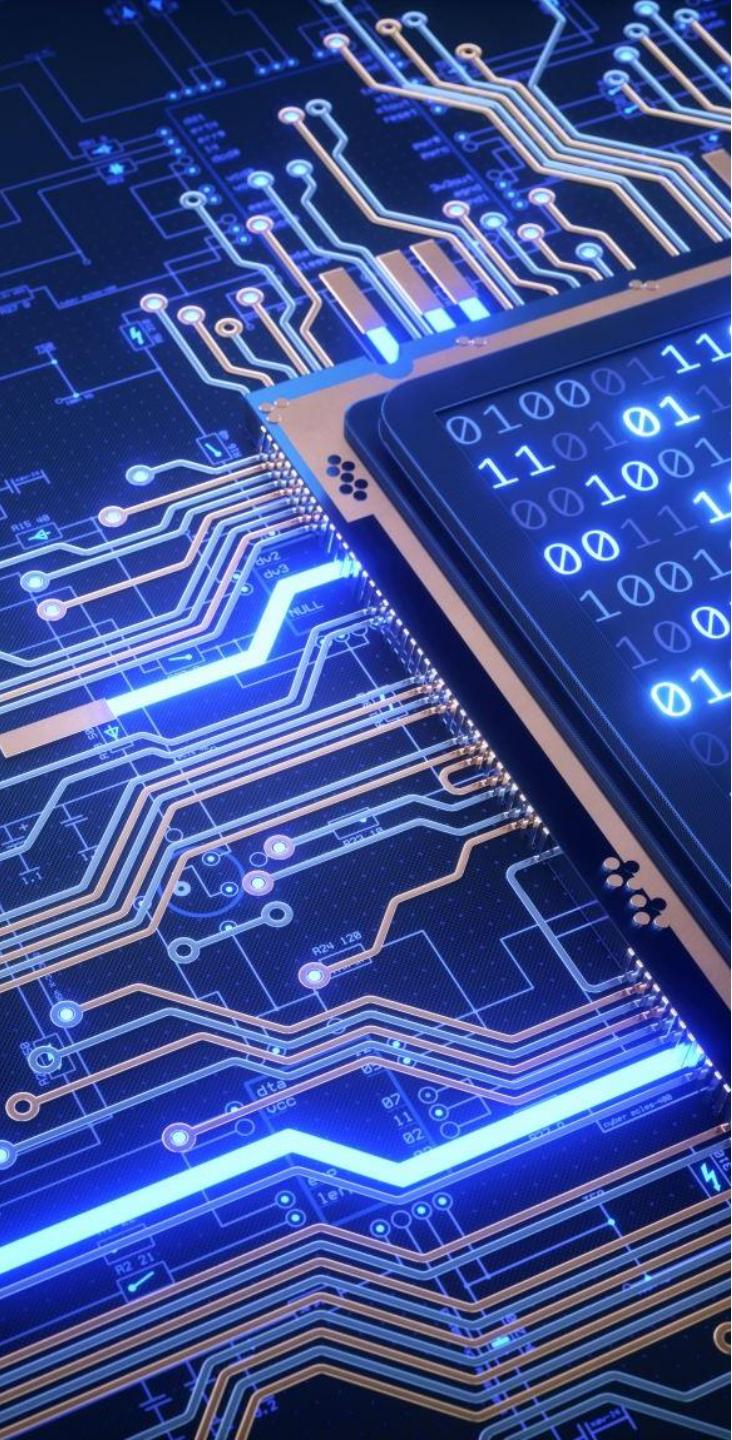


Intelligence Artificielle

Chapitre II

Machine Learning



Plan

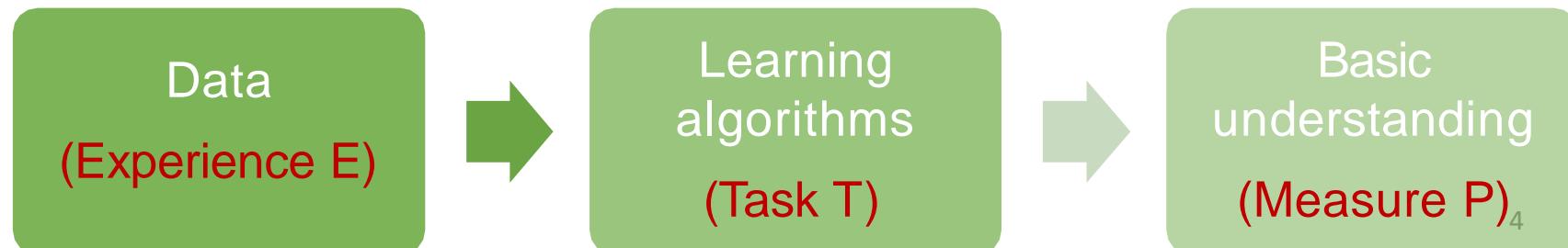
- **Définition de l'apprentissage automatique**
- Types d'apprentissage automatique
- Processus d'apprentissage automatique
- Autres méthodes clés d'apprentissage automatique
- Algorithmes courants d'apprentissage automatique

Introduction

- ✓ Le machine learning est un domaine captivant. Issu de nombreuses disciplines comme les statistiques, l'optimisation, l'algorithme ou le traitement du signal, c'est un champ d'études en mutation constante.
- ✓ Déjà utilisé depuis des décennies dans la reconnaissance automatique de caractères ou les filtres anti-spam, il sert maintenant à protéger contre la fraude bancaire, recommander des livres, films, identifier les visages dans le viseur de notre appareil photo, ou traduire automatiquement des textes d'une langue vers une autre.
- ✓ Dans les années à venir, le machine learning nous permettra vraisemblablement d'améliorer la sécurité routière, la réponse d'urgence aux catastrophes naturelles, le développement de nouveaux médicaments, ou l'efficacité énergétique de nos bâtiments et industries.

Qu'est ce que le machine learning ?

- En français **Apprentissage Automatique**
- Benureau (2015) : « *L'apprentissage est une modification d'un comportement sur la base d'une expérience* ».
- Arthur Samuel (1959) : « *domaine d'études qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés.* »
- Tom Mitchell (1998) : On dit qu'un programme informatique apprend de l'**expérience E** par rapport à une **tâche T** et à une **mesure de performance P**, si sa performance sur T, telle que mesurée par P, s'améliore avec l'expérience



Qu'est ce que le machine learning ?

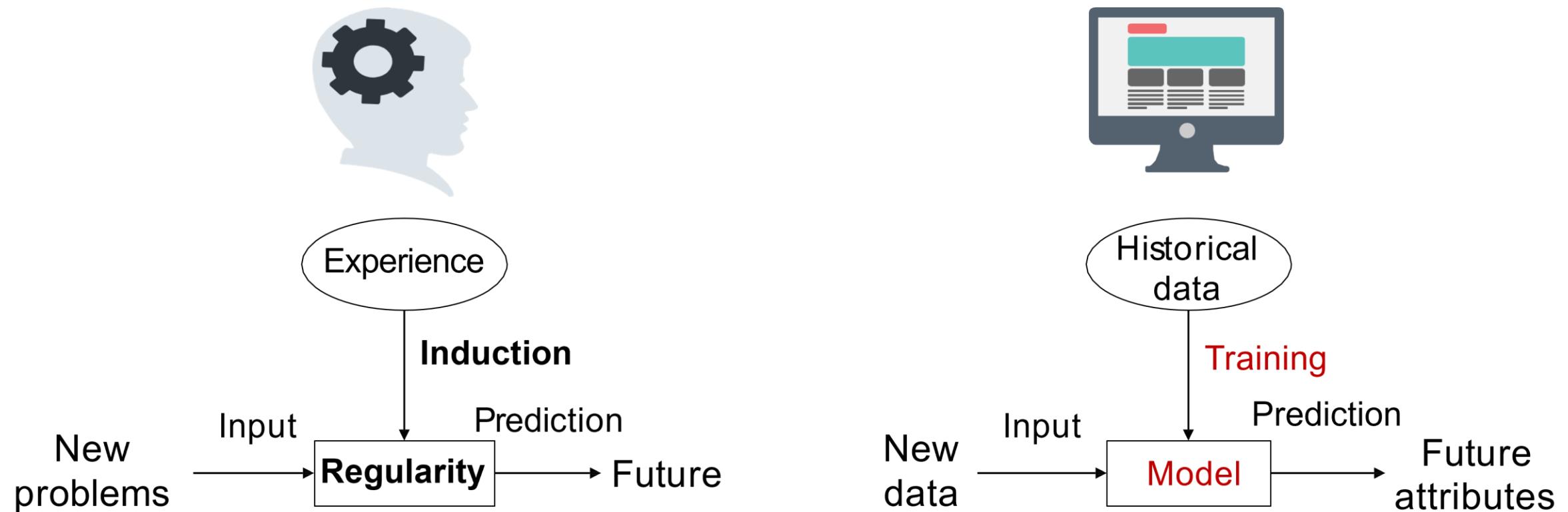


Qu'est ce que le machine learning ?

Supposons que votre programme de messagerie surveille les e-mails que vous marquez ou ne marquez pas comme spam et, en fonction de cela, apprenne à mieux filtrer le spam. Quelle est la tâche T dans ce cadre ?

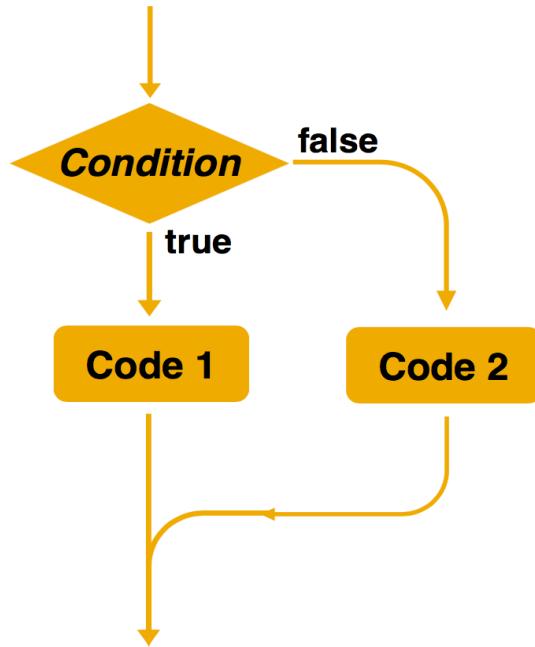
- Classer les e-mails comme spam ou non spam.
- Vous étiquetez les e-mails comme spam ou non spam.
- Le nombre d'e-mails correctement classés comme spam/non spam.
- Aucune des réponses ci-dessus : il ne s'agit pas d'un problème d'apprentissage automatique.

Qu'est ce que le machine learning ?



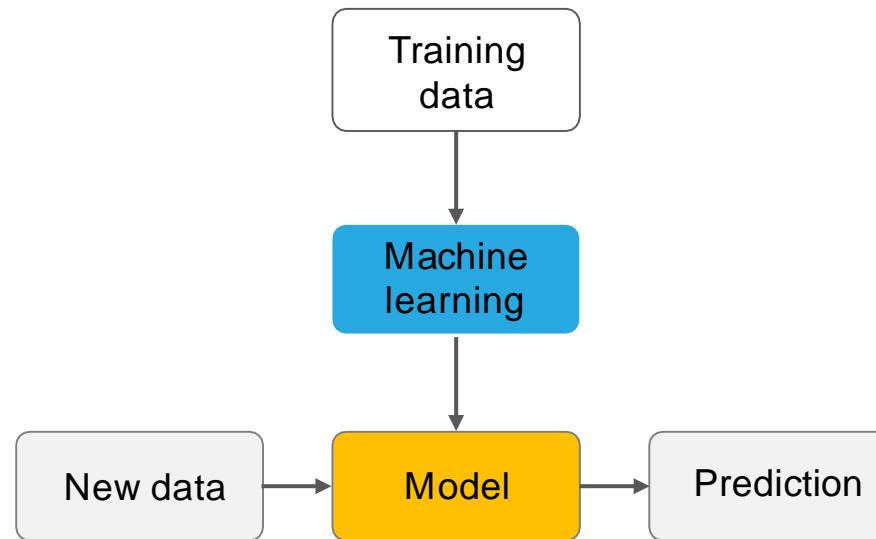
Différences entre ML et les algorithmes traditionnels basés sur des règles

Rule-based algorithms



- La programmation explicite est utilisée pour résoudre des problèmes.
- Les règles peuvent être spécifiées manuellement.

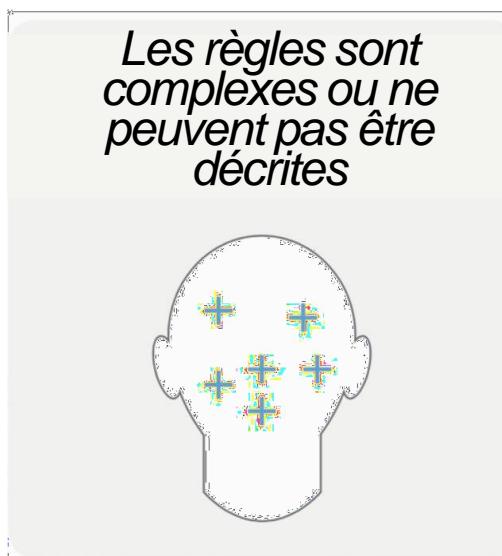
Machine learning



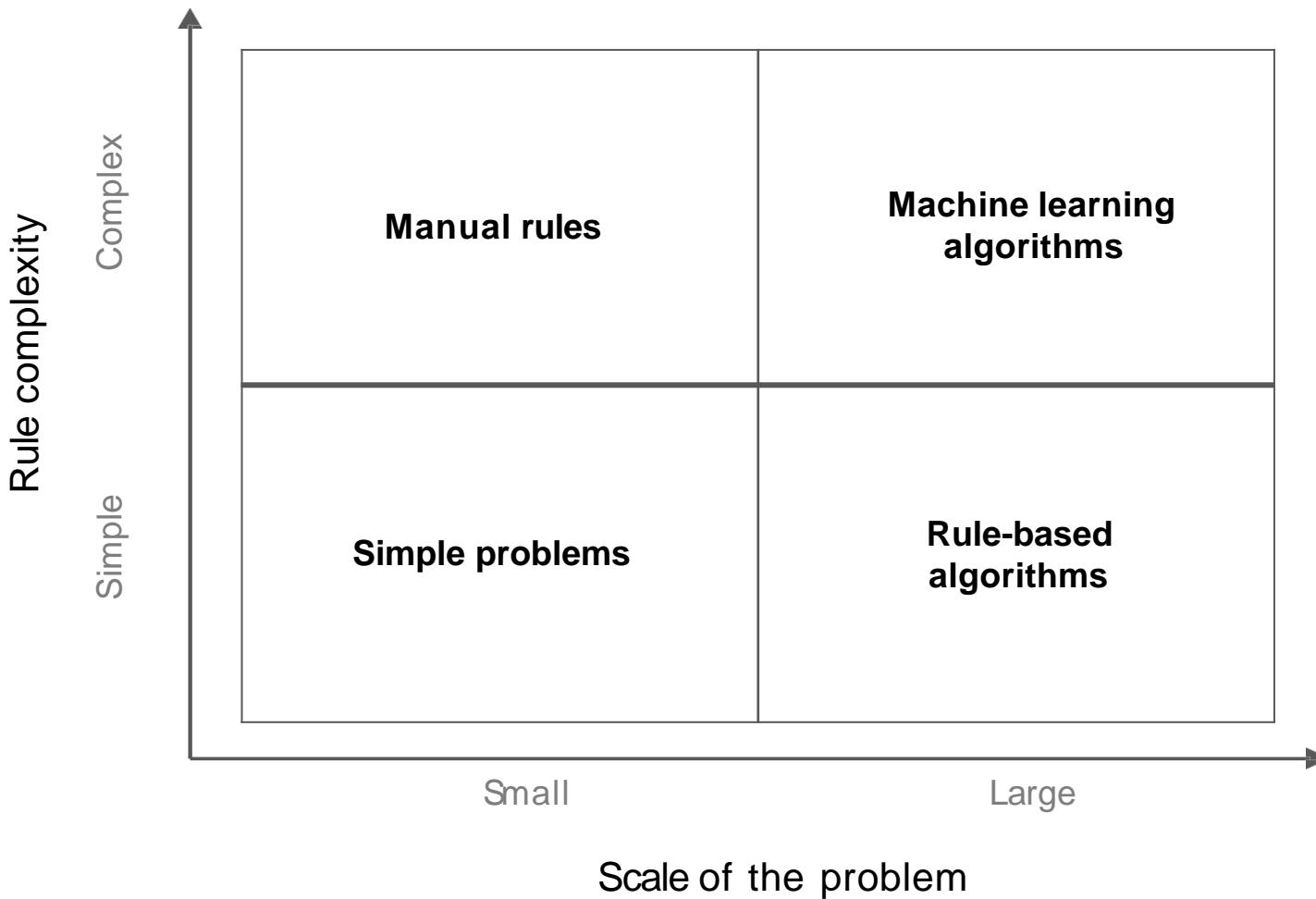
- Des échantillons sont utilisés pour l'entraînement.
- Les règles de prise de décision sont complexes ou difficiles à décrire.
- Les règles sont automatiquement apprises par les machines.

Scénarios d'application du ML

- La solution à un problème est complexe, ou le problème peut impliquer une grande quantité de données sans fonction de distribution de données claire.
- L'apprentissage automatique peut être utilisé dans les scénarios suivants :

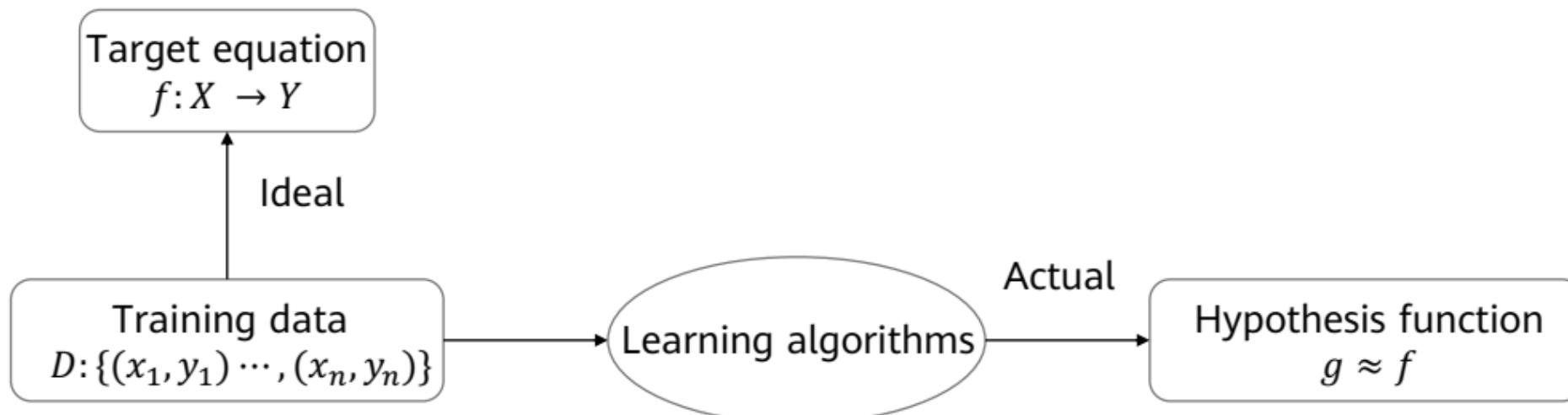


Scénarios d'application du ML



Compréhension rationnelle des algorithmes du ML

- La fonction cible f est inconnue. Les algorithmes d'apprentissage ne peuvent pas obtenir une fonction parfaite f .
- Supposons que la fonction d'hypothèse g se rapproche de la fonction f , mais peut être différente de la fonction f .



Termes et concepts de base

- **Ensemble de données (Dataset)** : fait référence à un ensemble de données utilisées dans les tâches d'apprentissage automatique. Chaque donnée est appelée un échantillon. L'événement ou l'attribut qui reflète la performance ou la nature d'un échantillon dans un certain aspect est appelé une caractéristique (**feature**).
- **Ensemble d'apprentissage (training set)** : fait référence à un ensemble de données utilisé dans le processus d'apprentissage, où chaque échantillon est appelé échantillon d'apprentissage. Le processus d'apprentissage d'un modèle à partir de données est appelé apprentissage (**training**).

Termes et concepts de base

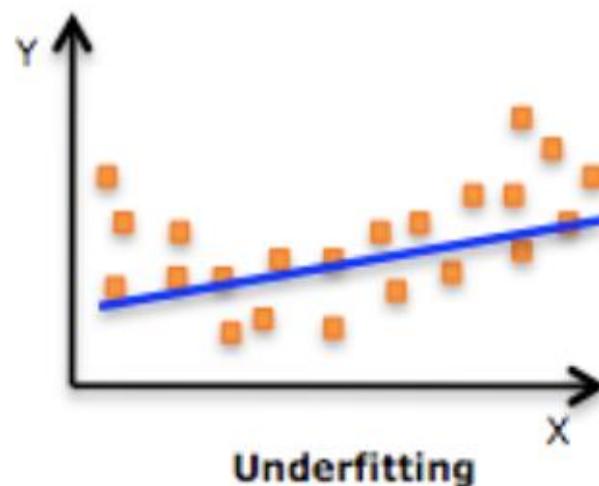
- **Ensemble de test (Test set)** : le test fait référence au processus d'utilisation du modèle appris pour la prédiction. L'ensemble de données utilisé est appelé ensemble de test et chaque échantillon est appelé échantillon de test.
- **Capacité de généralisation (Generalization capability)**: l'objectif de l'apprentissage automatique est que le modèle appris fonctionne bien sur les nouveaux échantillons, et pas seulement sur ceux sur lesquels le modèle a été formé. La capacité de bien performer sur de nouveaux échantillons est appelée capacité de généralisation.

Termes et concepts de base

- **Erreur** : fait référence à la différence entre le résultat de l'échantillon prédit par le modèle appris et le résultat de l'échantillon réel.
 - *Erreur d'apprentissage* : erreur du modèle sur l'ensemble d'apprentissage
 - *Erreur de généralisation* : erreur sur le nouvel échantillon. Évidemment, nous préférons un modèle avec une erreur de généralisation plus faible.
- **Sous-apprentissage (Underfitting)** : se produit lorsque l'erreur d'entraînement est trop importante.
- **Surapprentissage (Overfitting)** : se produit lorsque l'erreur d'apprentissage du modèle appris est faible mais que l'erreur de généralisation est importante (faible capacité de généralisation).

Termes et concepts de base

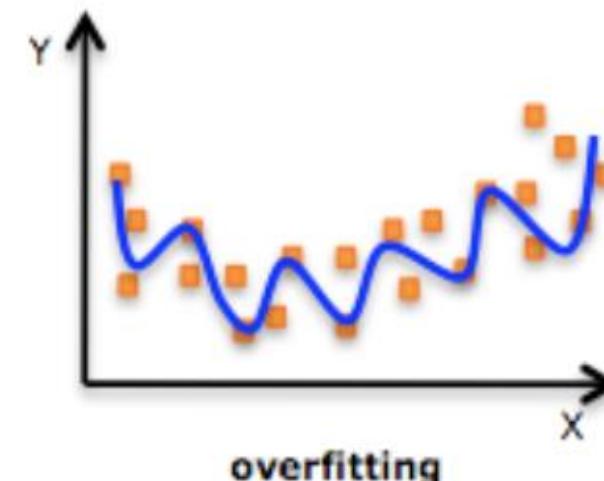
- **Capacité d'un modèle** : fait référence à la capacité de s'adapter à une grande variété de fonctions. Les algorithmes d'apprentissage automatique fonctionnent généralement mieux lorsque leur capacité est appropriée à la véritable complexité de la tâche qu'ils doivent effectuer et à la quantité de données d'entraînement qui leur sont fournies. Les modèles avec une capacité insuffisante sont incapables de résoudre des tâches complexes. Les modèles avec une capacité élevée peuvent résoudre des tâches complexes, mais lorsque leur capacité est supérieure à celle nécessaire pour résoudre la tâche actuelle, ils peuvent être surdimensionnés.



Underfitting



Just right!



overfitting

Principaux problèmes résolus par le ML

Classification

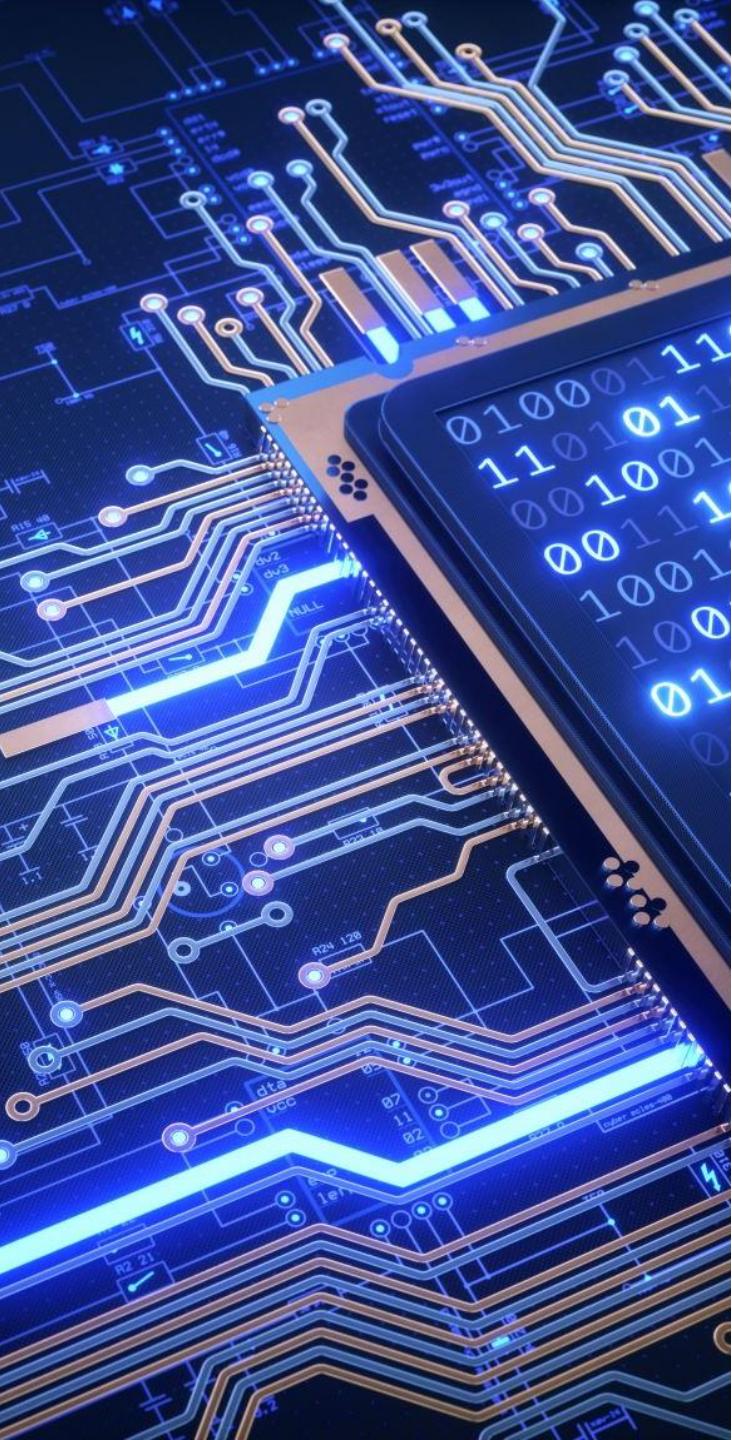
- Un programme informatique doit spécifier à laquelle des k catégories une entrée appartient. Pour accomplir cette tâche, les algorithmes d'apprentissage génèrent généralement une fonction $f : R^n \rightarrow (1, 2, \dots, k)$.

Régression

- Un programme informatique prédit la sortie pour l'entrée donnée. Les algorithmes d'apprentissage génèrent généralement une fonction $f : R^n \rightarrow R$

Clustering

- Une grande quantité de données d'un ensemble non étiqueté est divisée en plusieurs catégories en fonction de la similitude interne des données.



Plan

- Définition de l'apprentissage automatique
- **Types d'apprentissage automatique**
- Processus d'apprentissage automatique
- Autres méthodes clés d'apprentissage automatique
- Algorithmes courants d'apprentissage automatique

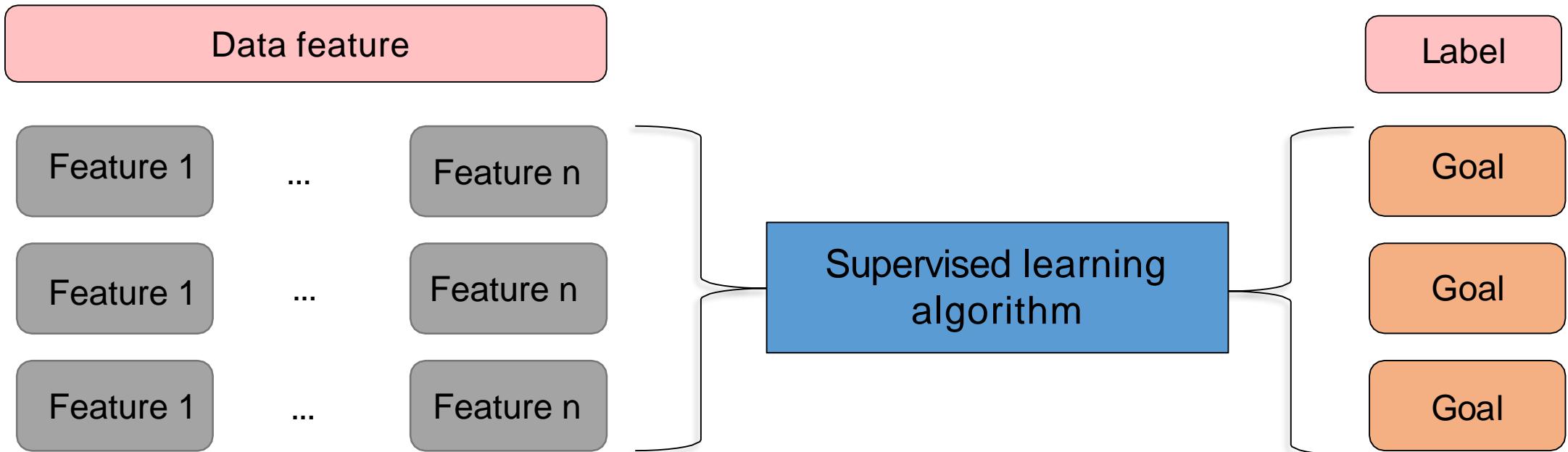
Types d'apprentissage automatique

- **Apprentissage supervisé (Supervised learning)** : Sur la base des échantillons de classes connues, obtenez un modèle optimal avec les performances requises grâce à la formation et à l'apprentissage. Ensuite, le modèle est utilisé pour mapper toutes les entrées aux sorties et effectuer un jugement simple sur les sorties (classification). Ce modèle classe les données inconnues.
- **Apprentissage non supervisé (Unsupervised learning)** : pour les échantillons non étiquetés, l'algorithme d'apprentissage modélise directement les ensembles de données d'entrée, tels que le clustering. Il nous suffit de rassembler des échantillons très similaires, de calculer la similitude de nouveaux échantillons et de les classer par similitude.

Types d'apprentissage automatique

- **Apprentissage semi-supervisé (Semi-supervised learning)** : dans une tâche, un modèle d'apprentissage automatique qui utilise automatiquement une grande quantité de données non étiquetées pour faciliter l'apprentissage direct d'une petite quantité de données étiquetées.
- **Apprentissage par renforcement (Reinforcement learning)** : il s'agit d'un domaine de l'apprentissage automatique qui concerne la manière dont les agents doivent entreprendre des actions dans un environnement pour maximiser une notion de récompense cumulative. La différence entre l'apprentissage par renforcement et l'apprentissage supervisé est le signal de l'enseignant. Le signal de renforcement fourni par l'environnement dans l'apprentissage par renforcement est utilisé pour évaluer l'action (signal scalaire) plutôt que de dire au système d'apprentissage comment effectuer les actions correctes.

Apprentissage supervisé : Classification



Weather	Temperature	Wind Speed
Sunny	Warm	Strong
Rainy	Cold	Fair
Sunny	Cold	Weak

Enjoy Sports
Yes
No
Yes

Apprentissage supervisé : Régression

LUNDI

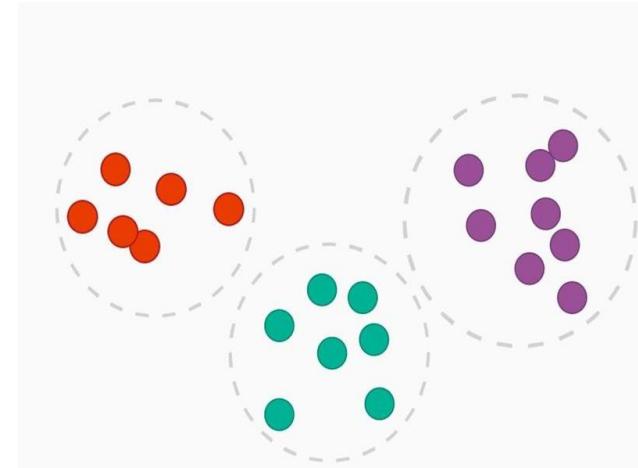
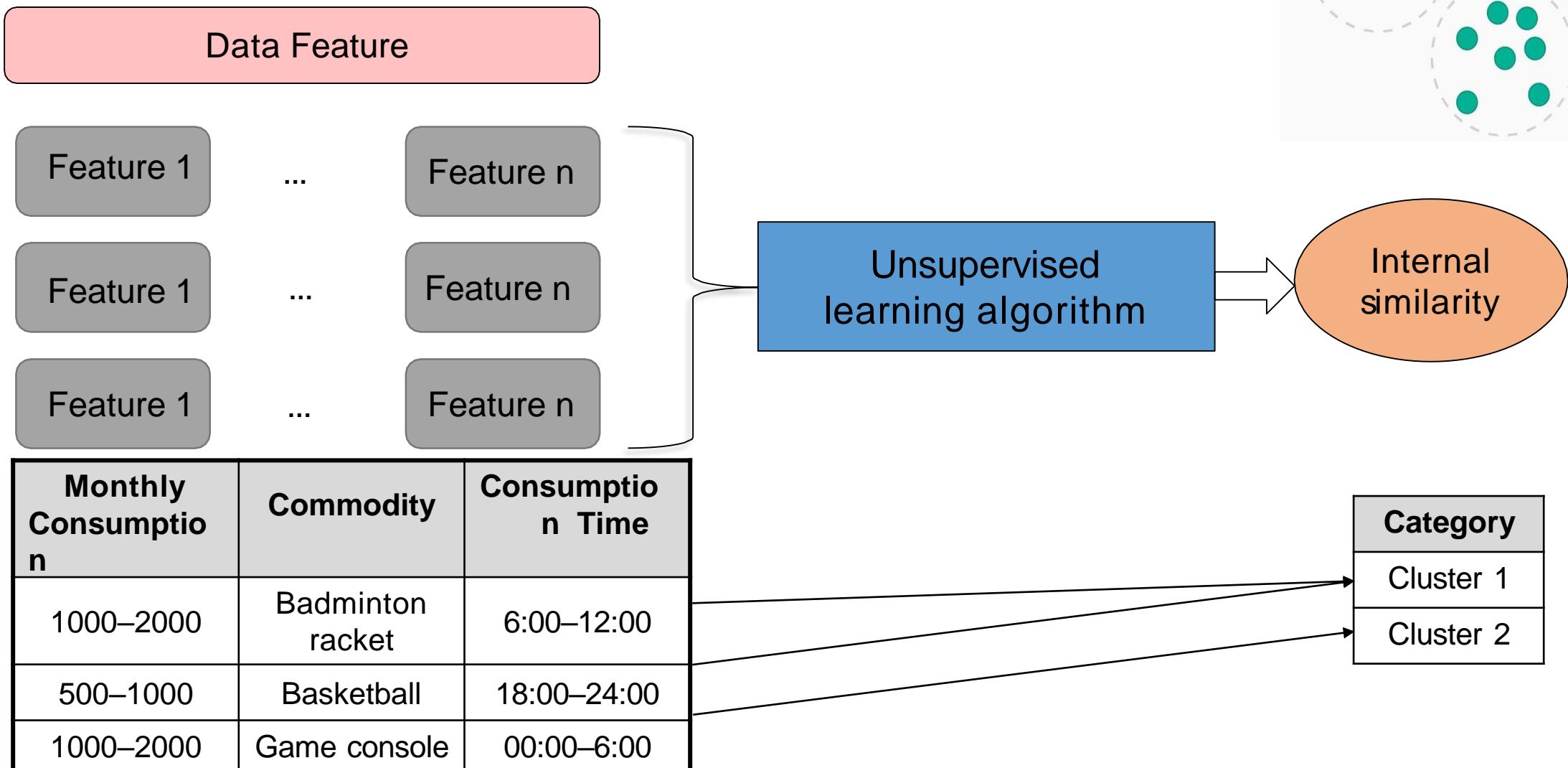


25°

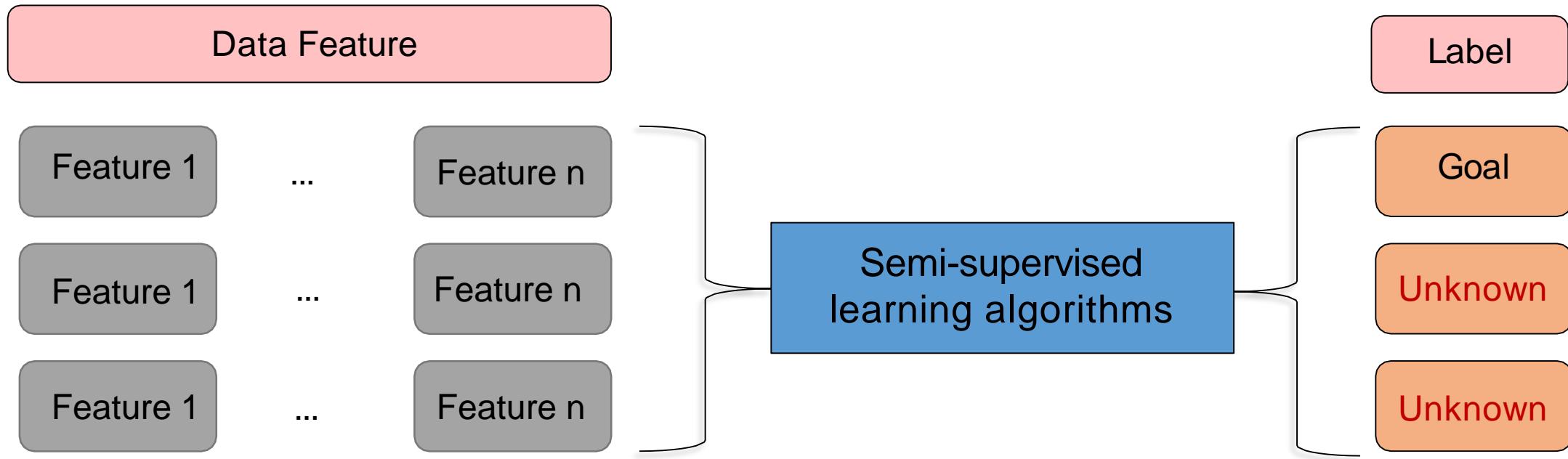
MARDI



Apprentissage non supervisé



Apprentissage semi supervisé

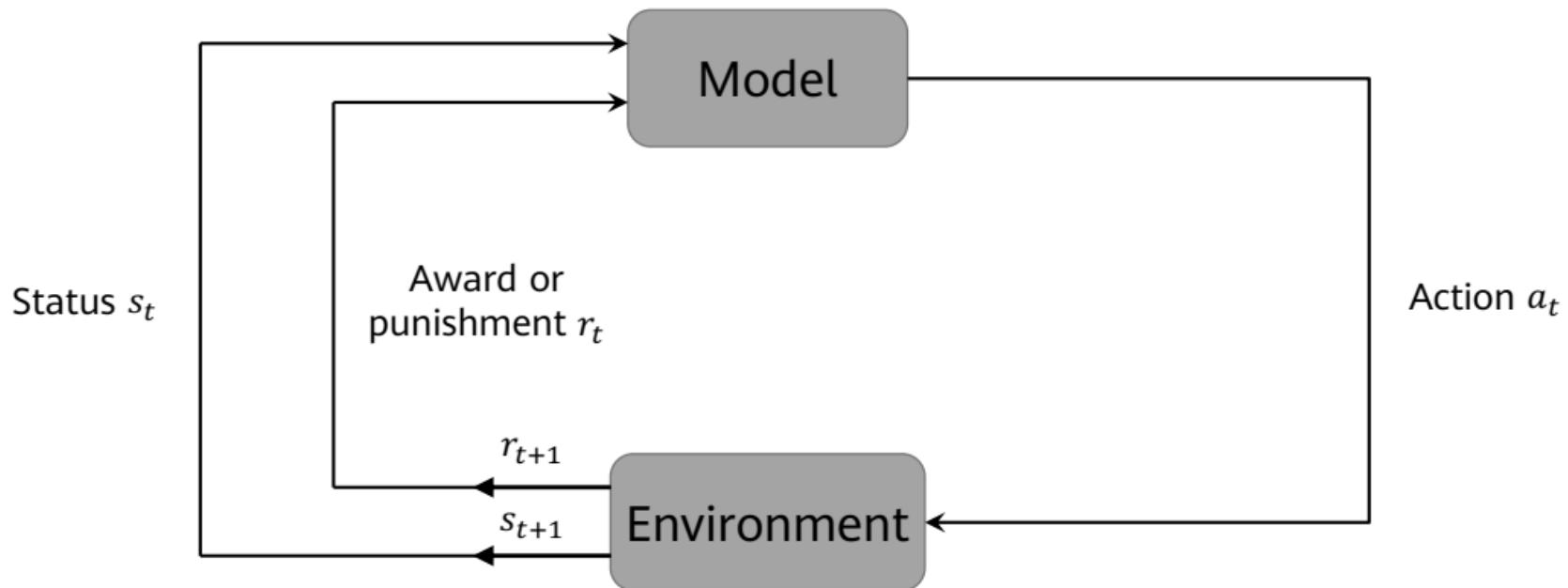


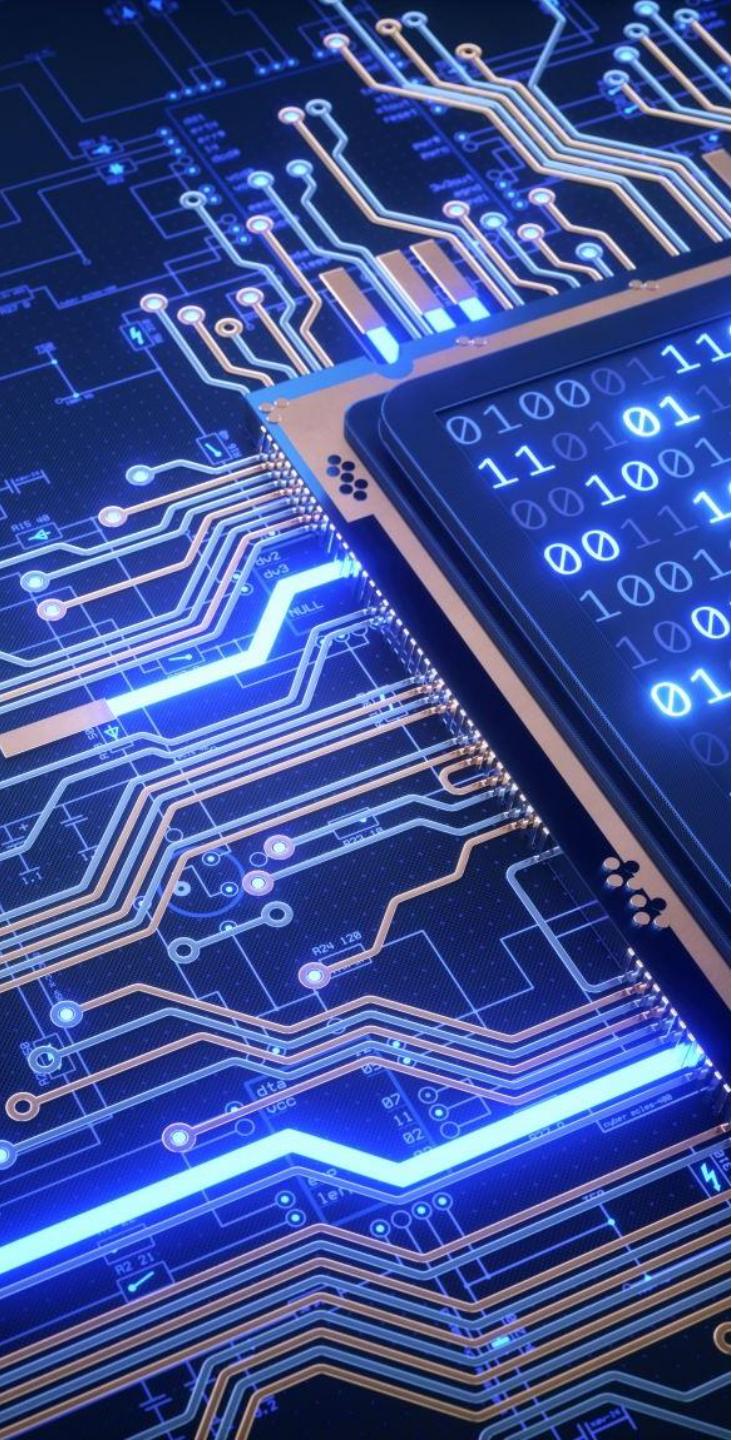
Weather	Temperature	Wind Speed
Sunny	Warm	Strong
Rainy	Cold	Fair
Sunny	Cold	Weak

Enjoy Sports
Yes
/
/

Apprentissage par renforcement

- Le modèle perçoit l'environnement, prend des mesures et fait des ajustements et des choix en fonction du statut et de l'attribution ou de la punition, cherche toujours les meilleurs comportements.
- L'apprentissage par renforcement s'adresse aux machines ou aux robots.

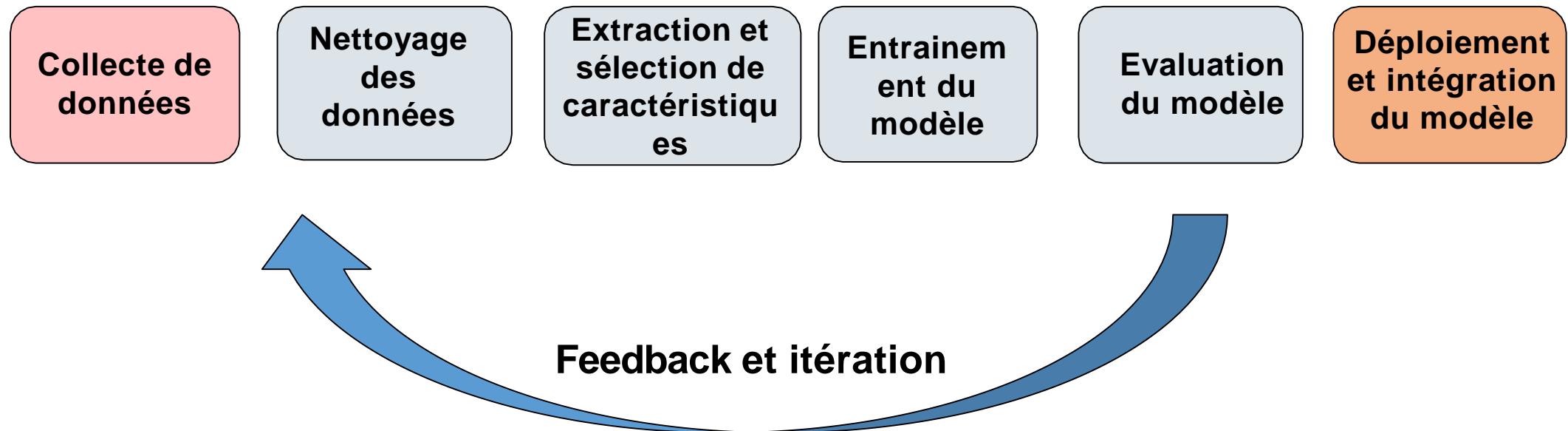




Plan

- Définition de l'apprentissage automatique
- Types d'apprentissage automatique
- **Processus d'apprentissage automatique**
- Autres méthodes clés d'apprentissage automatique
- Algorithmes courants d'apprentissage automatique

Processus ML



Vérification de l'ensemble des données

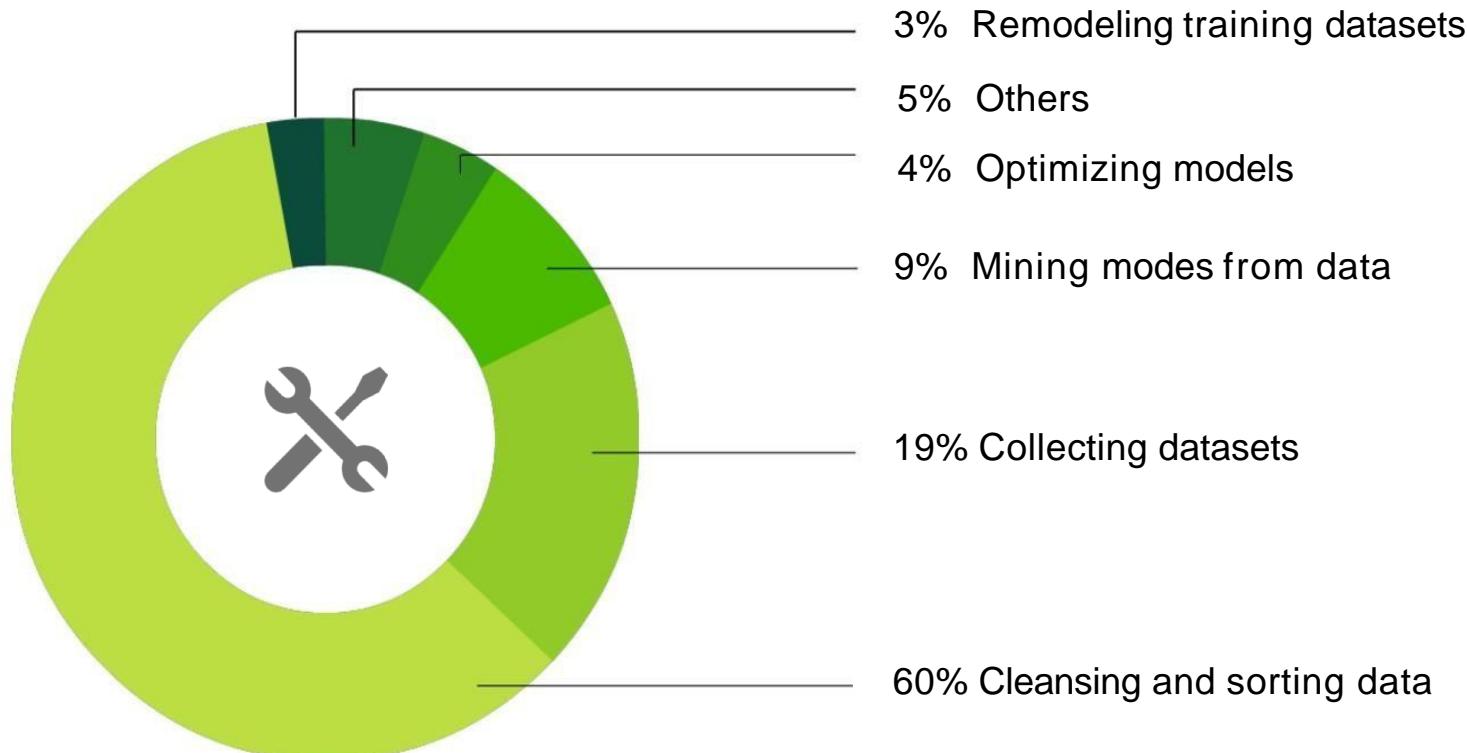
	Feature 1	Feature 2	Feature 3	Label	
No.	Area	School Districts	Direction	House Price	
Training set	1	100	8	South	1000
	2	120	9	Southwest	1300
	3	60	6	North	700
	4	80	9	Southeast	1100
Test set	5	95	3	South	850

Importance du traitement des données

- Les données sont cruciales pour les modèles. C'est le plafond des capacités du modèle. Sans bonnes données, il n'y a pas de bon modèle.



Charge de travail du nettoyage des données



CrowdFlower Data Science Report 2016

Nettoyage des données

- La plupart des modèles d'apprentissage automatique traitent des caractéristiques, qui sont généralement des représentations numériques des variables d'entrée pouvant être utilisées dans le modèle.
- Dans la plupart des cas, les données collectées ne peuvent être utilisées par des algorithmes qu'après avoir été prétraitées. Les opérations de prétraitement comprennent les suivantes :
 - Filtrage des données
 - Traitement des données perdues
 - Traitement des exceptions, erreurs ou valeurs anormales possibles
 - Combinaison de données provenant de plusieurs sources de données
 - Consolidation des données

Mauvaises données (1)

- Généralement, les données réelles peuvent avoir des problèmes de qualité.
 - **Incomplétude** : contient des valeurs manquantes ou des données sans attributs
 - **Bruit** : contient des enregistrements ou des exceptions incorrects.
 - **Incohérence** : contient des enregistrements incohérents.

Mauvaises données (2)

#	Id	Name	Birthday	Gender	IsTeacher	#Students	Country	City
1	111	John	31/12/1990	M	0	0	Ireland	Dublin
2	222	Mery	15/10/1978	F	1	15	Iceland	
3	333	Alice	19/04/2000	F	0	0	Spain	Madrid
4	444	Mark	01/11/1997	M	0	0	France	Paris
5	555	Alex	15/03/2000	A	1	23	Germany	Berlin
6	555	Peter	1983-12-01	M	1	10	Italy	Rome
7	777	Calvin	05/05/1995	M	0	0	Italy	Italy
8	888	Roxane	03/08/1948	F	0	0	Portugal	Lisbon
9	999	Anne	05/09/1992	F	0	5	Switzerland	Geneva
10	101010	Paul	14/11/1992	M	1	26	Ytali	Rome

Annotations pointing to specific errors:

- Invalid duplicate item: Points to row 6 where Id=555.
- Incorrect format: Points to Birthday in row 6 (1983-12-01) and row 8 (03/08/1948).
- Attribute dependency: Points to IsTeacher in row 5 and row 10.
- Missing value: Points to City in row 2.
- Invalid value: Points to Gender in row 5 (A).
- Value that should be in another column: Points to City in rows 7 and 9.
- Misspelling: Points to City in row 10 (Ytali).

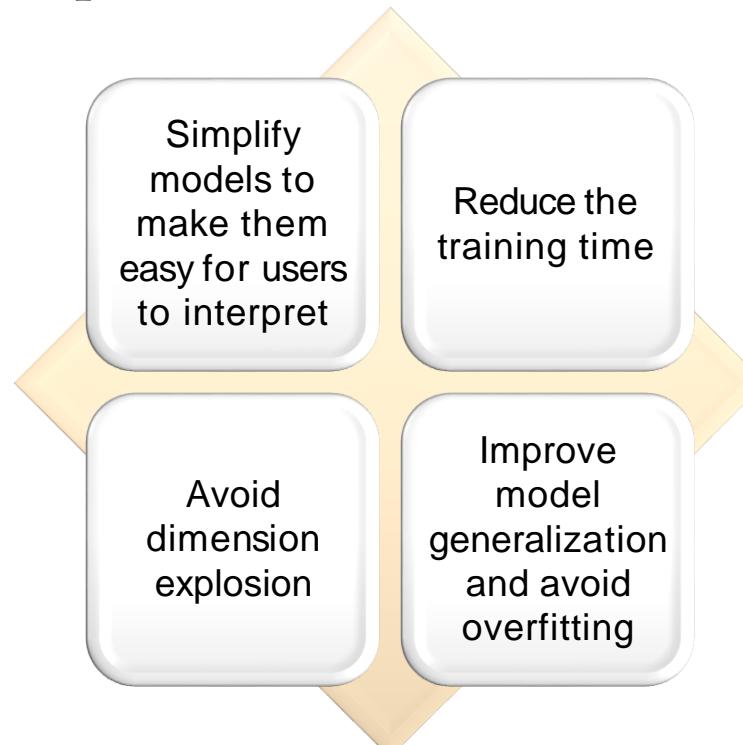
Conversion des données

Après avoir été prétraitées, les données doivent être converties en une forme de représentation adaptée au modèle d'apprentissage. Les formes de conversion de données courants sont les suivants :

- En ce qui concerne la classification, les données de catégorie sont codées dans une représentation numérique correspondante.
- Les données de valeur sont converties en données de catégorie pour réduire la valeur des variables (pour la segmentation par âge).
- Autres données
 - Dans le texte, le mot est converti en un vecteur de mot par insertion de mots (généralement en utilisant le modèle word2vec, modèle BERT, etc.).
 - Traiter les données d'image (espace colorimétrique, niveaux de gris, changement géométrique et amélioration de l'image)
 - Ingénierie des fonctionnalités
 - Normaliser les caractéristiques pour garantir les mêmes plages de valeurs pour les variables d'entrée du même modèle.
 - Extension des fonctionnalités : combinez ou convertissez des variables existantes pour générer de nouvelles fonctionnalités, telles que la moyenne.

Nécessité de la sélection des fonctionnalités

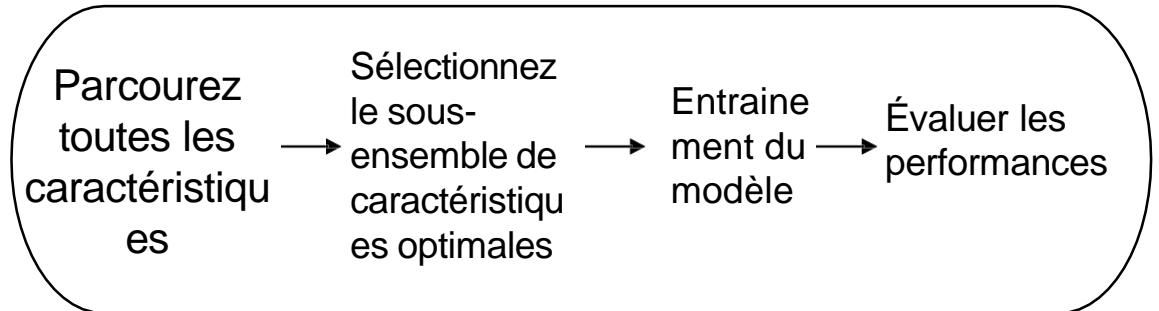
- Généralement, un ensemble de données a de nombreuses caractéristiques, dont certaines peuvent être redondantes ou sans rapport avec la valeur à prédire.
- La sélection des caractéristiques est nécessaire dans les aspects suivants :



Méthodes de sélection des caractéristiques -

Filtre

- Les méthodes de filtrage sont indépendantes du modèle lors de la sélection des caractéristiques.



Procédure d'une méthode de filtrage

En évaluant la corrélation entre chaque caractéristique et l'attribut cible, ces méthodes utilisent une mesure statistique pour attribuer une valeur à chaque caractéristique. Les fonctionnalités sont ensuite triées par score, ce qui est utile pour préserver ou éliminer des fonctionnalités spécifiques.

Méthodes courantes :

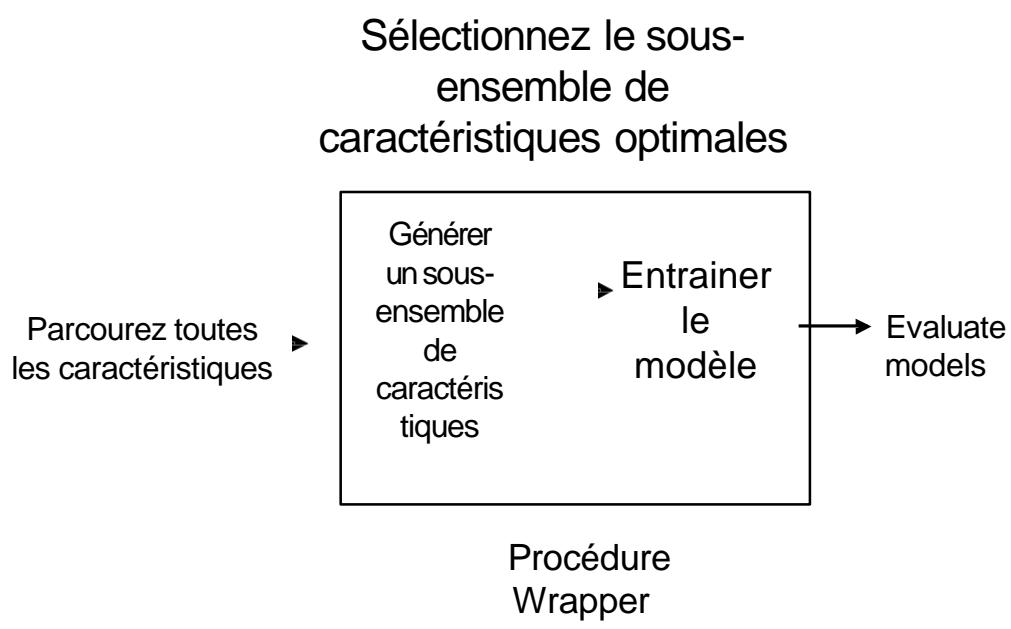
- Coefficient de corrélation de Pearson
- Coefficient Chi-square
- Informations mutuelles

Limites :

- La méthode de filtrage a tendance à sélectionner des variables redondantes car la relation entre les caractéristiques n'est pas prise en compte.

Méthodes de sélection de caractéristiques - Wrapper

- Les méthodes wrapper utilisent un modèle de prédiction pour évaluer les sous-ensembles de caractéristiques.



Les méthodes wrapper considèrent la sélection de caractéristiques comme un problème de recherche pour lequel différentes combinaisons sont évaluées et comparées. Un modèle prédictif est utilisé pour évaluer une combinaison de caractéristiques et attribuer un score basé sur la précision du modèle.

Méthodes communes

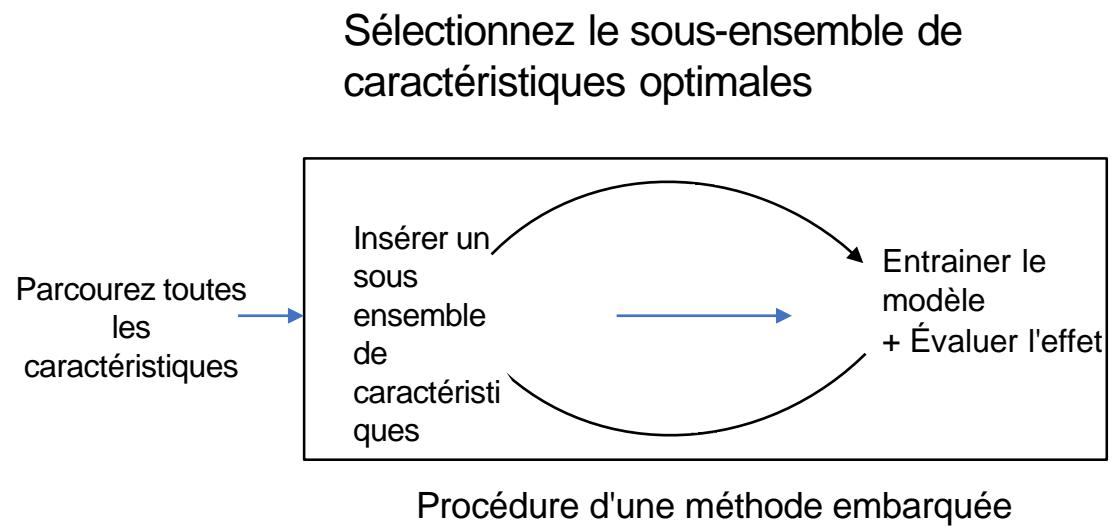
- Recursive feature elimination (RFE)

Limites

- Les méthodes wrapper entraînent un nouveau modèle pour chaque sous-ensemble, ce qui entraîne **un grand nombre de calculs**.
- Un ensemble de caractéristiques avec les meilleures performances est généralement fourni pour un type spécifique de modèle

Méthodes de sélection de fonctionnalités – Intégrées (Embedded)

- Les méthodes intégrées considèrent la sélection des caractéristiques comme faisant partie de la construction du modèle.

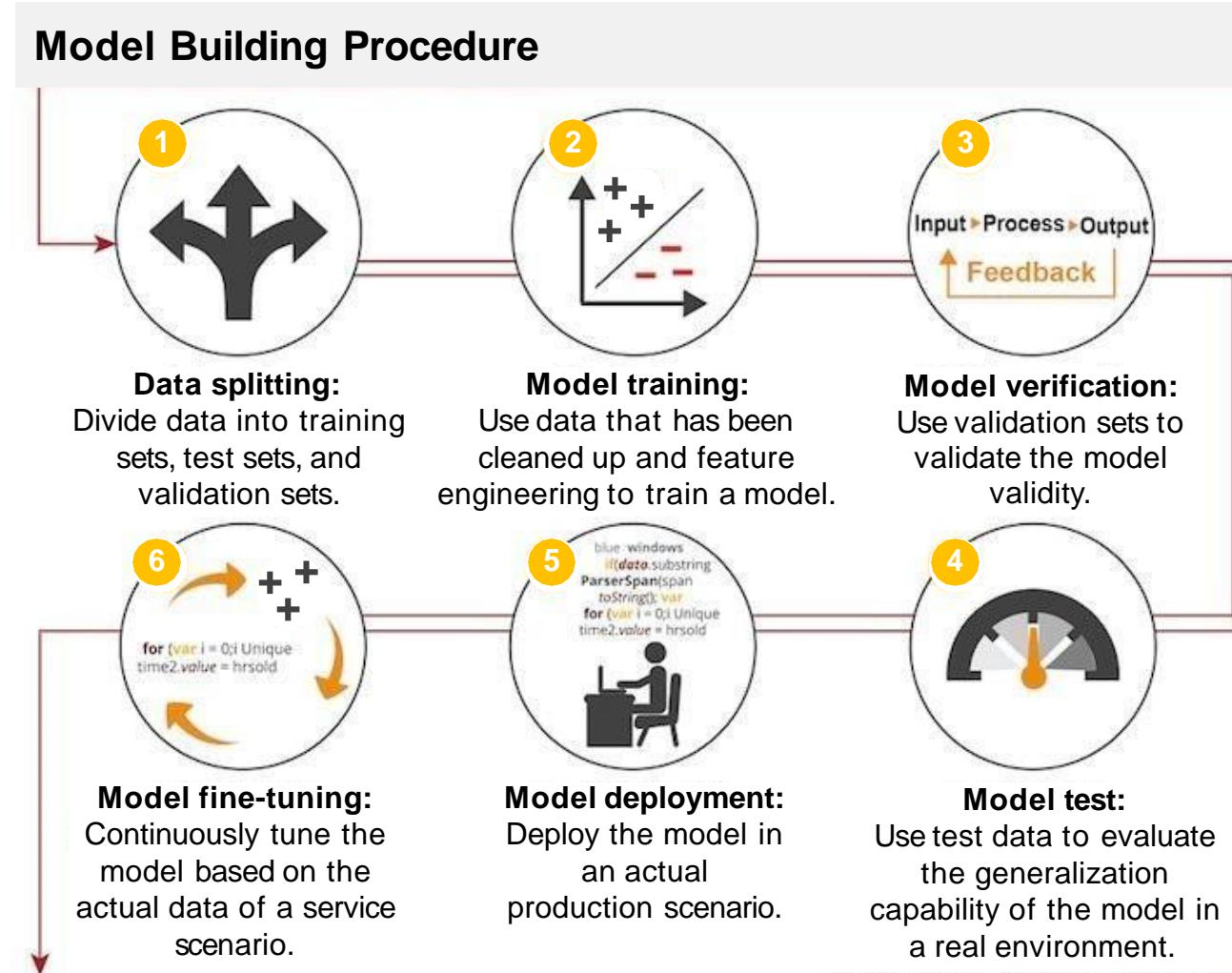


Le type le plus courant de méthode de sélection de caractéristiques intégrées est la **méthode de régularisation**. Les méthodes de régularisation sont également appelées méthodes de pénalisation qui introduisent des contraintes supplémentaires dans l'optimisation d'un algorithme prédictif qui biaissent le modèle vers une complexité moindre et réduisent le nombre de caractéristiques.

Méthodes communes

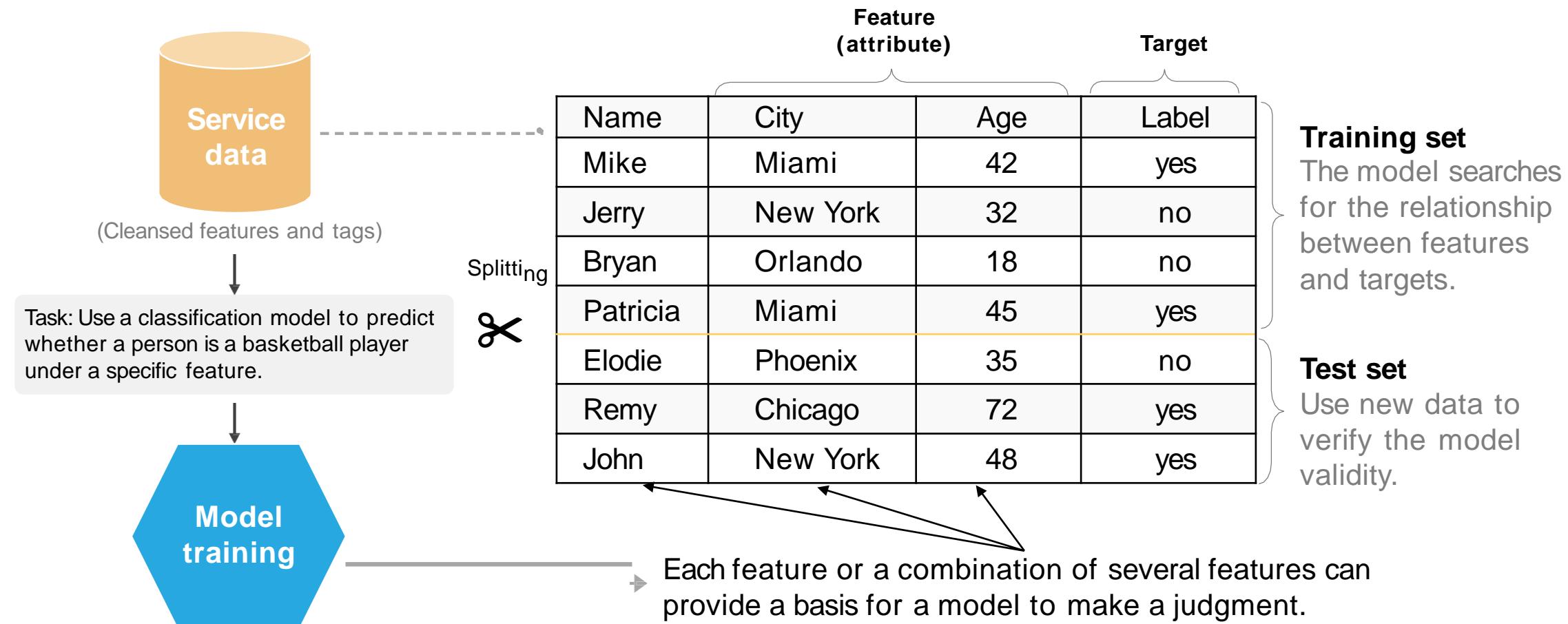
- Lasso regression
- Ridge regression

Procédure globale de construction d'un modèle

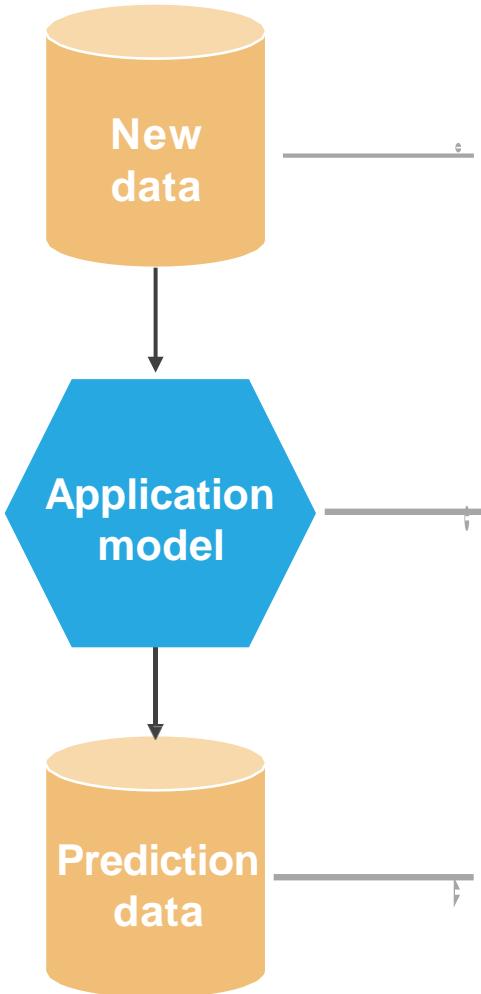


Exemples d'apprentissage supervisé - Phase d'apprentissage

- Utilisez le modèle de classification pour prédire si une personne est un joueur de basket-ball.



Exemples d'apprentissage supervisé - Phase de prédiction



Name	City	Age	Label
Marine	Miami	45	?
Julien	Miami	52	?
Fred	Orlando	20	?
Michelle	Boston	34	?
Nicolas	Phoenix	90	?

Unknown data

Recent data, it is not known whether the people are basketball players.

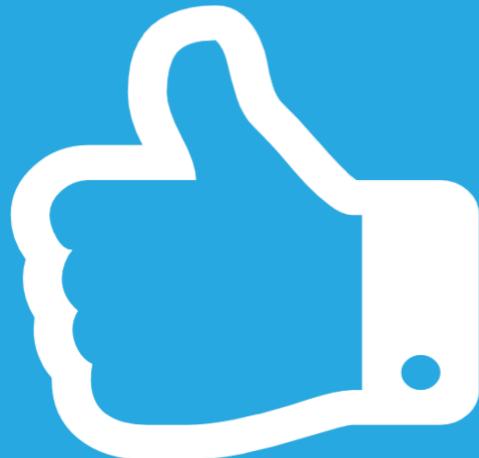
IF city = Miami → Probability = +0.7
IF city= Orlando → Probability = +0.2
IF age > 42 → Probability = +0.05*age + 0.06
IF age ≤ 42 → Probability = +0.01*age + 0.02

Name	City	Age	Prediction
Marine	Miami	45	0.3
Julien	Miami	52	0.9
Fred	Orlando	20	0.6
Michelle	Boston	34	0.5
Nicolas	Phoenix	90	0.4

Possibility prediction

Apply the model to the new data to predict whether the customer will change the supplier.

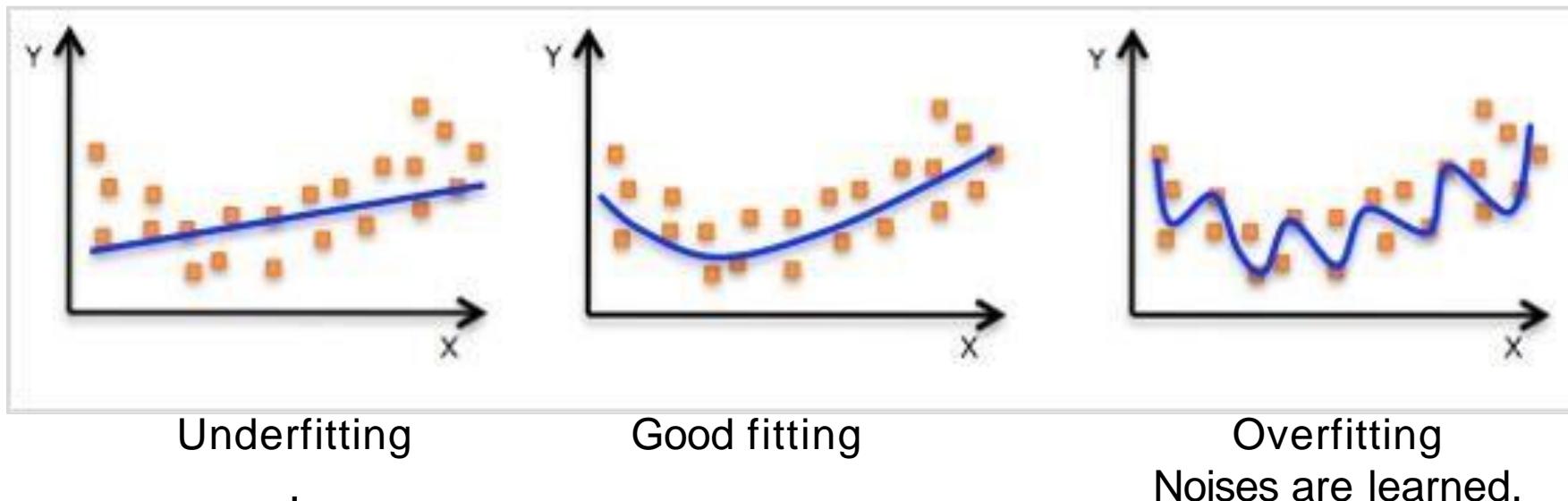
Qu'est-ce qu'un bon modèle ?



- **Capacité de généralisation**
Peut-il prédire avec précision les données de réelles ?
- **Interprétabilité**
Le résultat de la prédiction est-il facile à interpréter ?
- **Vitesse de prédiction**
Combien de temps faut-il pour prédire chaque donnée ?
- **Praticabilité**
Le taux de prédiction est-il toujours acceptable lorsque le volume de données devient énorme ?

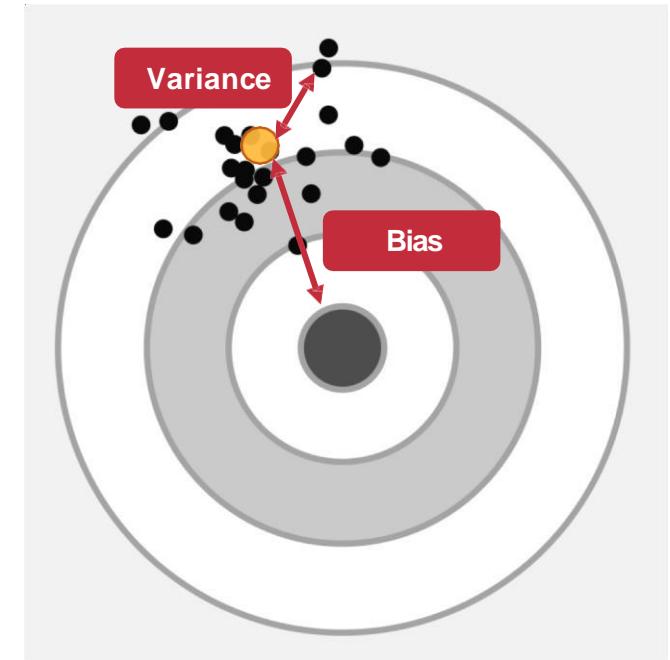
Validité du modèle

- Capacité du modèle : capacité du modèle à ajuster les fonctions, également appelée complexité du modèle.
 - Lorsque la capacité correspond à la complexité de la tâche et à la quantité de données d'entraînement fournies, l'effet de l'algorithme est généralement optimal.
 - Les modèles avec une capacité insuffisante ne peuvent pas résoudre des tâches complexes et un sous-apprentissage peut se produire.
 - Un modèle à haute capacité peut résoudre des tâches complexes, mais un surapprentissage peut se produire si la capacité est supérieure à celle requise par une tâche.



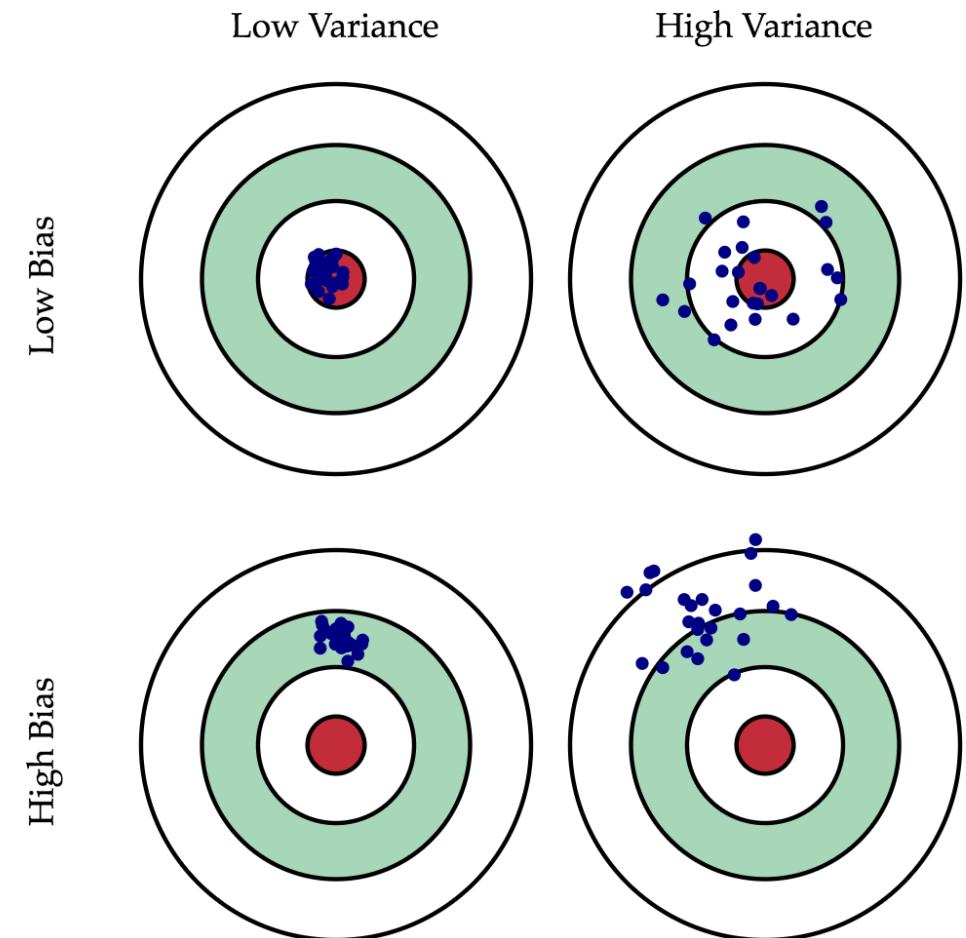
Cause du sur-apprentissage – Erreur

- **Erreur totale de la prédiction finale = Bias + Variance + Erreur irréductible**
- Généralement, l'erreur de prédiction peut être divisée en deux types :
 - Erreur causée par "biais"
 - Erreur causée par l'écart « Variance »
- Variance:
 - Décalage du résultat de prédiction par rapport à la valeur moyenne
 - Erreur causée par la sensibilité du modèle aux petites fluctuations de l'ensemble d'apprentissage
- Bias:
 - Différence entre la valeur de prédiction attendue (ou moyenne) et la valeur correcte que nous essayons de prédire.



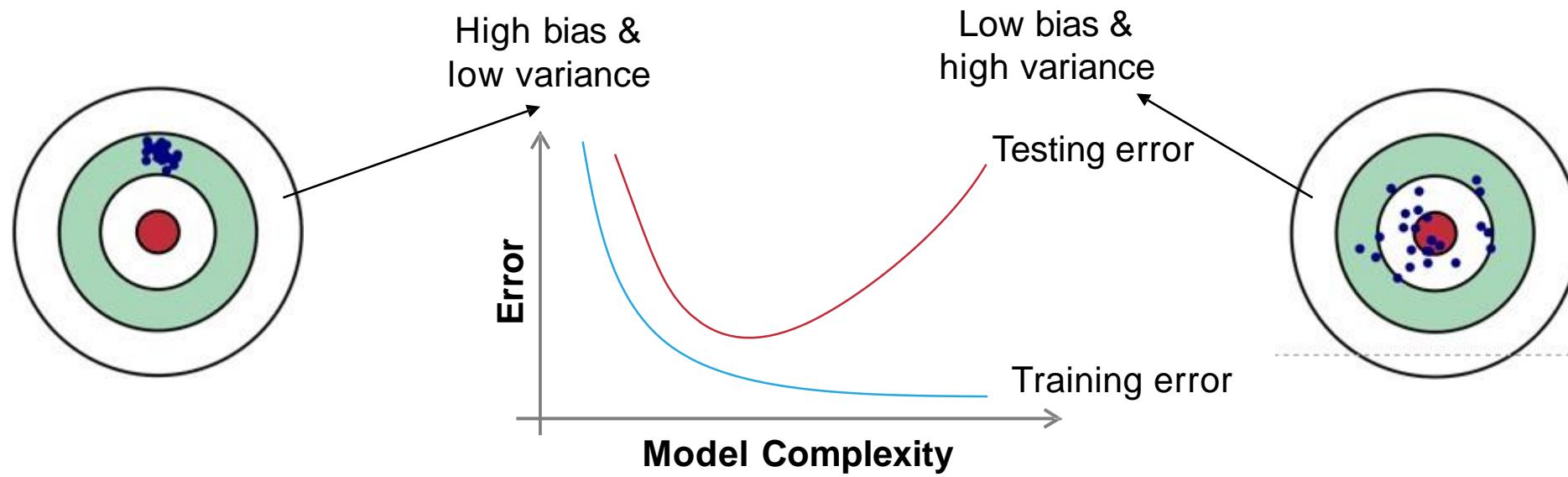
Variance et Bias

- Les combinaisons de variance et de biais sont les suivantes :
 - Faible biais et faible variance -> Bon modèle
 - Faible biais et variance élevée
 - Biais élevé et faible variance
 - Biais élevé et variance élevée -> Modèle médiocre
- Idéalement, nous voulons un modèle capable de capturer avec précision les règles dans les données d'apprentissage et de résumer les données invisibles (nouvelles données). Cependant, il est généralement impossible pour le modèle d'effectuer les deux tâches en même temps.



Complexité du modèle et erreur

- Au fur et à mesure que la complexité du modèle augmente, l'erreur d'apprentissage diminue.
- À mesure que la complexité du modèle augmente, l'erreur de test diminue jusqu'à un certain point puis augmente dans le sens inverse, formant une courbe convexe.



Évaluation des performances du ML - Régression

- Plus l'erreur absolue moyenne (Mean Absolute Error, MAE) est proche de 0, mieux le modèle peut s'adapter aux données d'apprentissage.

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$$

- Erreur quadratique moyenne (Mean Square Error, MSE)


$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

- La plage de valeurs de R^2 (range) est $(-\infty, 1]$. Une valeur plus élevée indique que le modèle peut mieux s'adapter aux données d'apprentissage. TSS indique la différence entre les échantillons. RSS indique la différence entre la valeur prédite et la valeur de l'échantillon.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{\sum_{i=1}^m (y_i - \bar{y}_i)^2}$$

Évaluation des performances du ML - Classification (1)

- P : positif, indiquant le nombre de cas réels positifs dans les données.
- N : négatif, indiquant le nombre de cas réels négatifs dans les données.
- TP : vrai positif, indiquant le nombre de cas positifs correctement classés par le classifieur.
- TN : vrai négatif, indiquant le nombre de cas négatifs qui sont correctement classés par le classifieur.
- FP : faux positif, indiquant le nombre de cas positifs mal classés par le classificateur.
- FN : faux négatif, indiquant le nombre de cas négatifs mal classés par le classificateur.

Évaluation des performances du ML - Classification (2)

- Matrice de confusion : au moins une table $m \times m$. $CM_{i,j}$ des m premières lignes et m colonnes indiquent le nombre de cas qui appartiennent réellement à la classe i mais qui sont classés dans la classe j par le classificateur.
 - Idéalement, pour un classificateur de haute précision, la plupart des valeurs de prédiction devraient être situées dans la diagonale de $CM_{1,1}$ à $CM_{m,m}$ du tableau tandis que les valeurs en dehors de la diagonale sont 0 ou proches de 0.

Estimated amount		yes	no	Total
Actual amount	yes	TP	FN	P
no	no	FP	TN	N
Total		P'	N'	$P + N$

Confusion matrix

Évaluation des performances du ML - Classification (3)

Measurement	Ratio
Accuracy and recognition rate	$\frac{TP + TN}{P + N}$
Error rate and misclassification rate	$\frac{FP + FN}{P + N}$
Sensitivity, true positive rate, and recall	$\frac{TP}{P}$
Specificity and true negative rate	$\frac{TN}{N}$
Precision	$\frac{TP}{TP + FP}$
F_1 , harmonic mean of the recall rate and precision	$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
F_β , where β is a non-negative real number	$\frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$

Exemple d'évaluation des performances

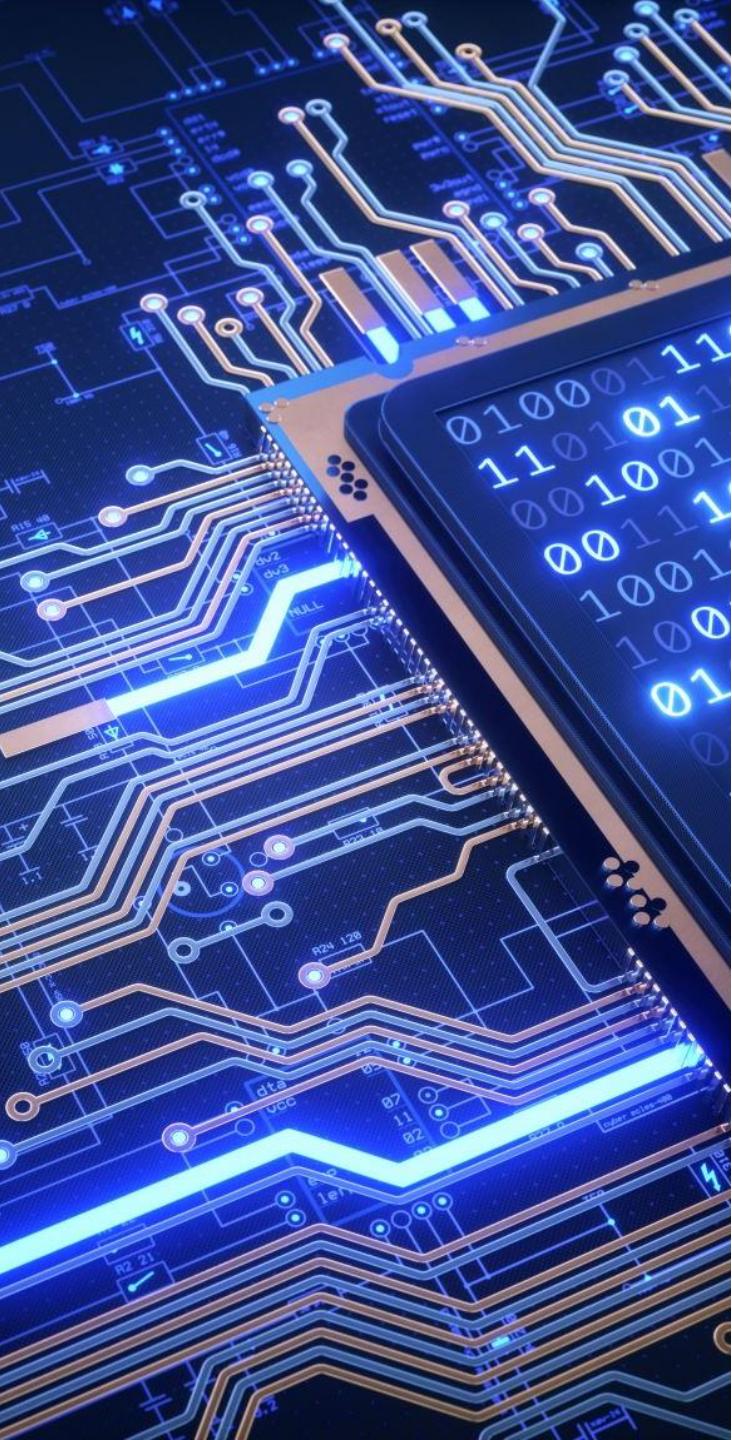
Nous avons entraîné un modèle d'apprentissage automatique pour identifier si l'objet dans une image est un chat. Maintenant, nous utilisons 200 images pour vérifier les performances du modèle. Parmi les 200 images, les objets sur 170 images sont des chats, tandis que d'autres ne le sont pas. Le résultat d'identification du modèle est que les objets dans 160 images sont des chats, tandis que d'autres ne le sont pas.

$$\text{Precision: } P = \frac{TP}{TP+FP} = \frac{140}{140+20} = 87.5\%$$

$$\text{Recall: } R = \frac{TP}{P} = \frac{140}{170} = 82.4\%$$

$$\text{Accuracy: } ACC = \frac{TP+TN}{P+N} = \frac{140+10}{170+30} = 75\%$$

Actual amount	Estimated amount		Total
	yes	no	
yes	140	30	170
no	20	10	30
Total	160	40	200



Plan

- Définition de l'apprentissage automatique
- Types d'apprentissage automatique
- Processus d'apprentissage automatique
- **Autres méthodes clés d'apprentissage automatique**
- Algorithmes courants d'apprentissage automatique

Exemple Régression linéaire à une variable

$$y = ax + b$$

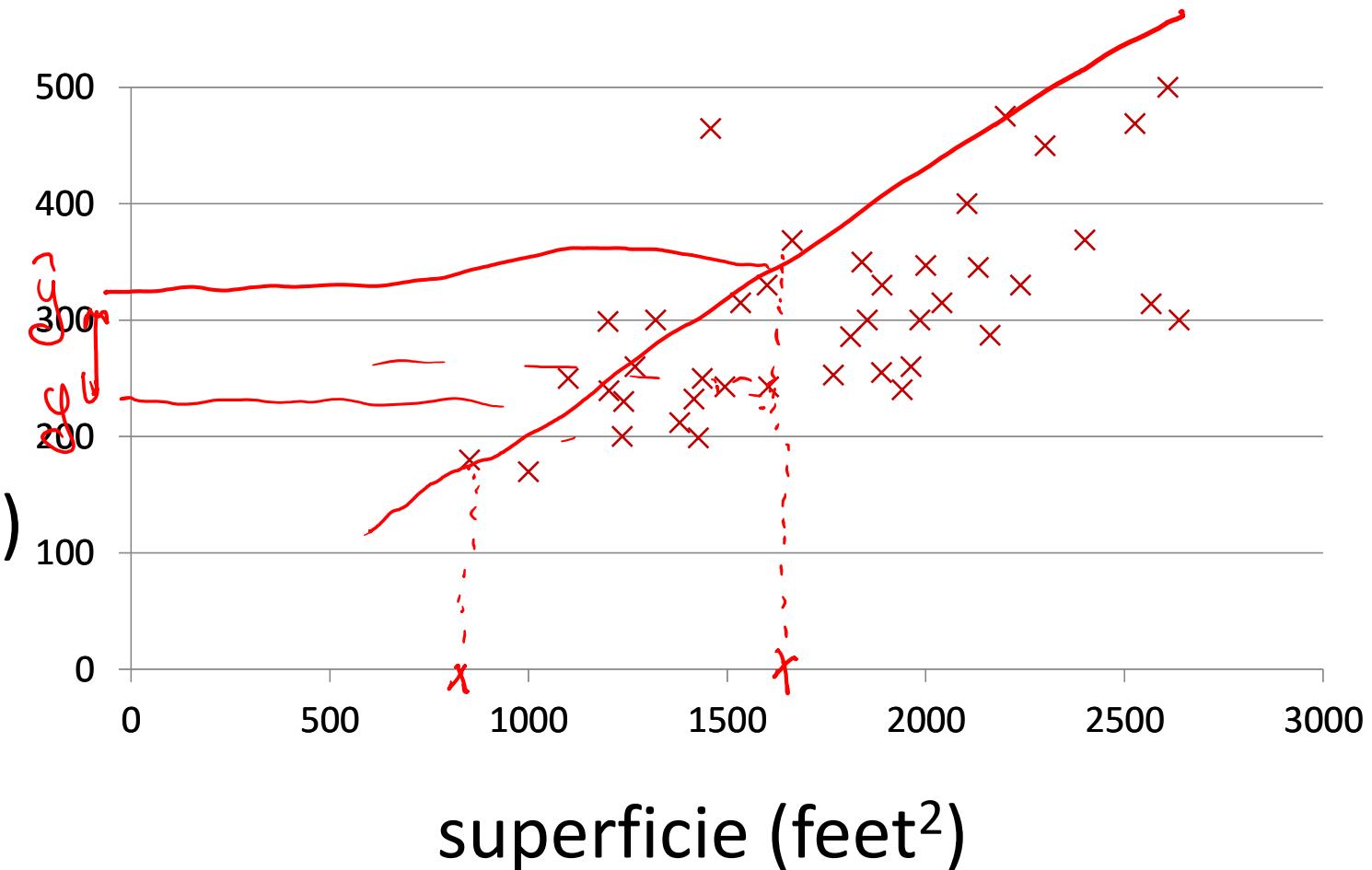
Apprentissage supervisé

On donne la « bonne réponse » pour chaque exemple dans les données.

prix
(en 1000\$)

Problème de régression

Prédire la valeur réelle



Exemple Régression linéaire à une variable

Superficie en feet ² (x)	Prix en 1000\$ (y)
2104	460
1416	232
1534	315
852	178
...	...

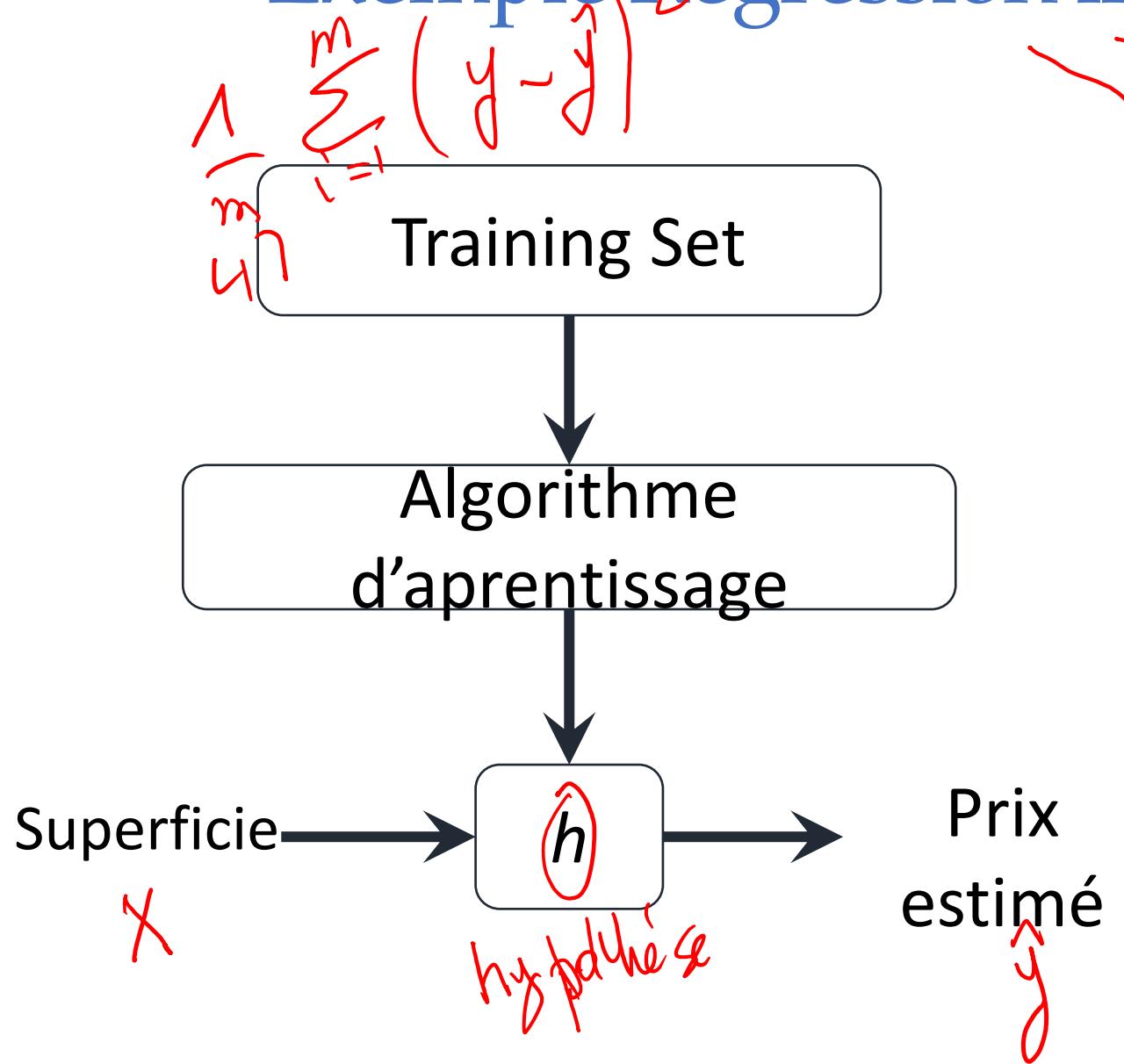
Notation:

m = Nombre d'échantillons

x = variable "d'entrée" / caractéristiques

y = variable « sortie » / variable « target »

Exemple Régression linéaire à une variable



Comment représente-t-on h ?

$$\hat{y} = h(x) = \theta_0 + \theta_1 x$$

La fonction de coût

~~max: f(y)~~

Superficie en feet ² (x)	Prix en 1000\$ (y)
2104	460
1416	232
1534	315
852	178

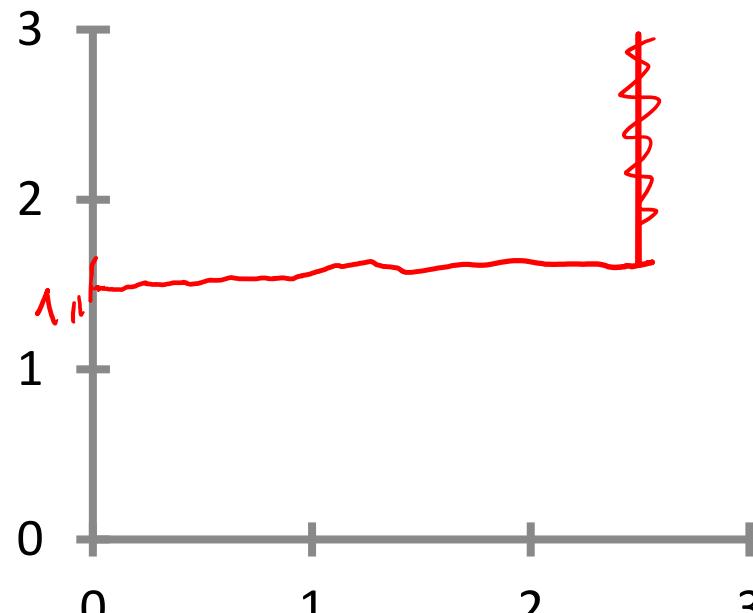
Hypothèse: $h_{\theta}(x) = \theta_0 + \theta_1 x$

Choisir θ_i ?

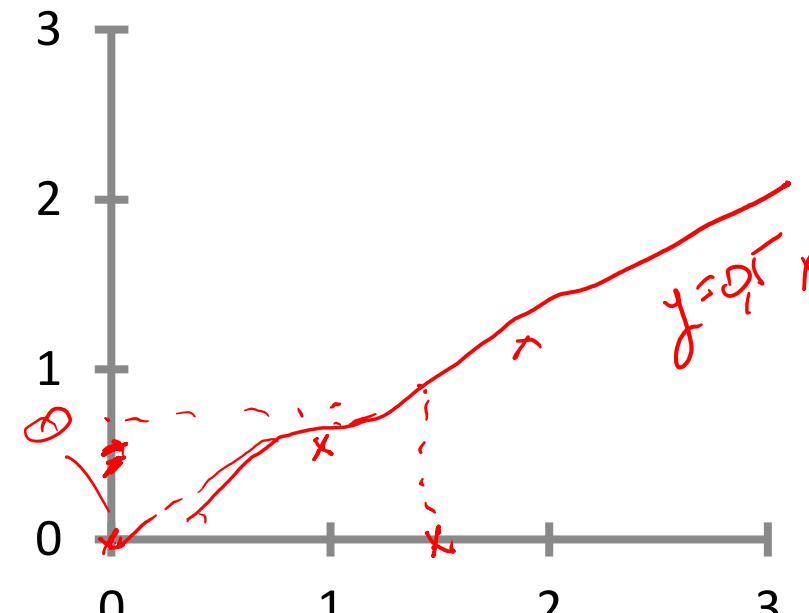
paramètres

La fonction de coût

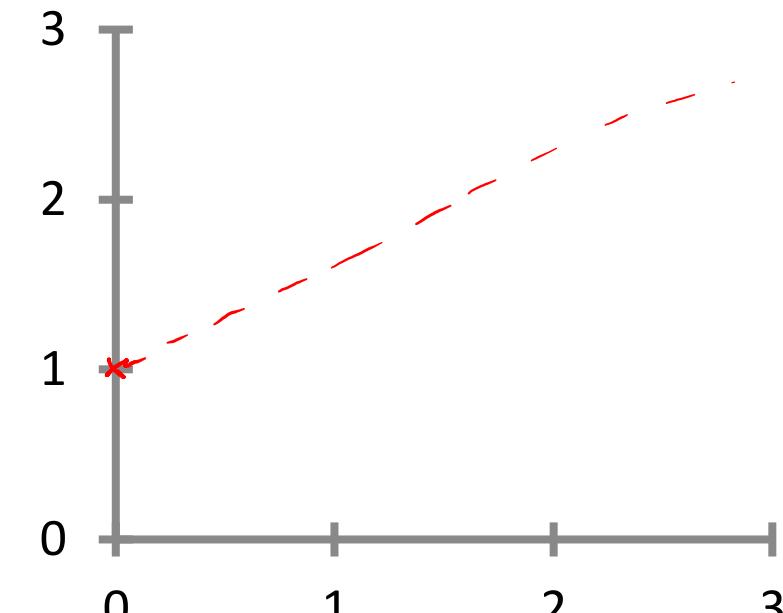
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



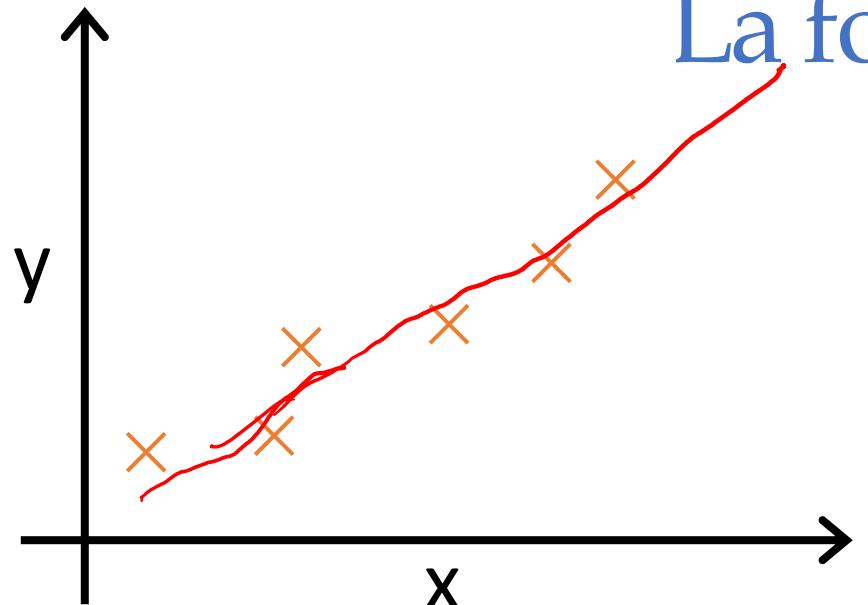
$$\begin{aligned}\theta_0 &= 1.5 \\ \theta_1 &= 0\end{aligned}$$



$$\begin{aligned}\theta_0 &= 0 \\ \theta_1 &= 0.5\end{aligned}$$



$$\begin{aligned}\theta_0 &= 1 \\ \theta_1 &= 0.5\end{aligned}$$



La fonction de coût

$\underset{\theta_0, \theta_1}{\text{min}}$

$$\sum_{i=1}^m \frac{1}{m} (y_i - h_\theta(x_i))^2$$

$J(\theta_0, \theta_1)$
cost function

Idée: Choisir θ_0, θ_1 tels que
 $h_\theta(x)$ est proche de y
pour un exemple (x, y)
d'entraînement

La fonction de coût

Hypothèse:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Paramètres:

$$\theta_0, \theta_1$$

Fonction de coût

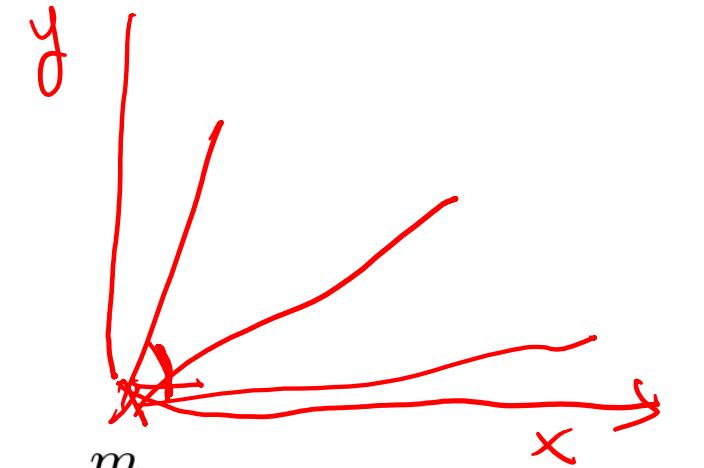
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Objectif: $\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$

Simplifié

$$h_{\theta}(x) = \theta_1 x$$

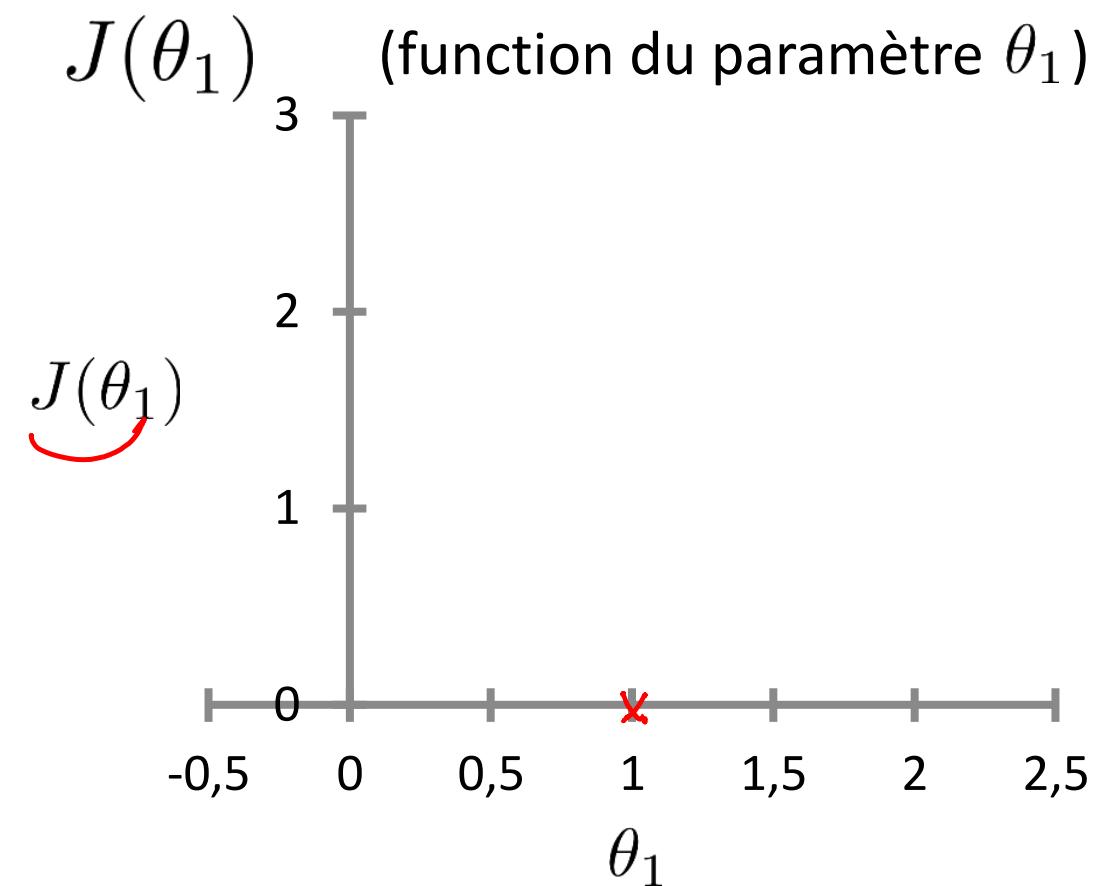
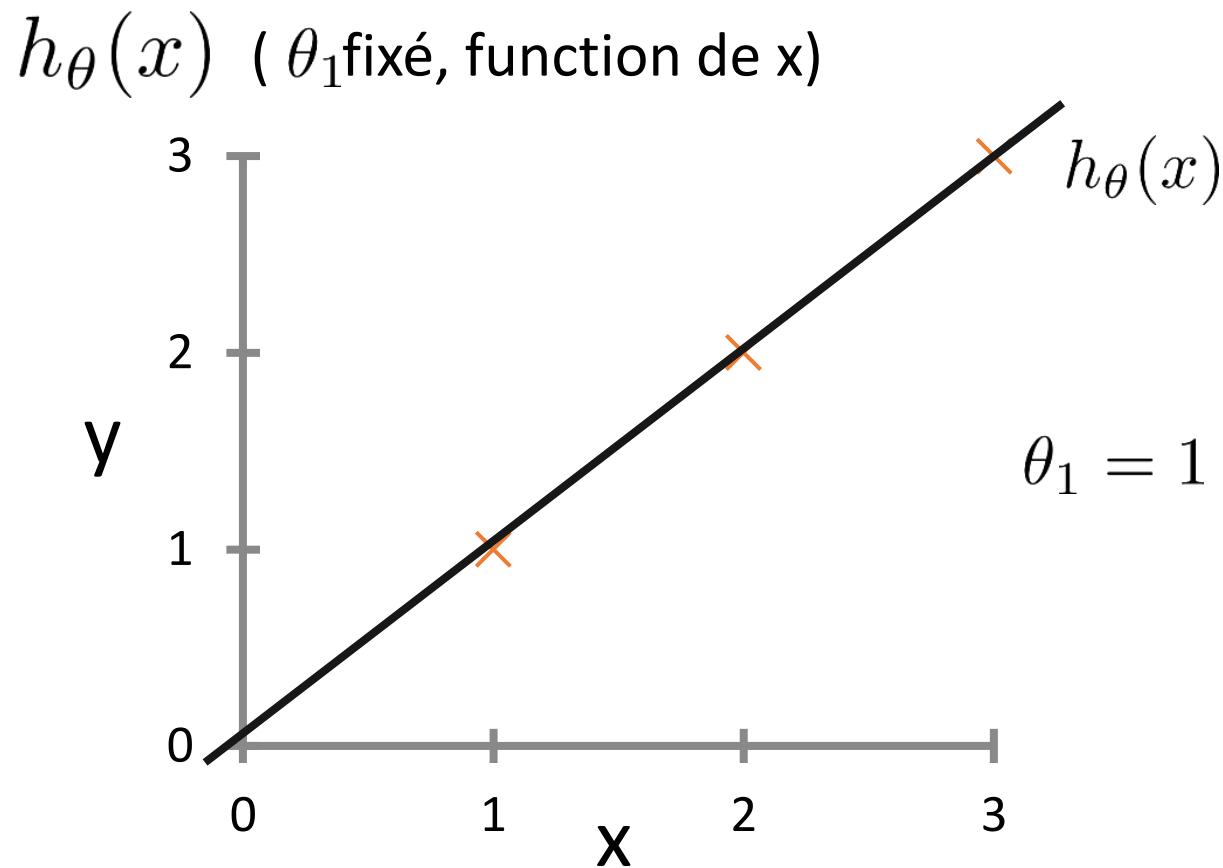
$$\theta_1$$



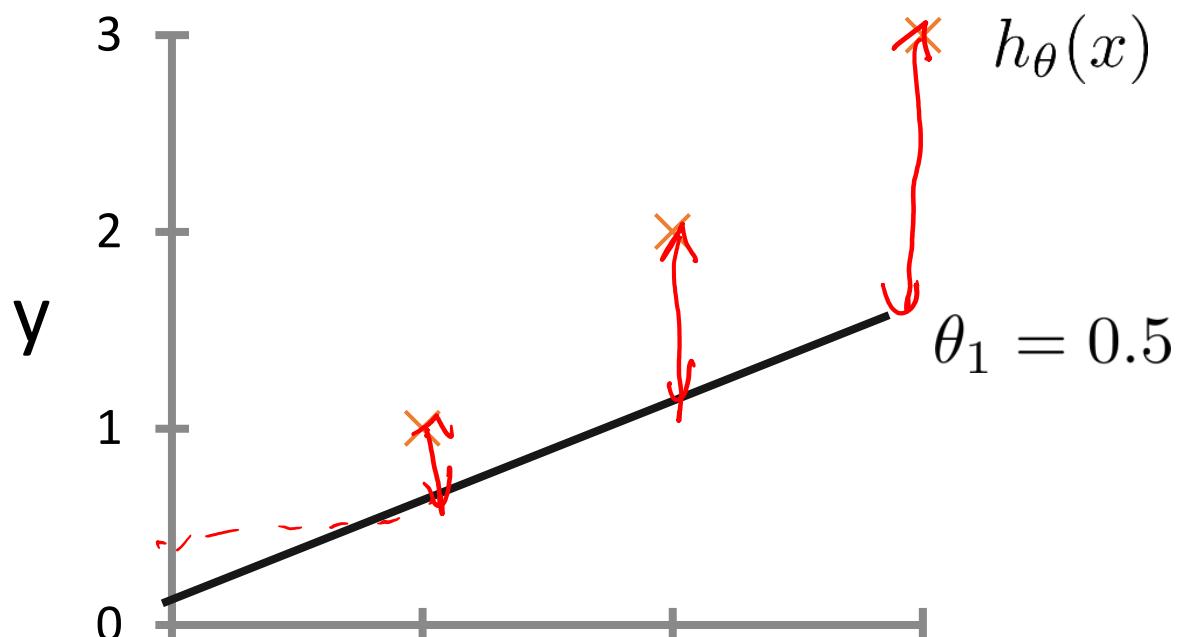
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$\underset{\theta_1}{\text{minimize}} J(\theta_1)$

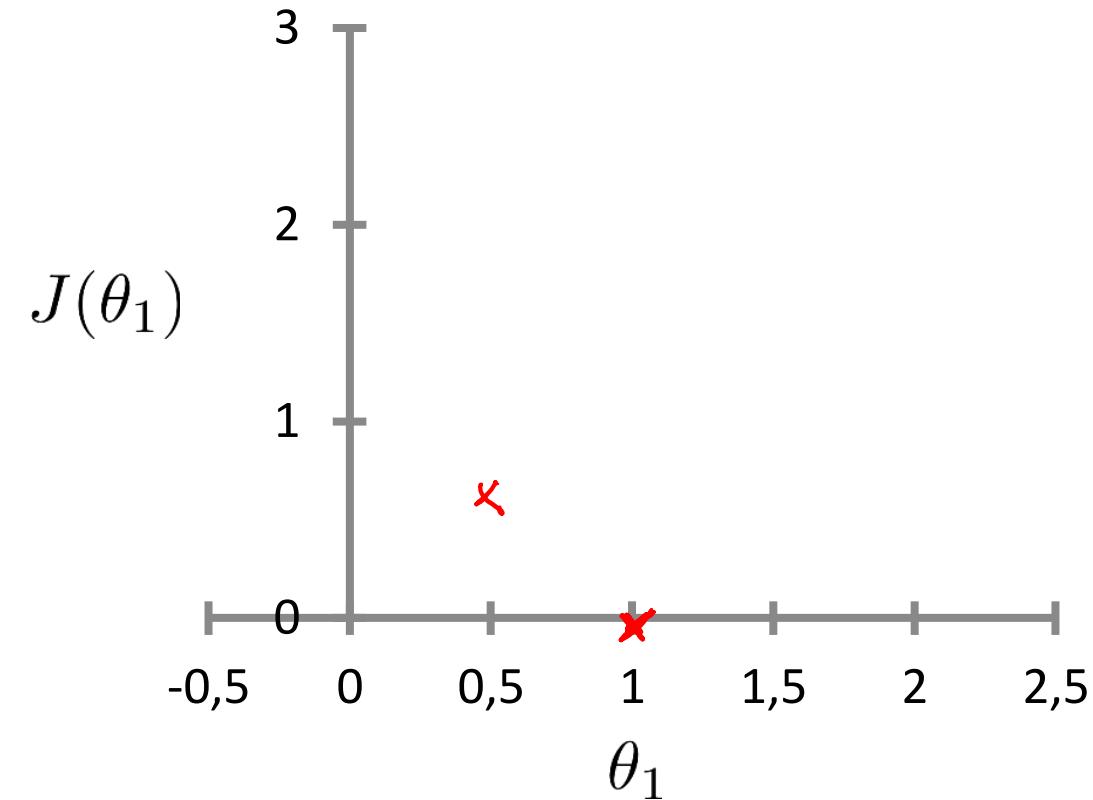
La fonction de coût



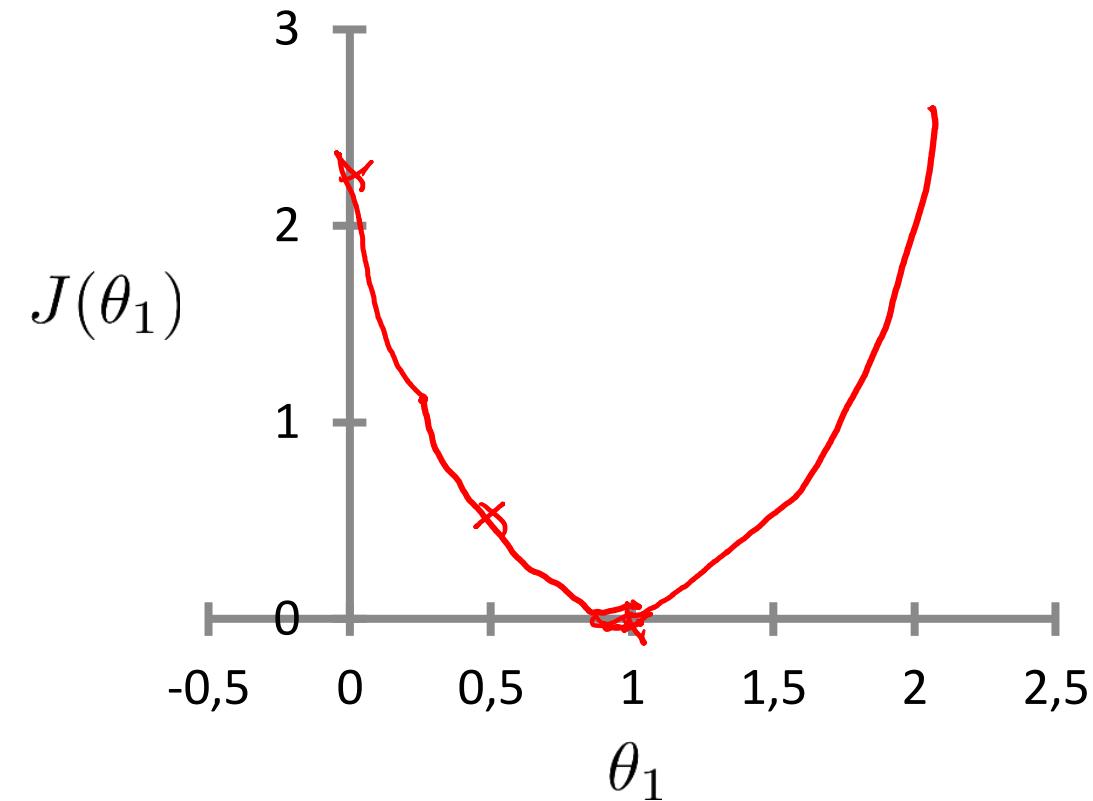
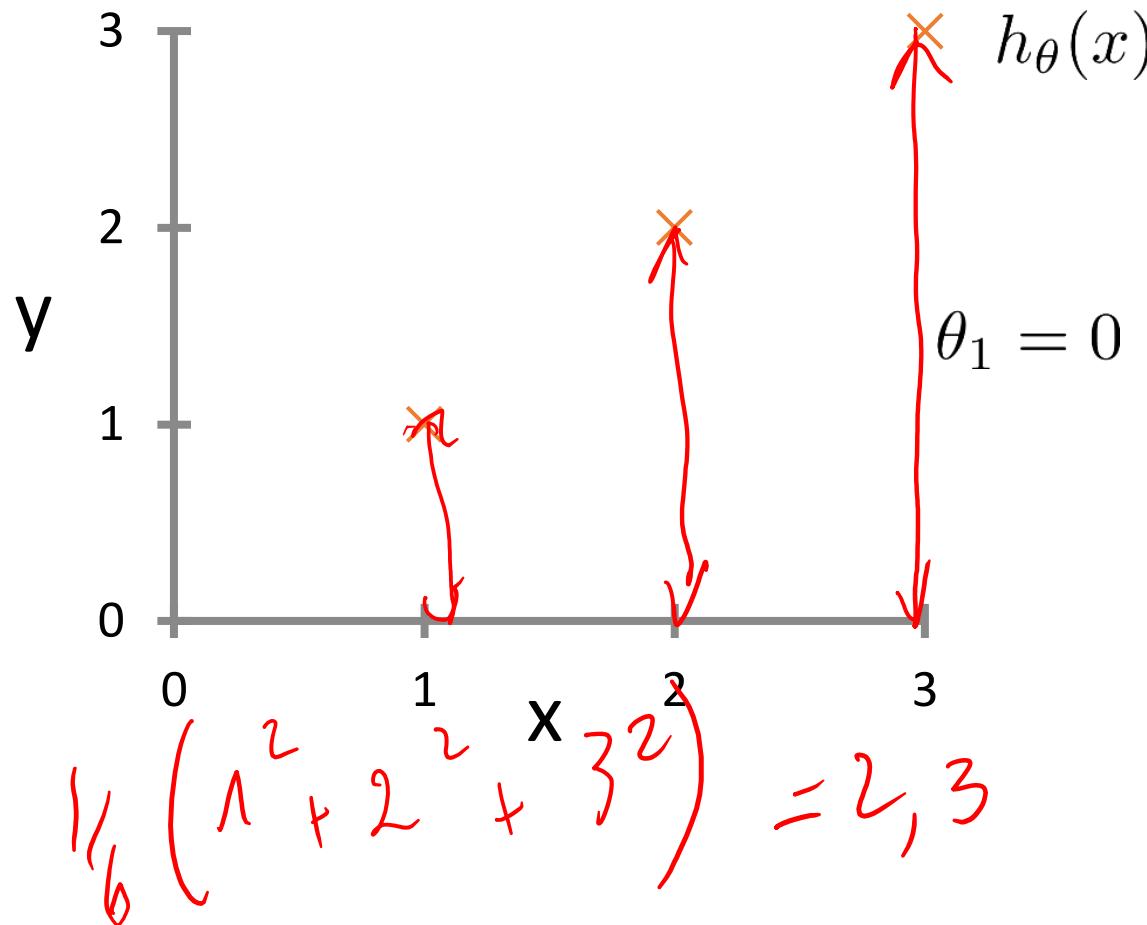
La fonction de coût



$$\frac{1}{6} \sum_{i=1}^3 \left[(0.5 \cdot 1 - 1)^2 + (0.5 \cdot 2 - 2)^2 + (0.5 \cdot 3 - 3)^2 \right] = 0.158$$



La fonction de coût



La fonction de coût

Hypothèse: $h_{\theta}(x) = \theta_0 + \theta_1 x$

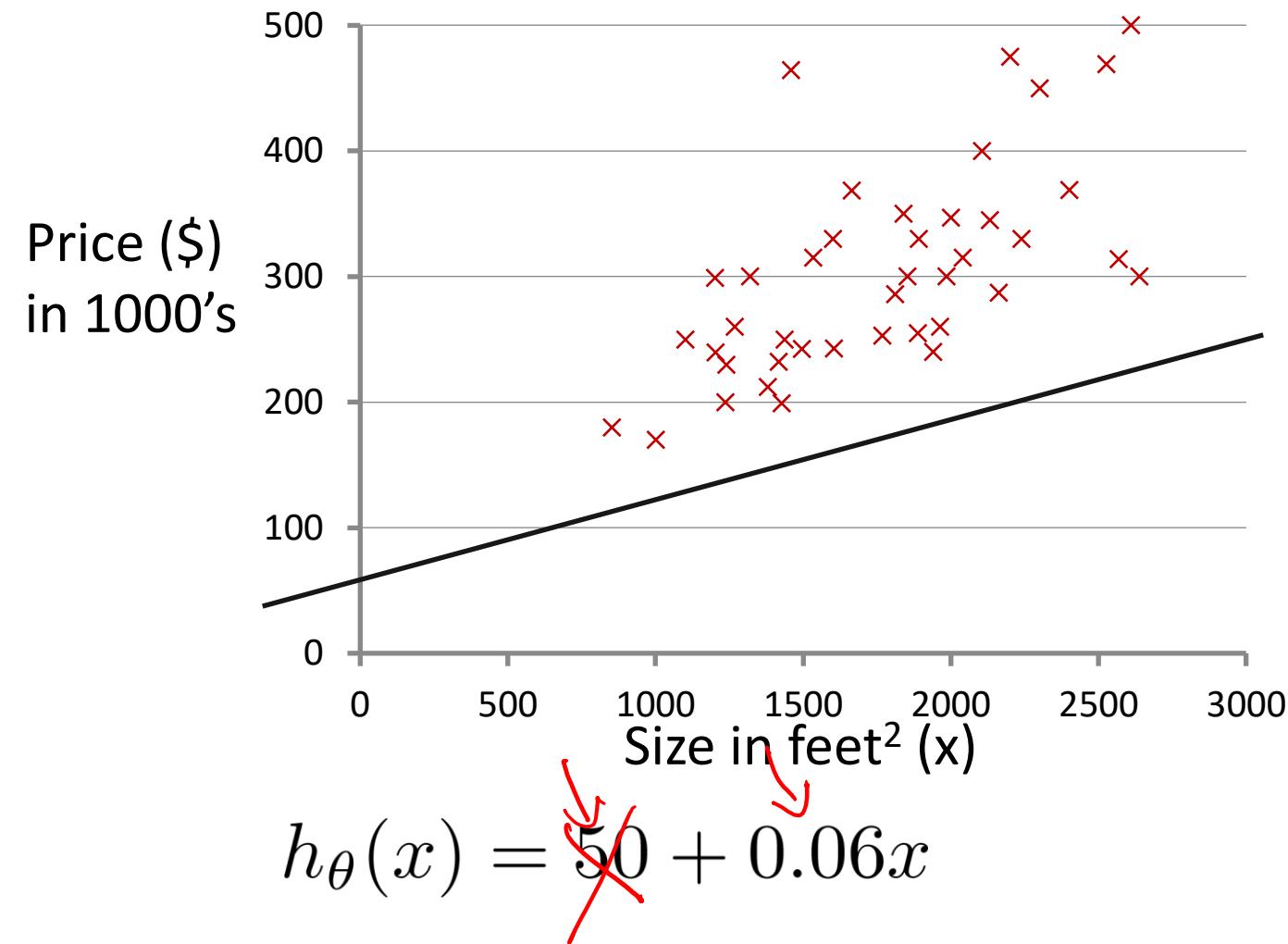
Paramètres: θ_0, θ_1

Fonction de coût: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

but: $\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$

La fonction de coût

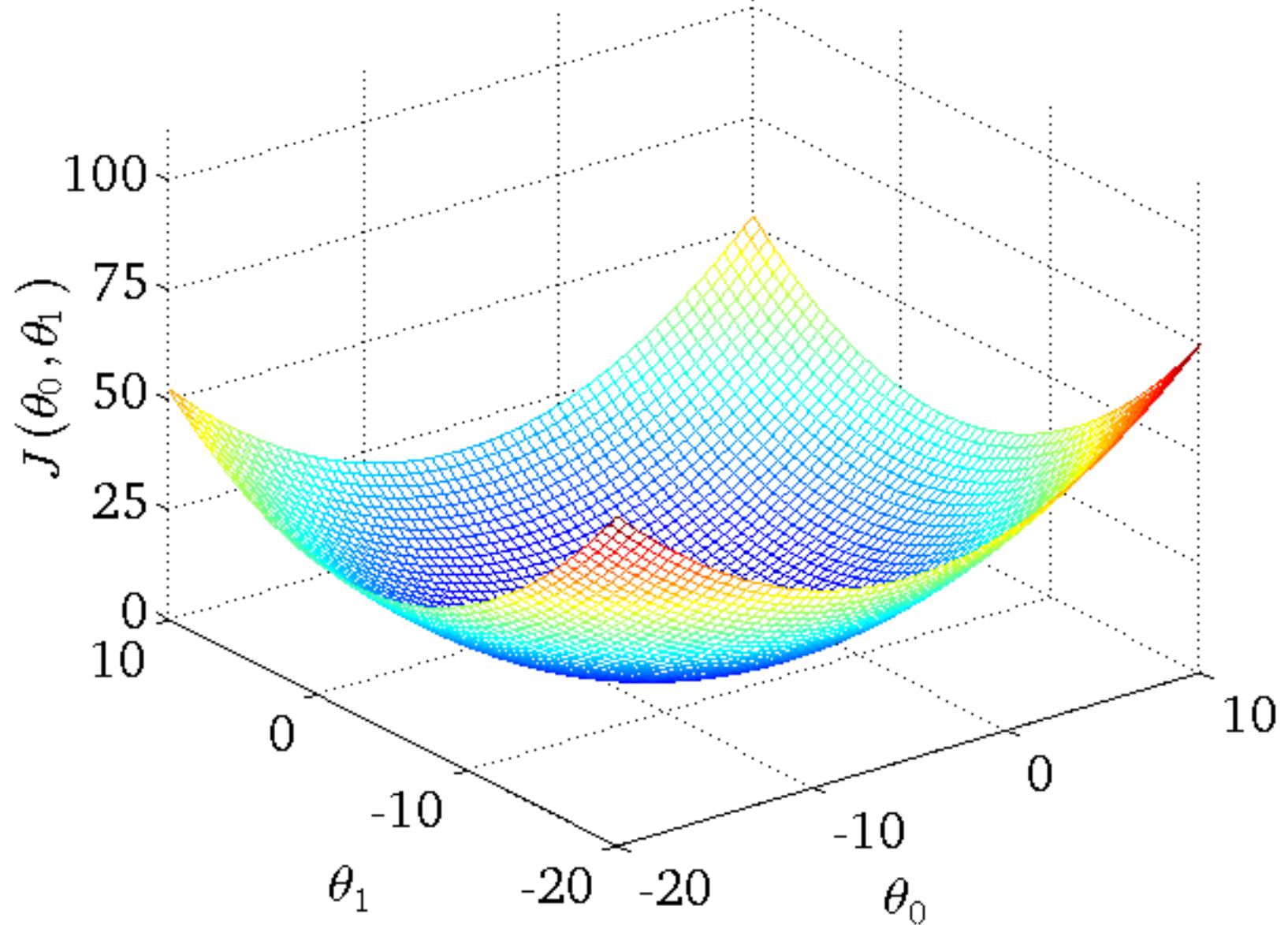
$h_\theta(x)$ (θ_0, θ_1 fixes, en fonction de x)



$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

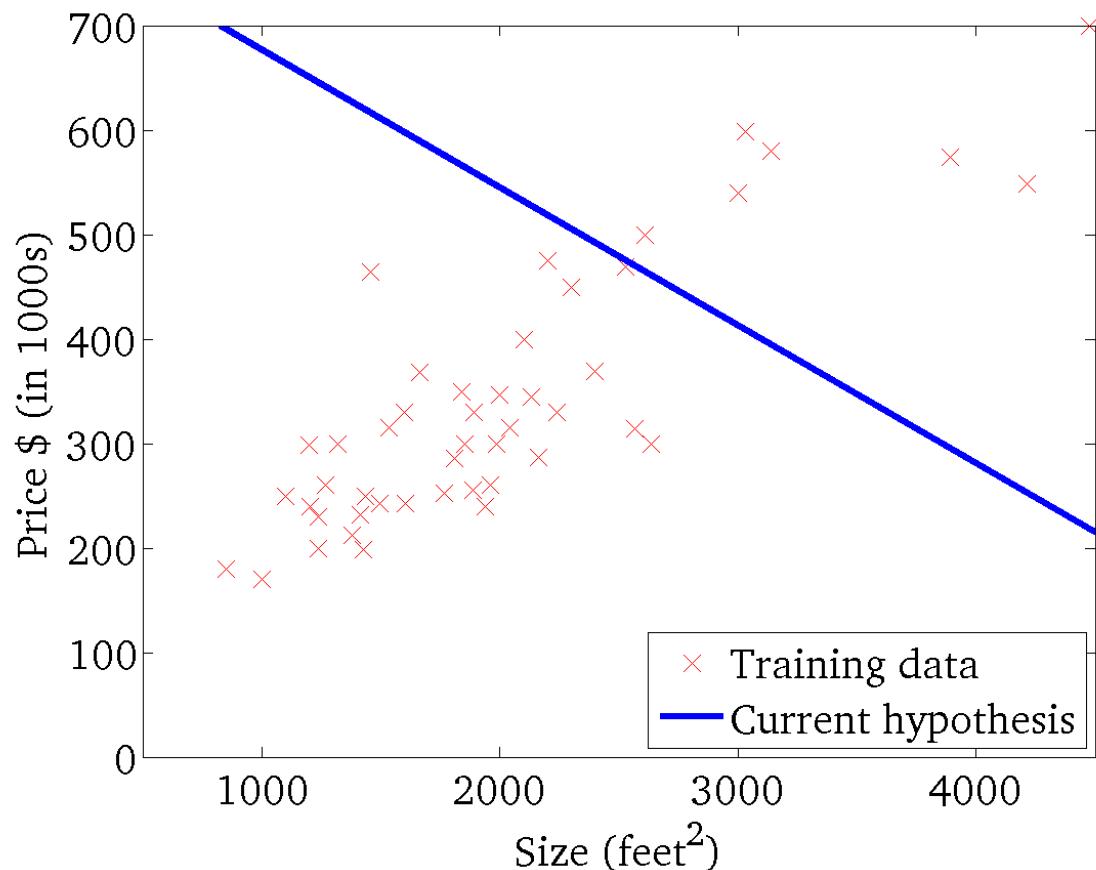


La fonction de coût

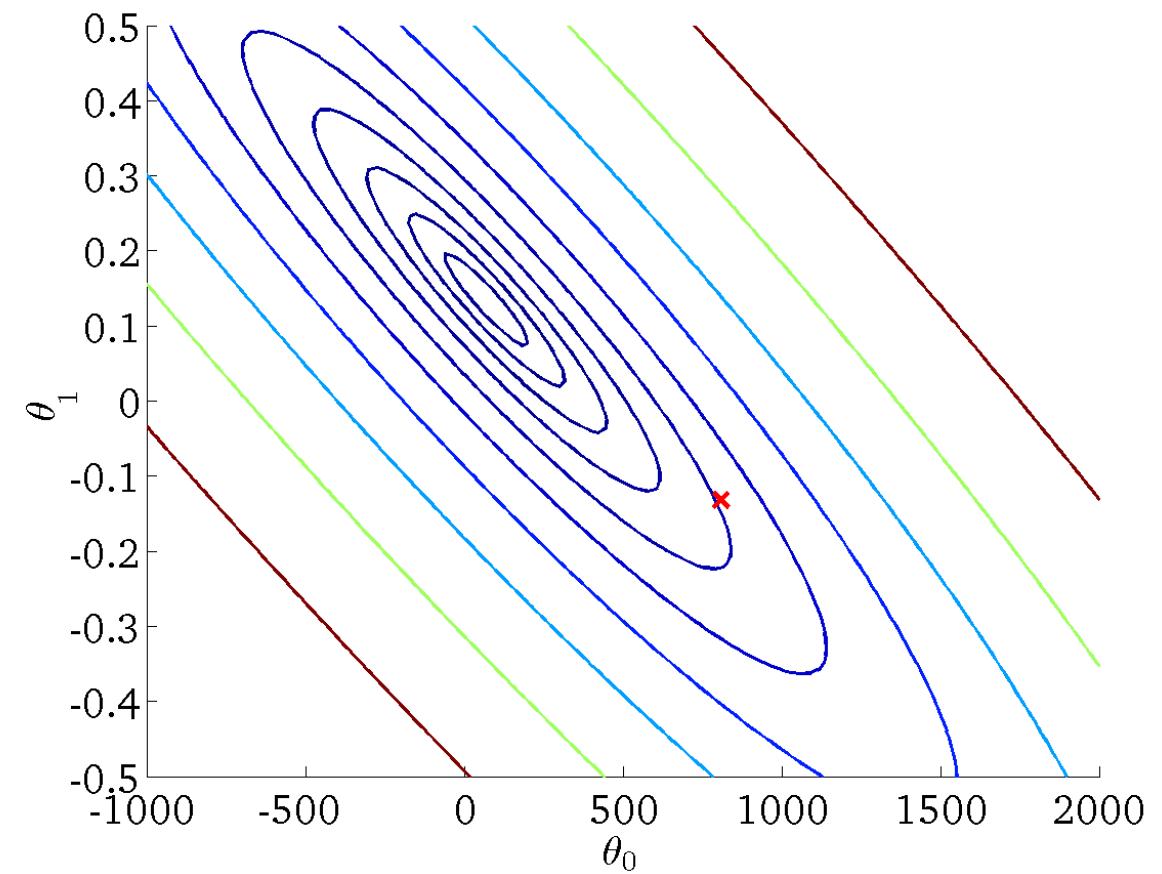


La fonction de coût

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

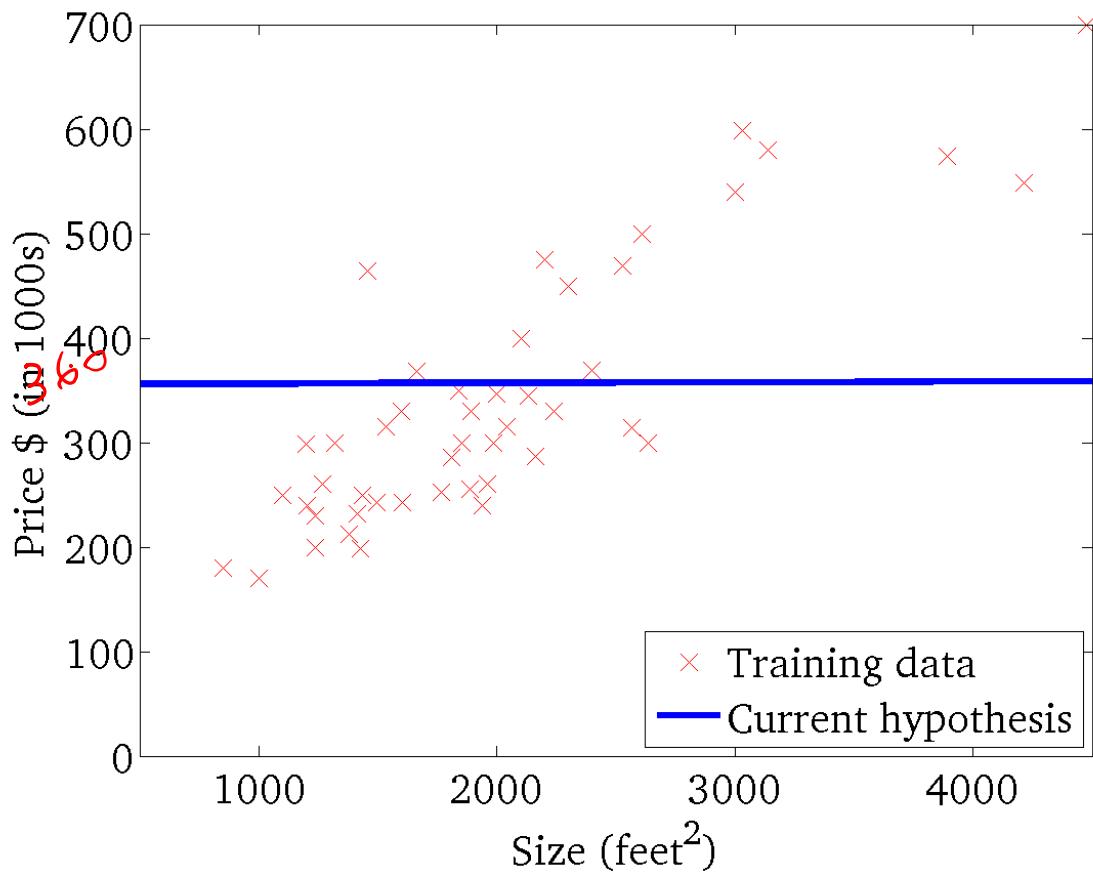


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)



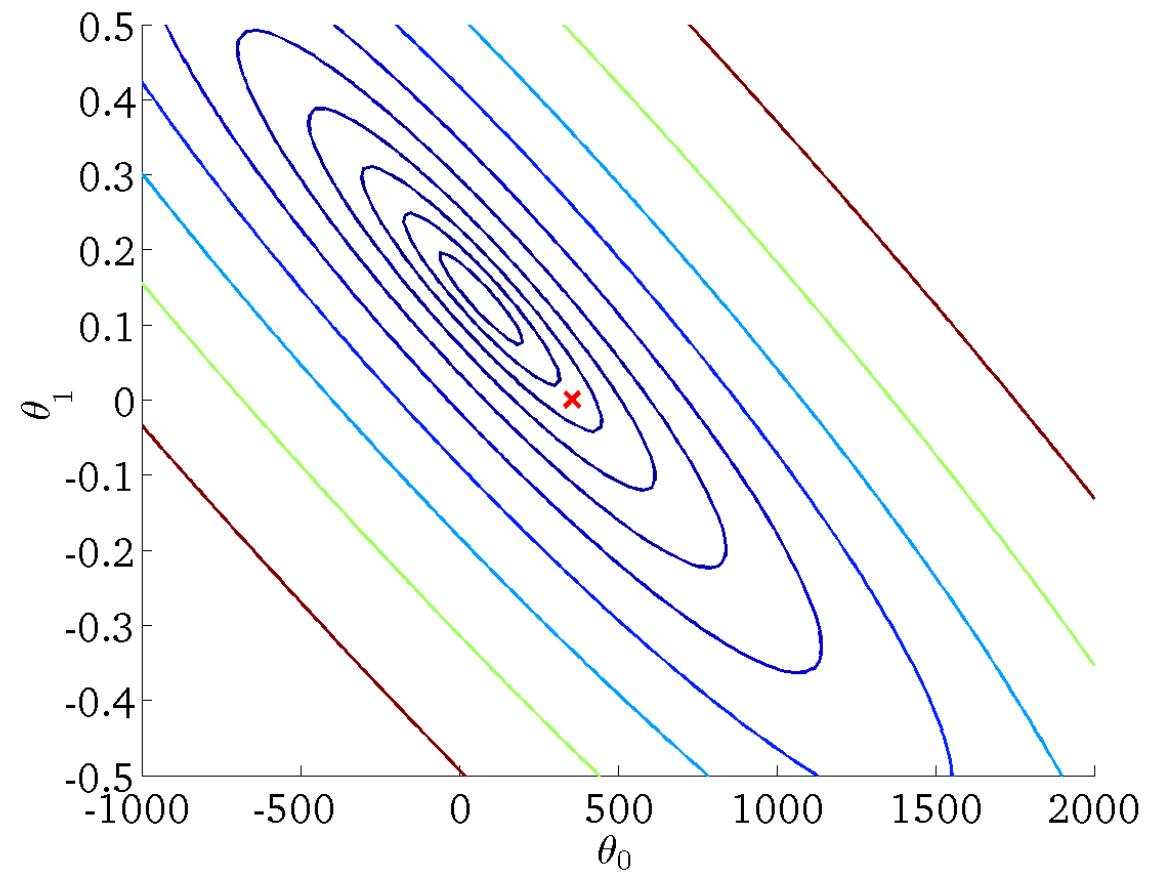
La fonction de coût

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)



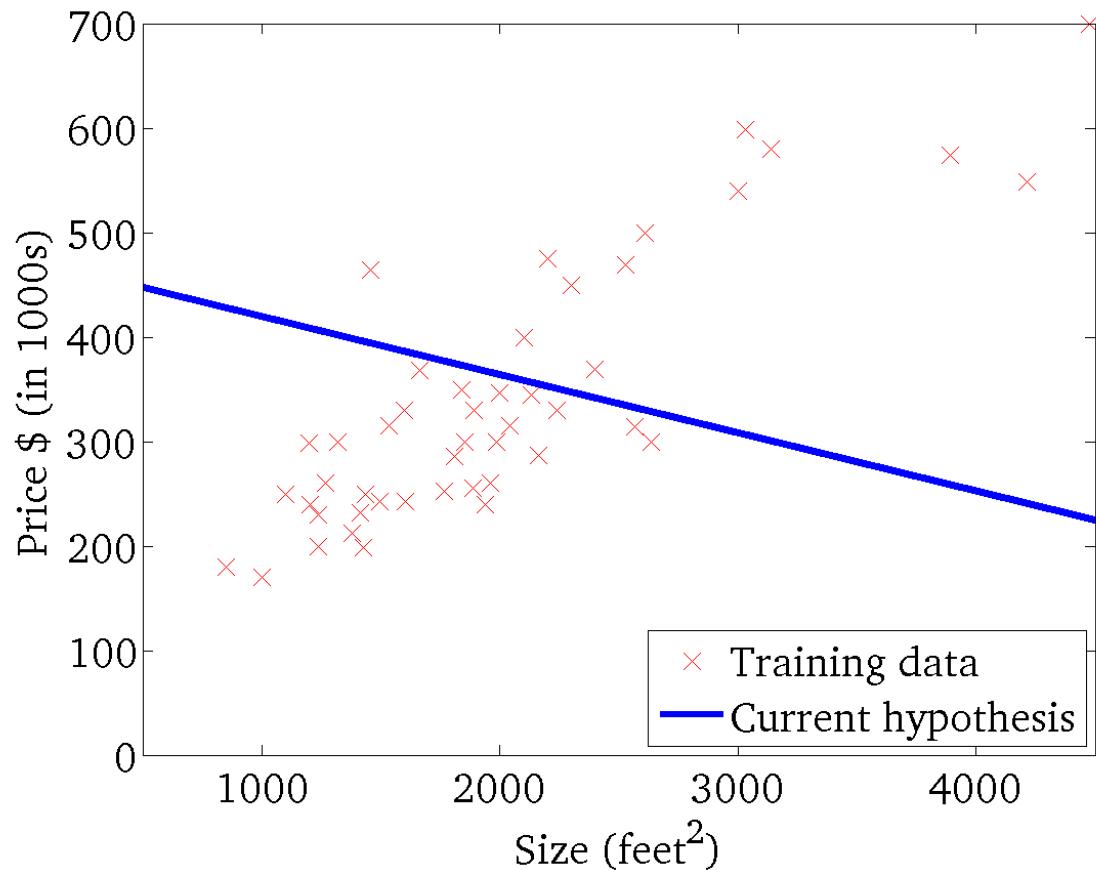
$$J(\theta_0, \theta_1)$$

(en fonction des paramètres θ_0, θ_1)



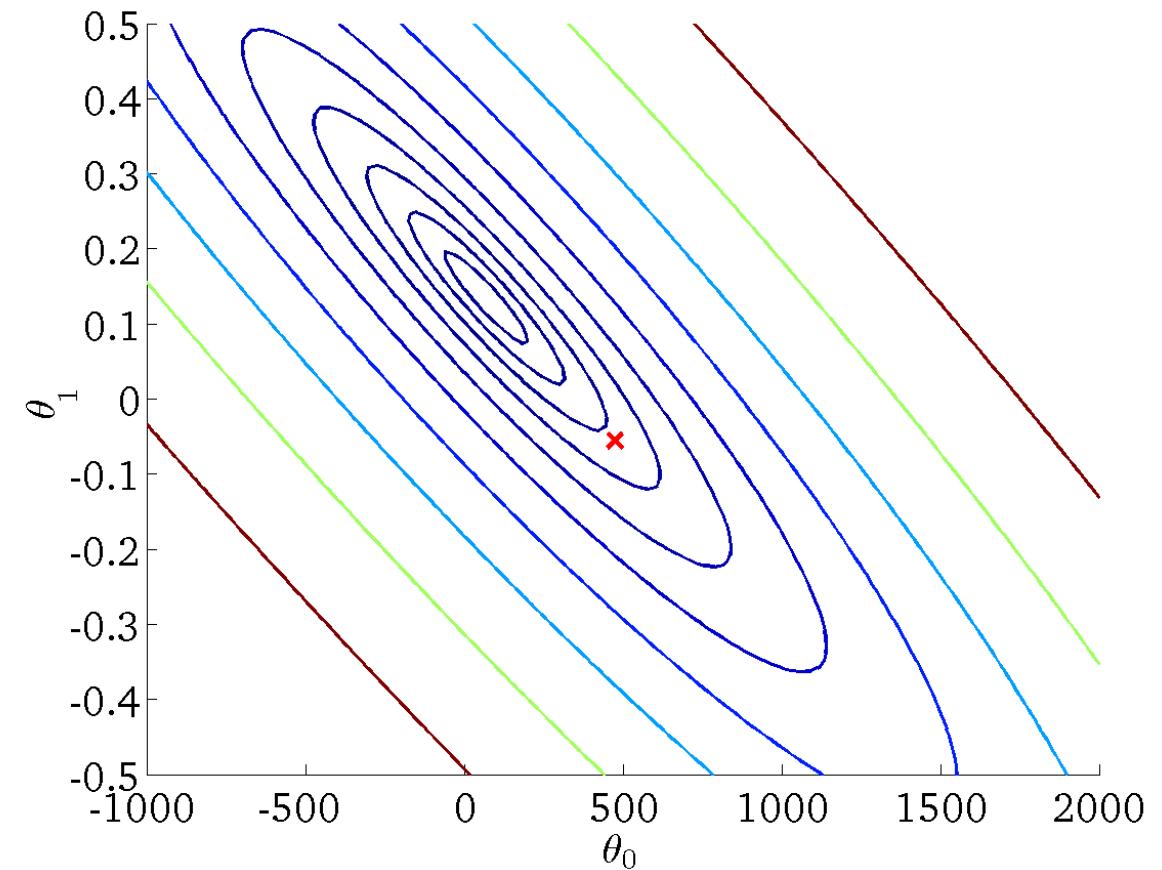
La fonction de coût

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)



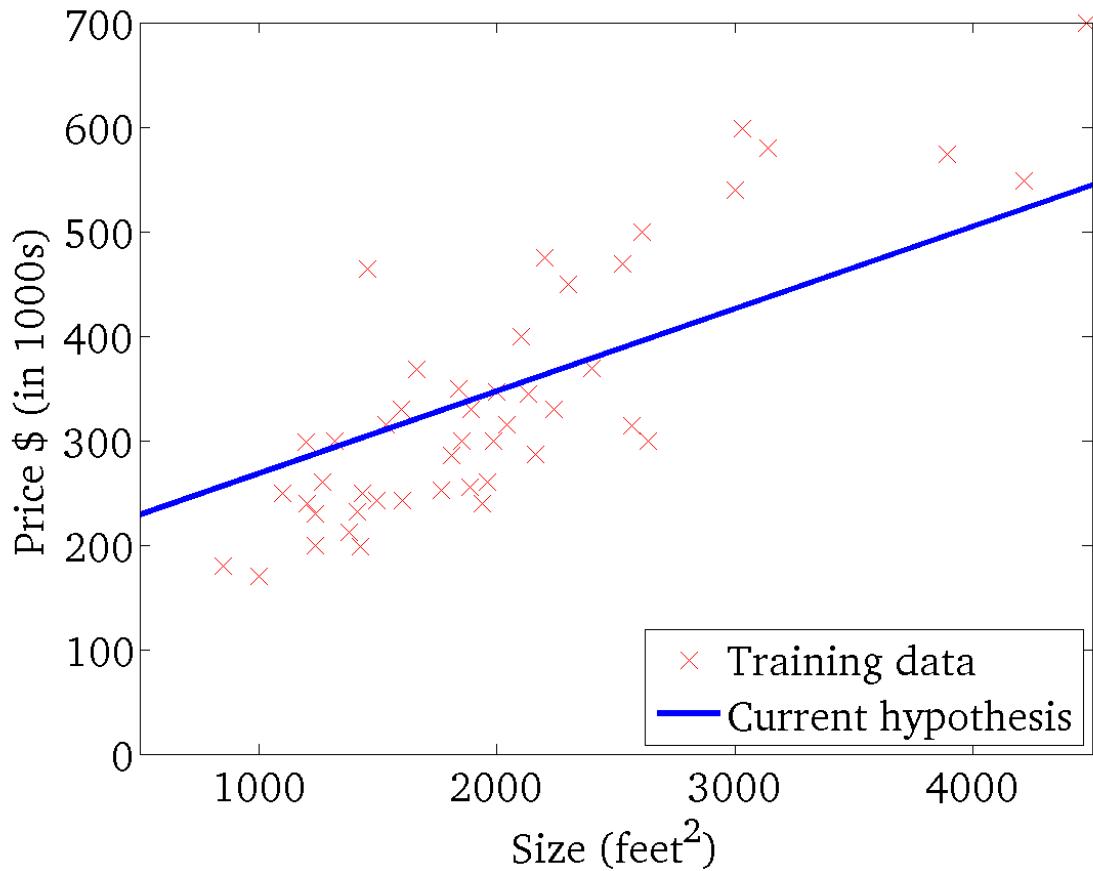
$$J(\theta_0, \theta_1)$$

(en fonction des paramètres θ_0, θ_1)

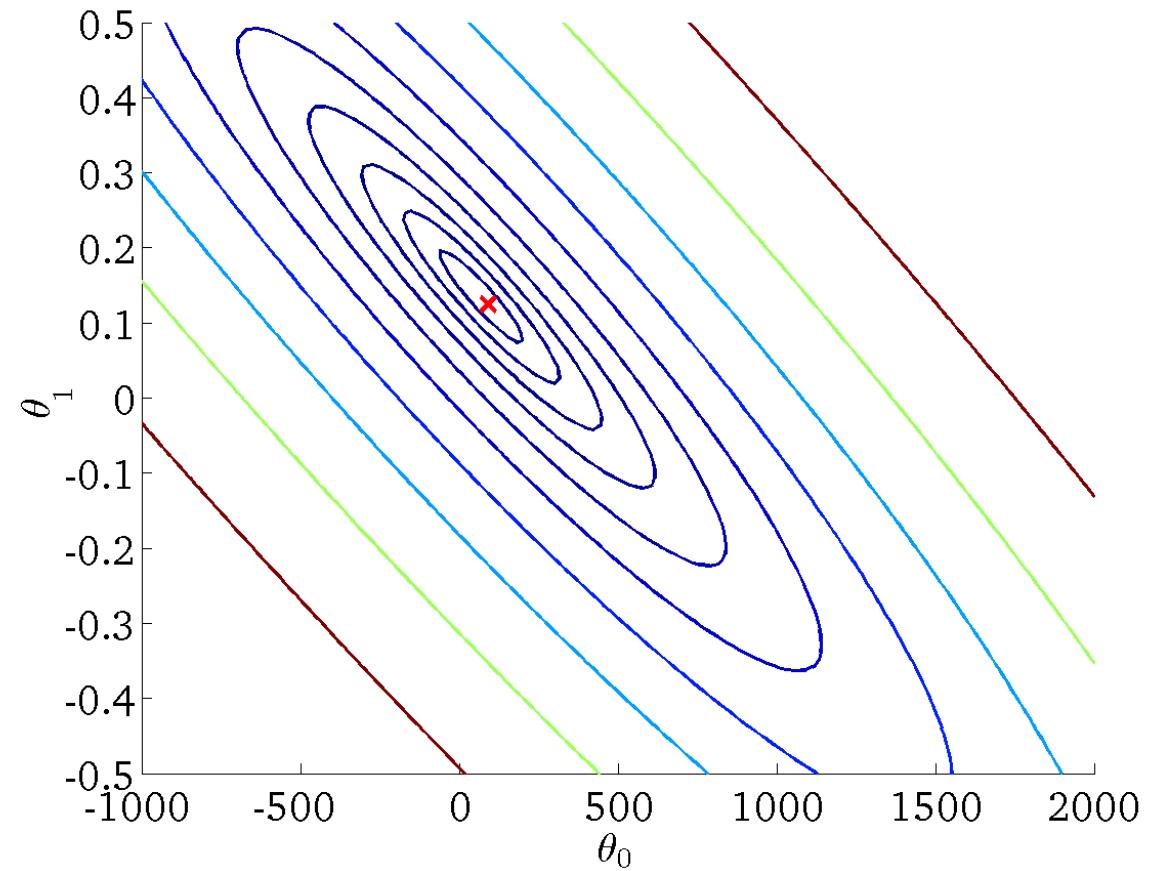


La fonction de coût

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)



$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)



La décente du gradient

Soit la function : $J(\theta_0, \theta_1)$

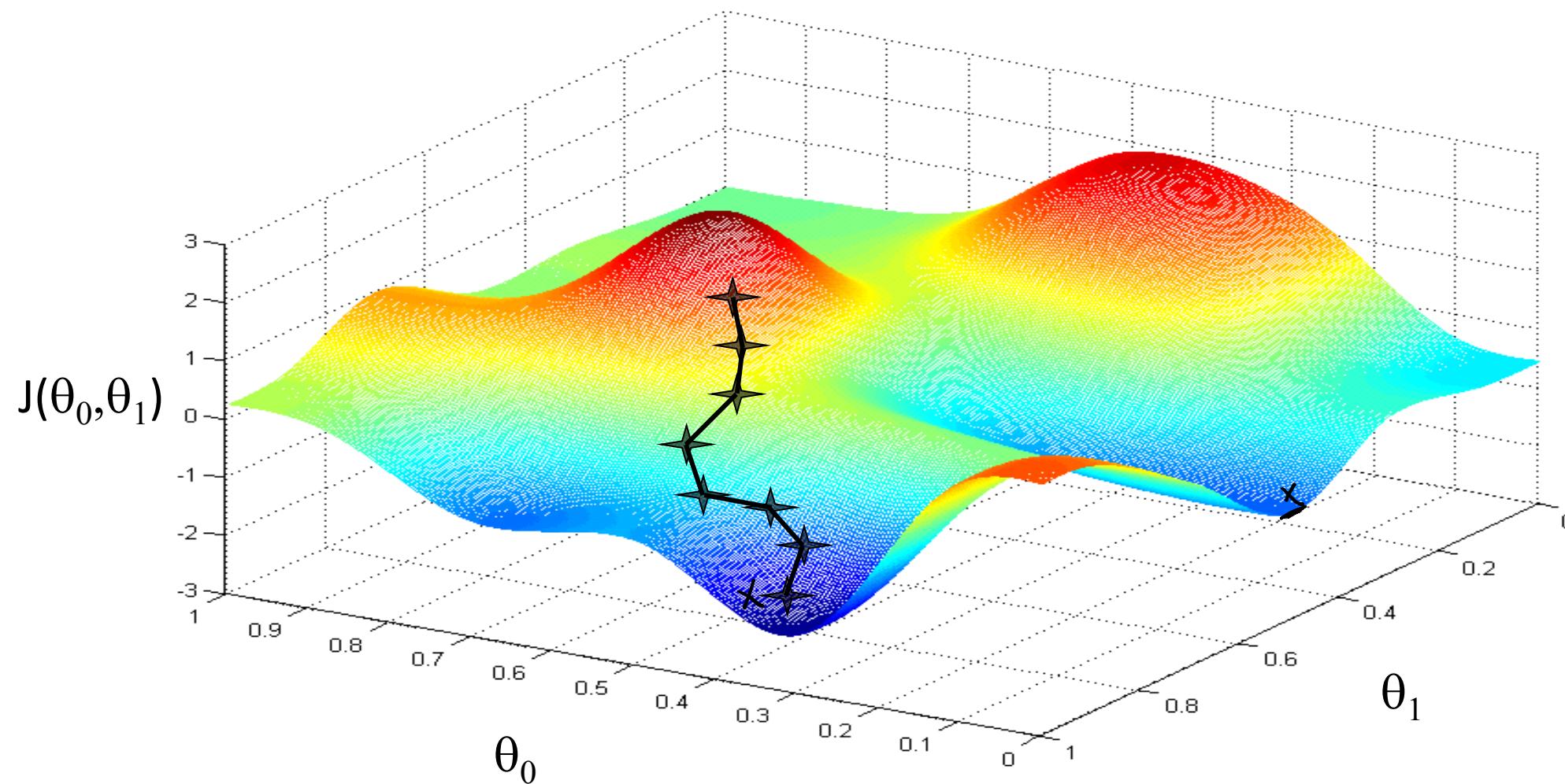
On veut $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Plan:

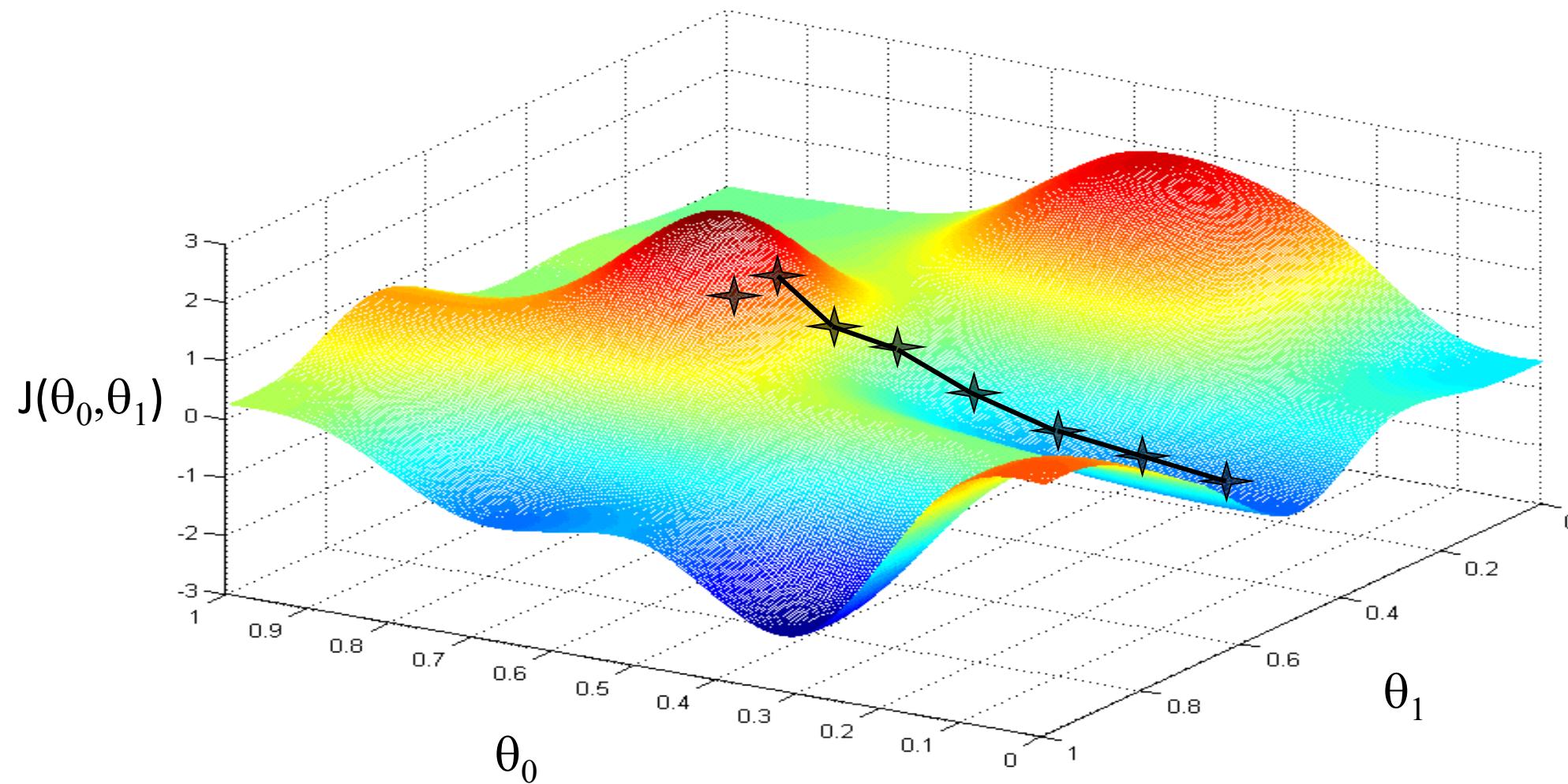
- On choisit arbitrairement θ_0, θ_1
- On modifie θ_0, θ_1 pour réduire $J(\theta_0, \theta_1)$

Jusqu'à un minimum

La décente du gradient



La décente du gradient



La décente du gradient : Algorithme

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$

}

Correct: Mise à jour simultanée

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$

Incorrect:

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\theta_0 := \text{temp0}$$

$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_1 := \text{temp1}$$

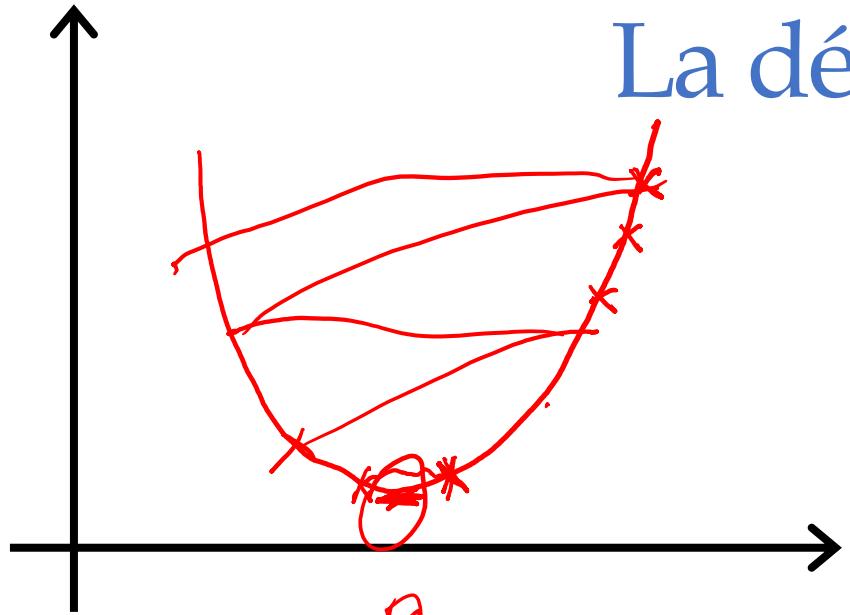
La décente du gradient : Algorithme

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \begin{matrix} \text{(simultaneously update} \\ j = 0 \text{ and } j = 1 \end{matrix}$$

}

La décente du gradient

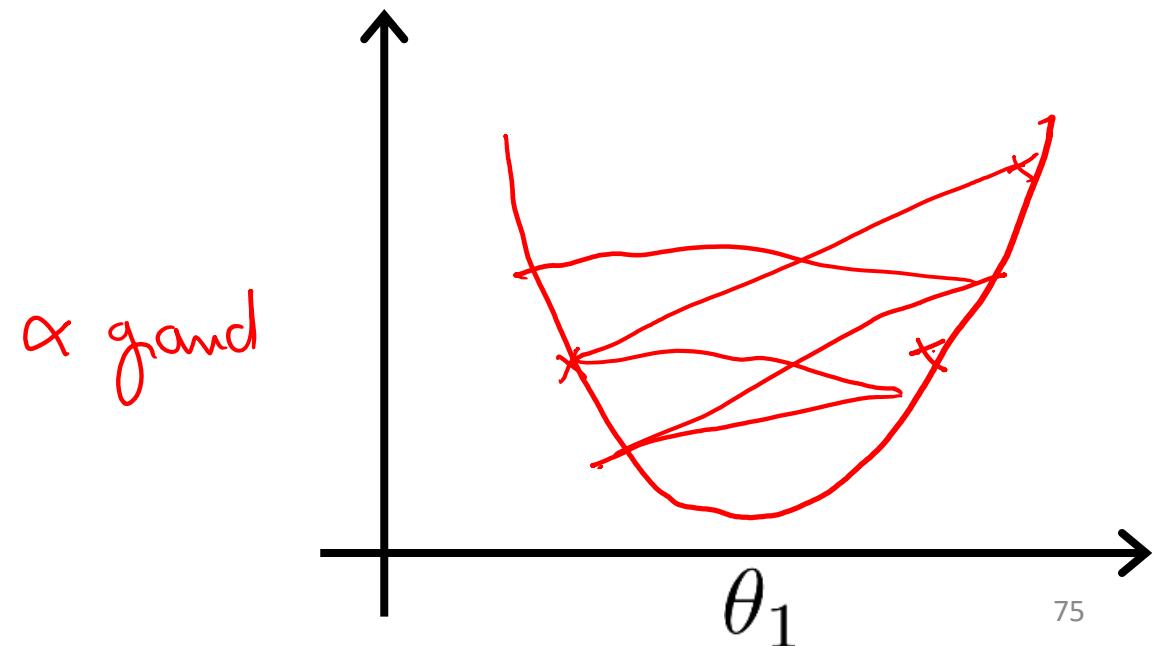
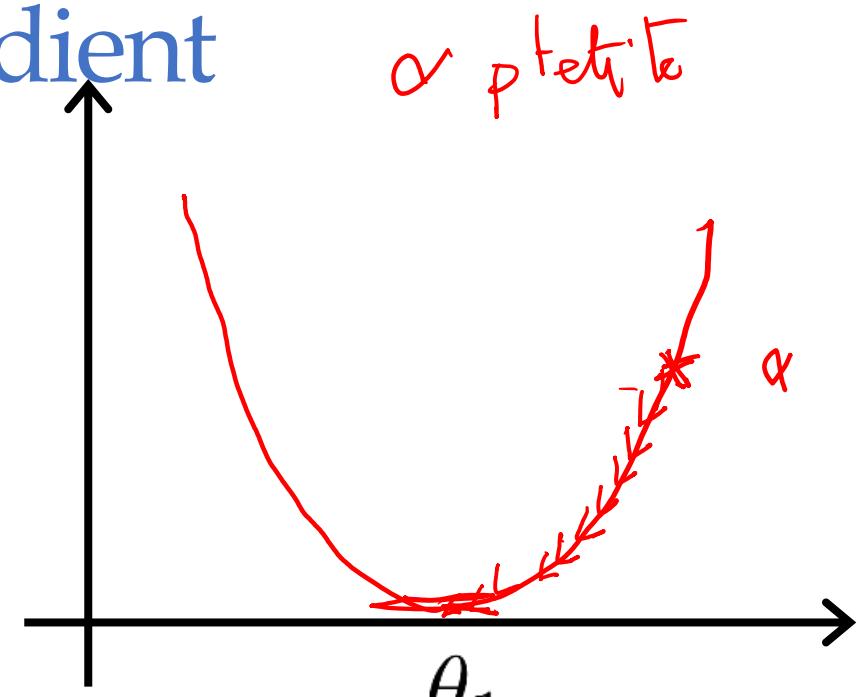


La décente du gradient

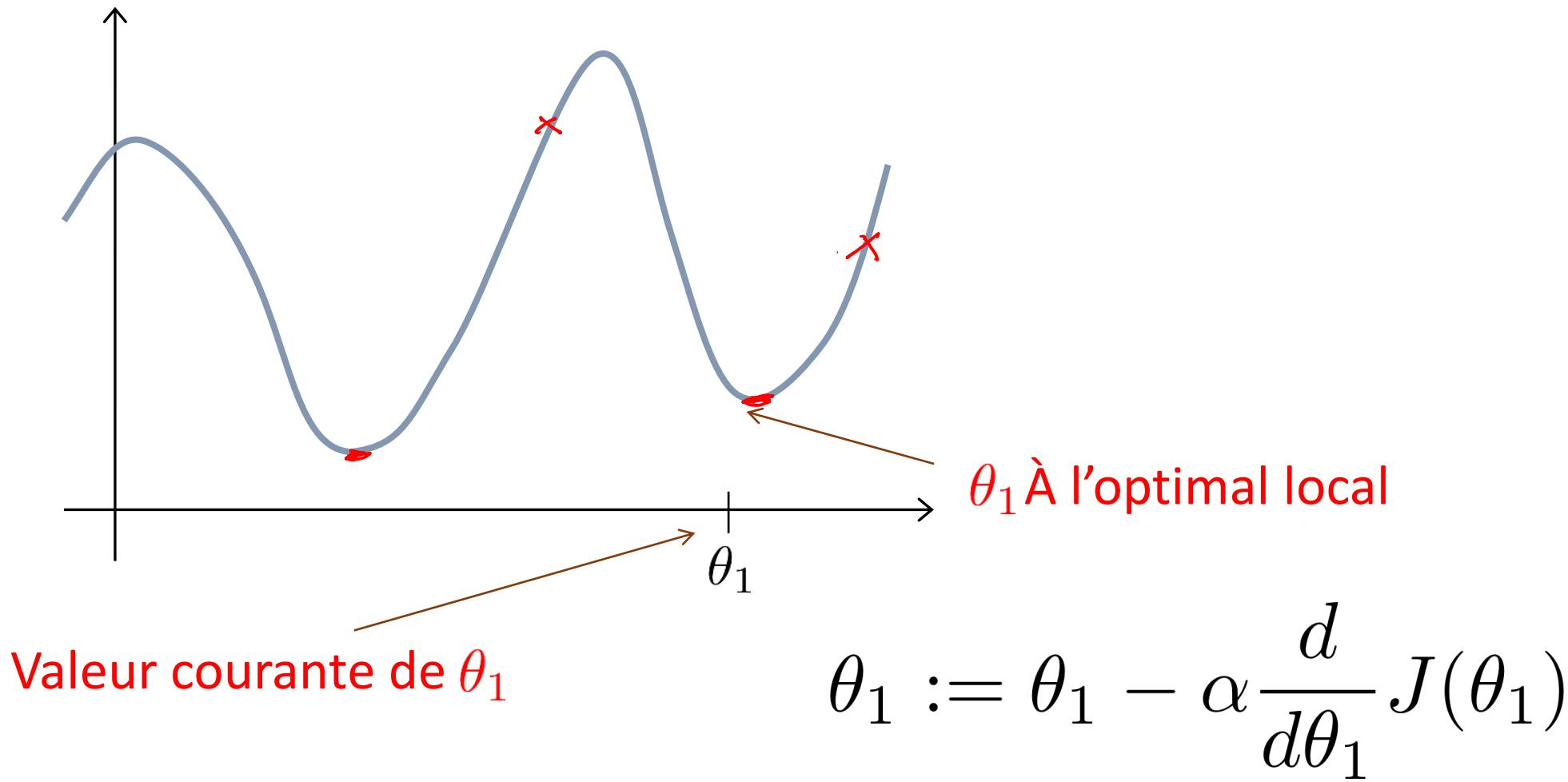
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Si α est très petit la décente du gradient sera lente

Si α est trop grand, la descente de gradient peut dépasser le minimum. Il peut ne pas converger, voire diverger.



La décente du gradient

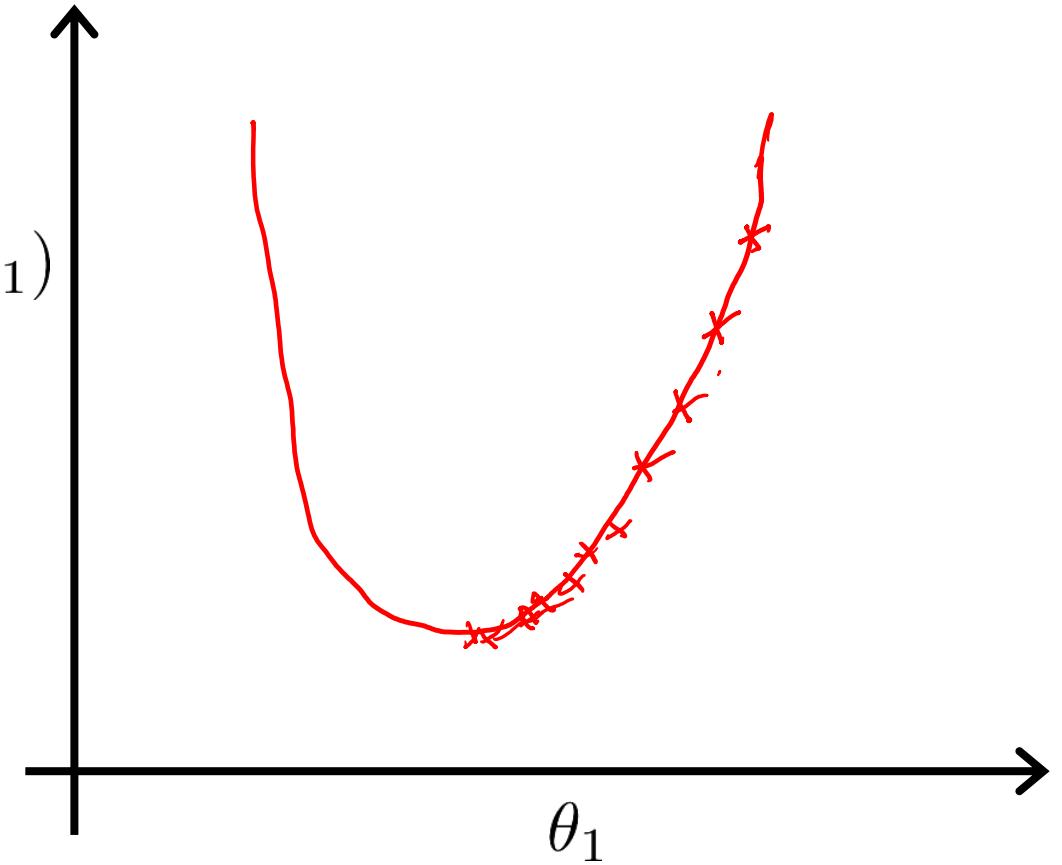


La décente du gradient

La descente de gradient peut converger vers un minimum local, même avec le taux d'apprentissage α fixé.

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

À mesure que nous approchons d'un minimum local, la descente de gradient effectuera automatiquement des pas plus petits. Donc, pas besoin de diminuer α au fil du temps.



La décente du gradient

Algorithme de la décente du gradient

```
repeat until convergence {  
     $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$   
    (for  $j = 1$  and  $j = 0$ )  
}
```

Modèle de regression linéaire

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

La décente du gradient

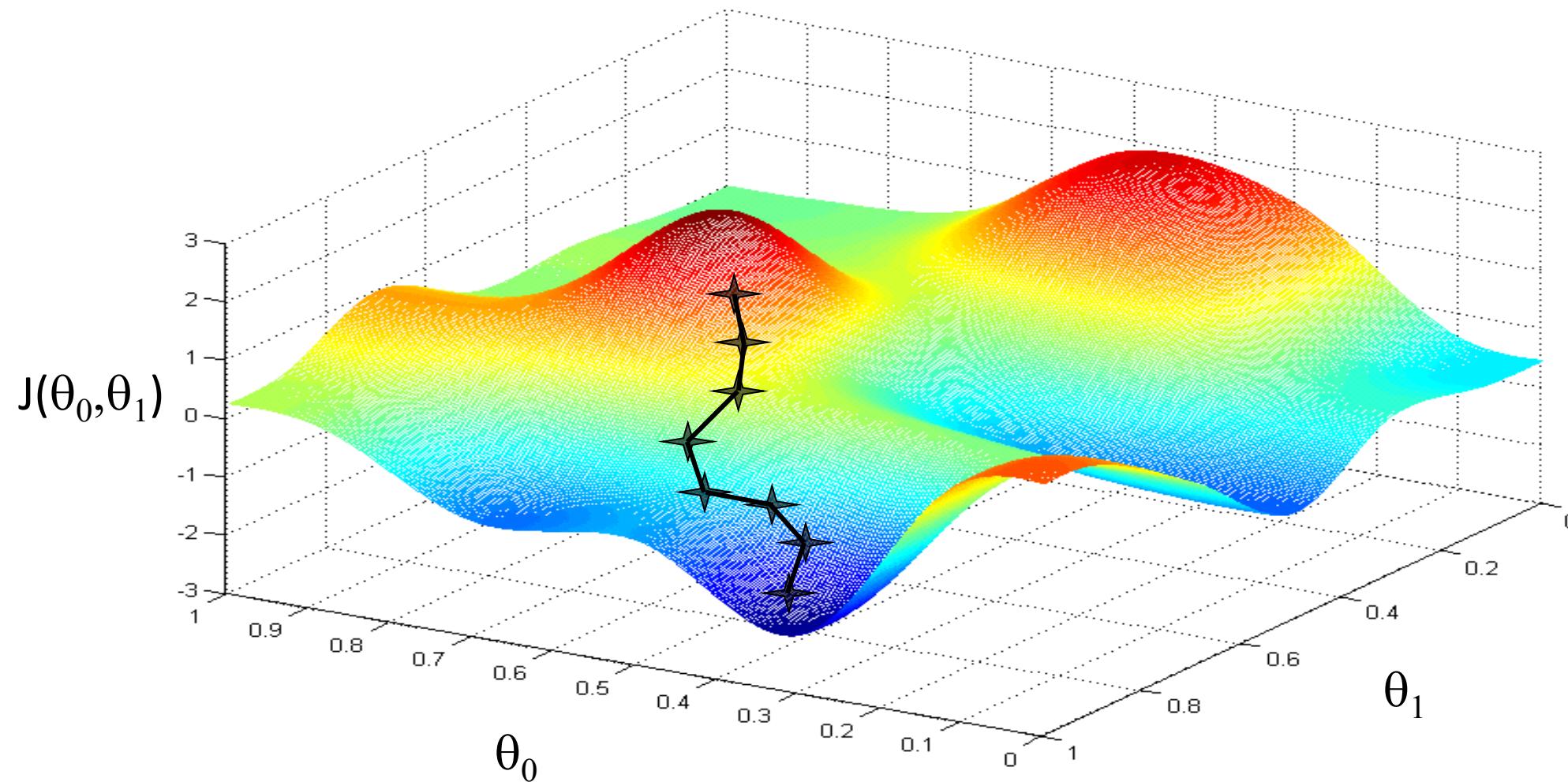
repeat until convergence {

$$\left. \begin{aligned} \theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \end{aligned} \right\}$$

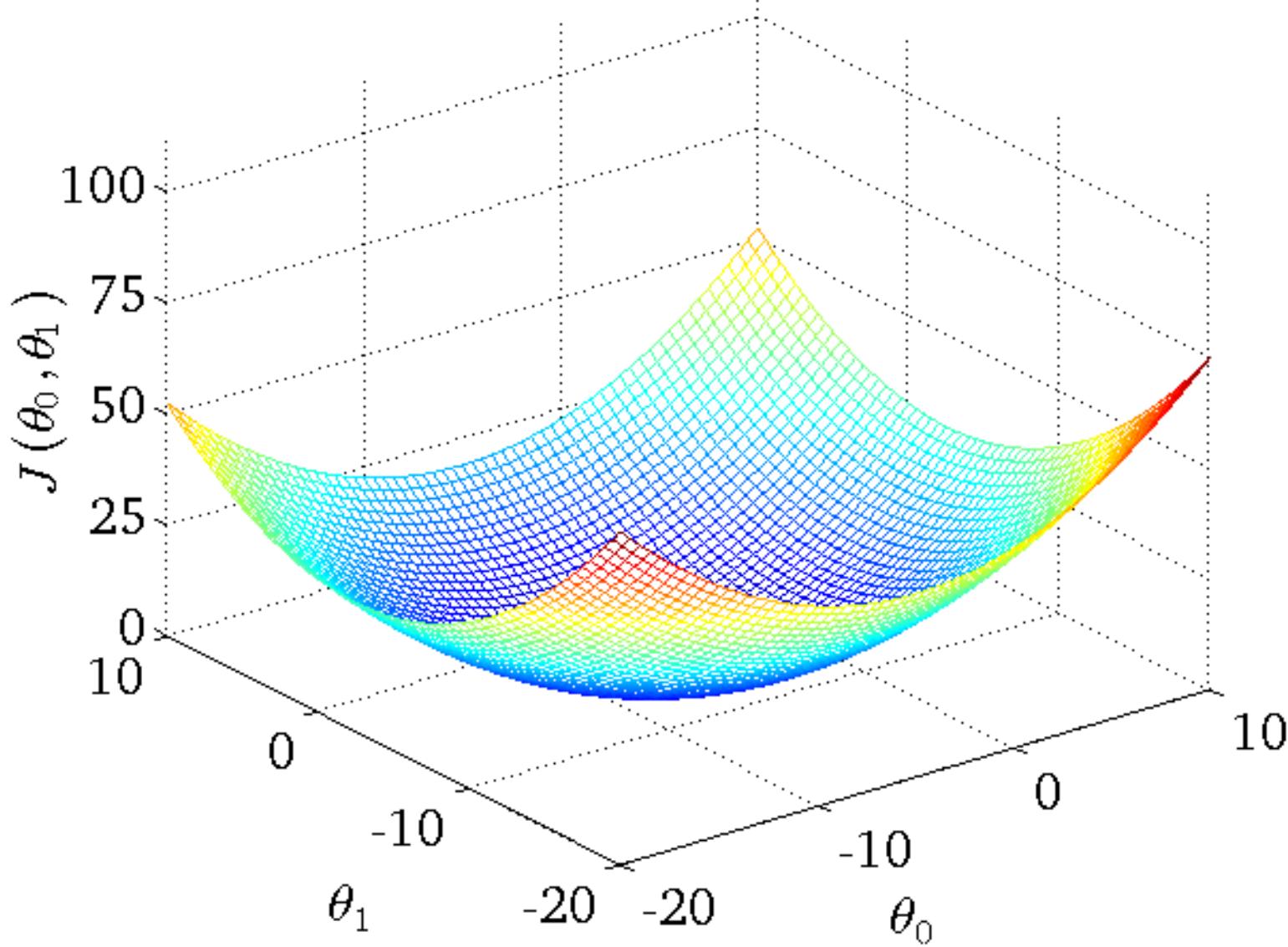
}

update
 θ_0 and θ_1
simultaneously

La décente du gradient

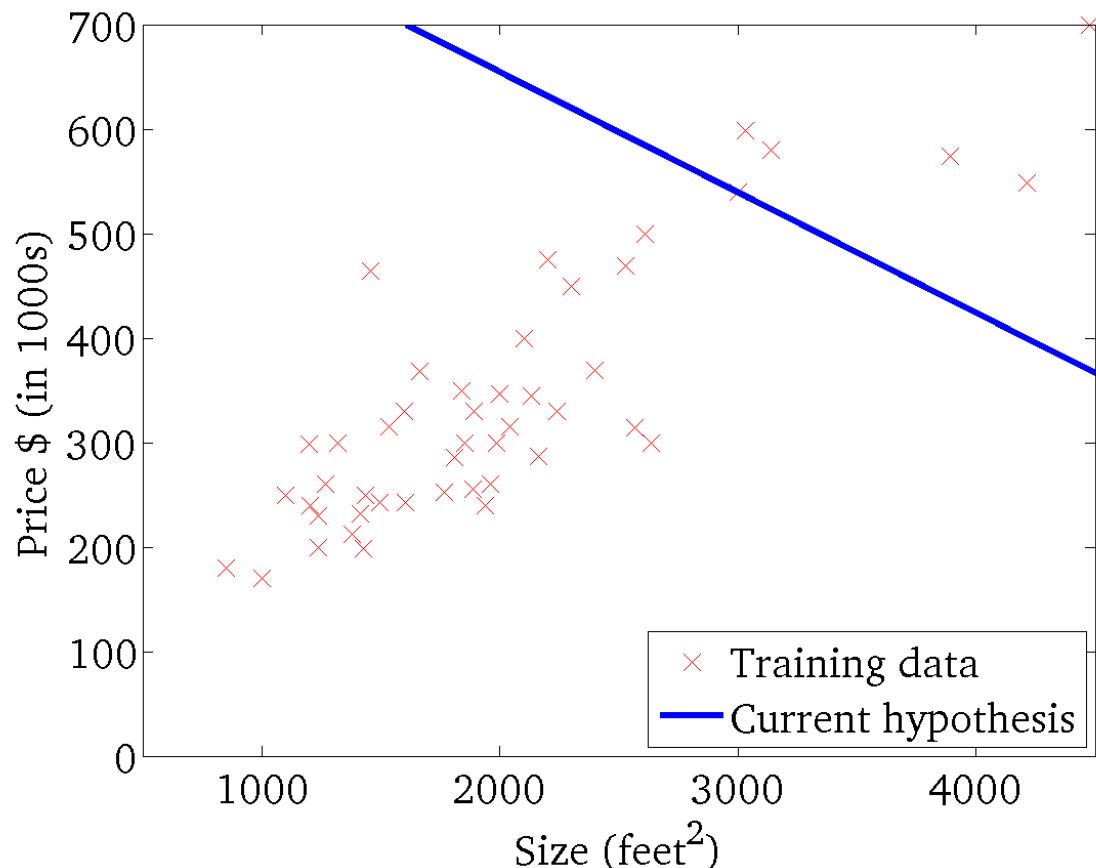


La décente du gradient

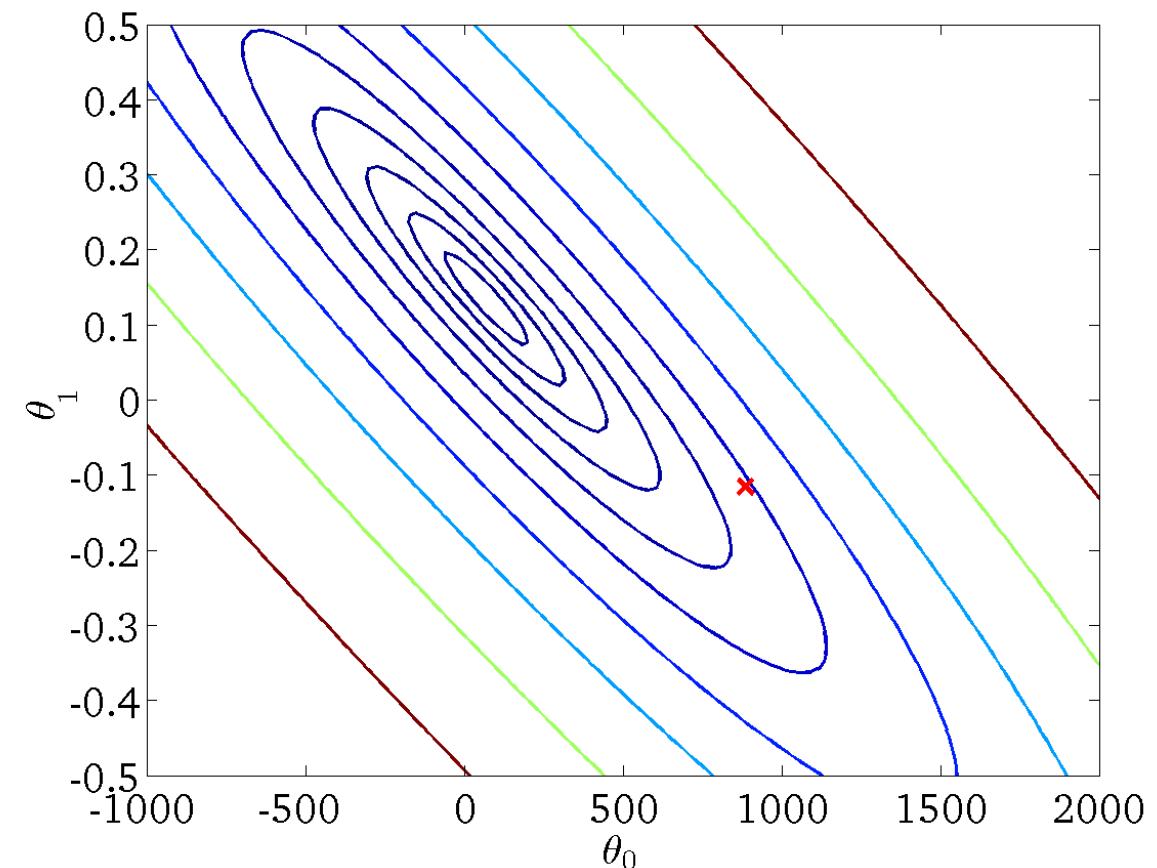


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

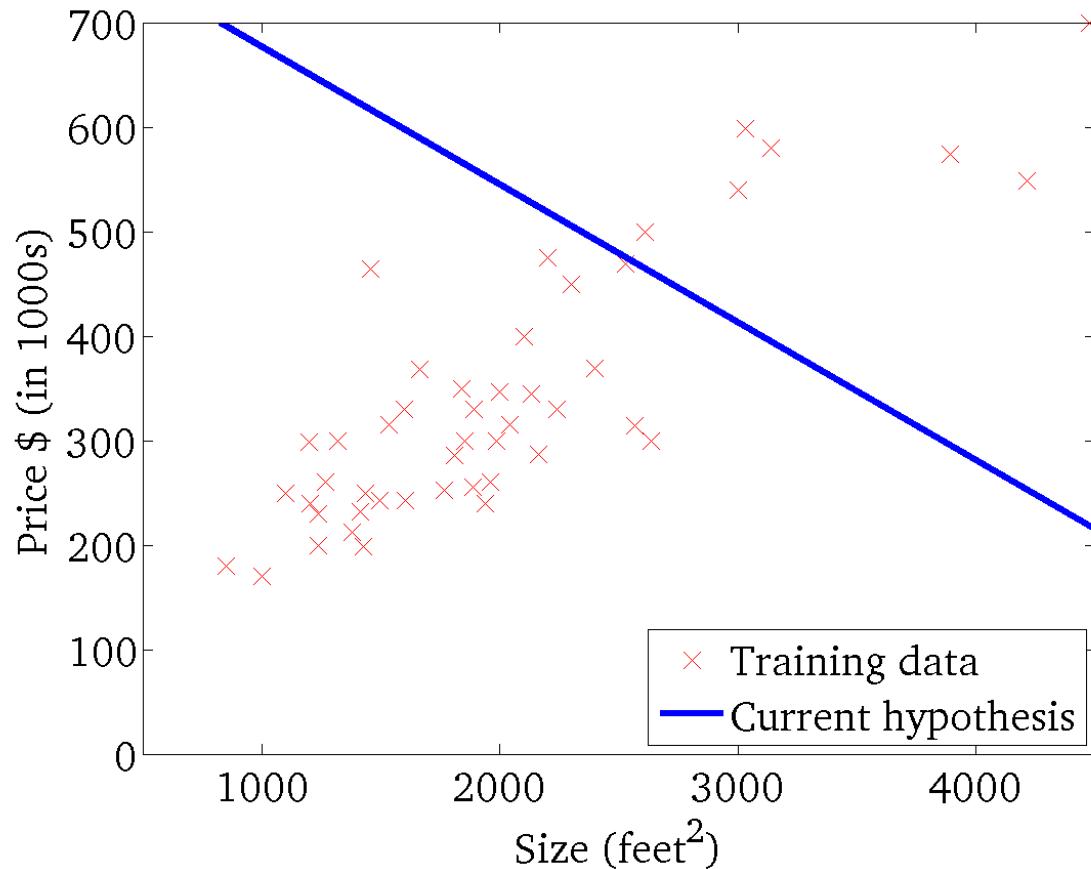


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

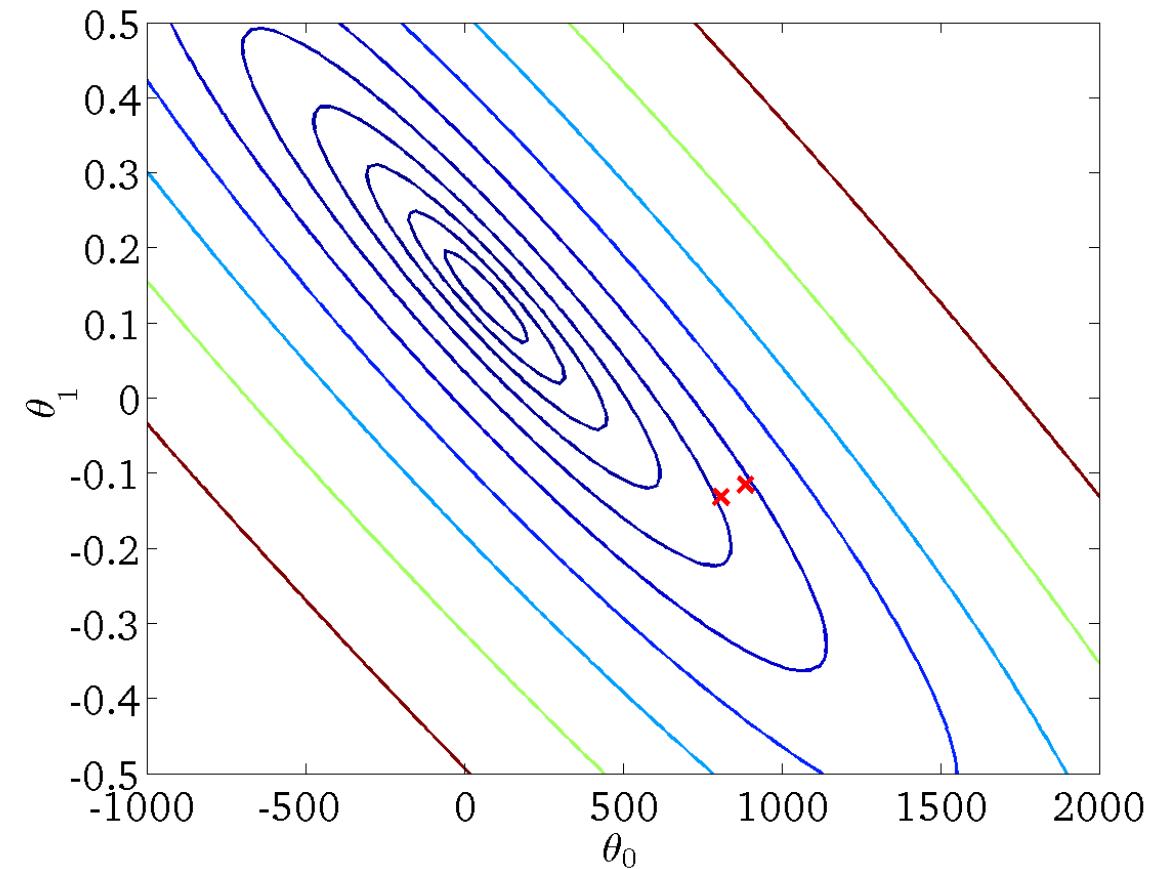


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

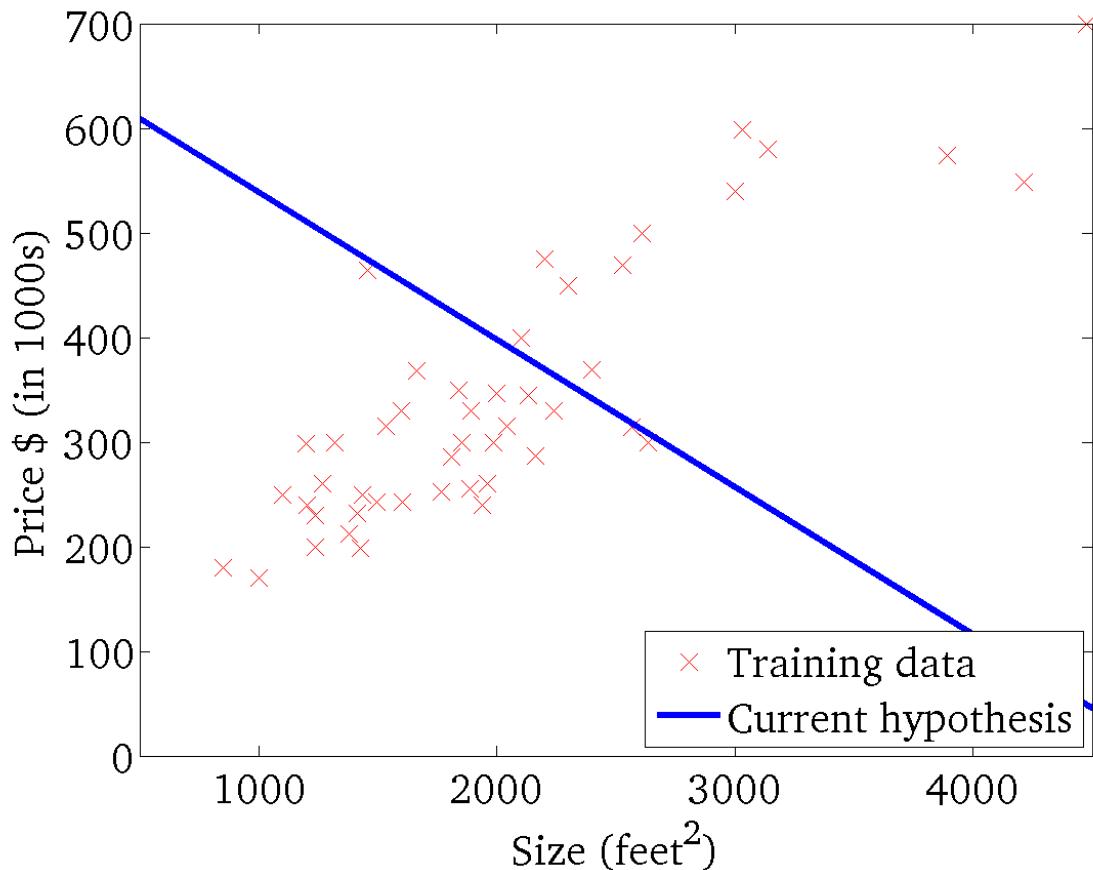


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

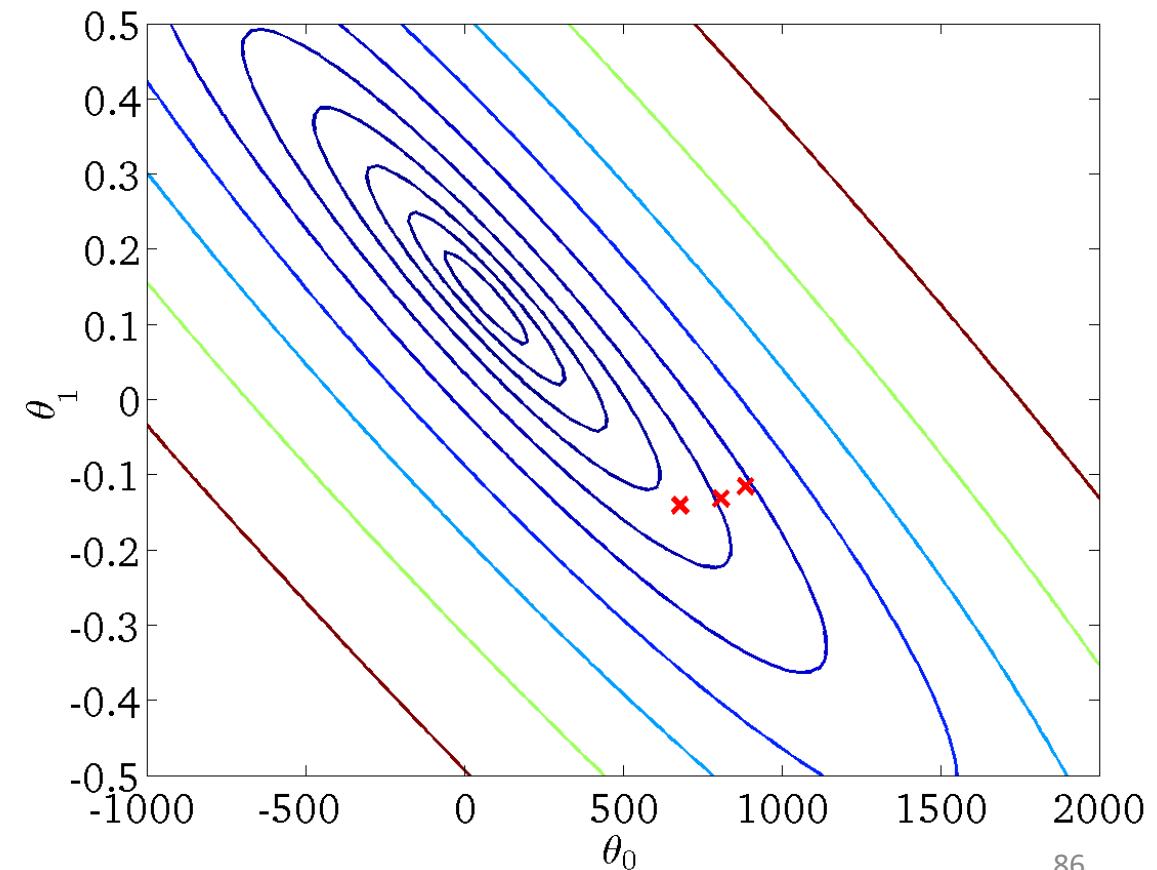


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

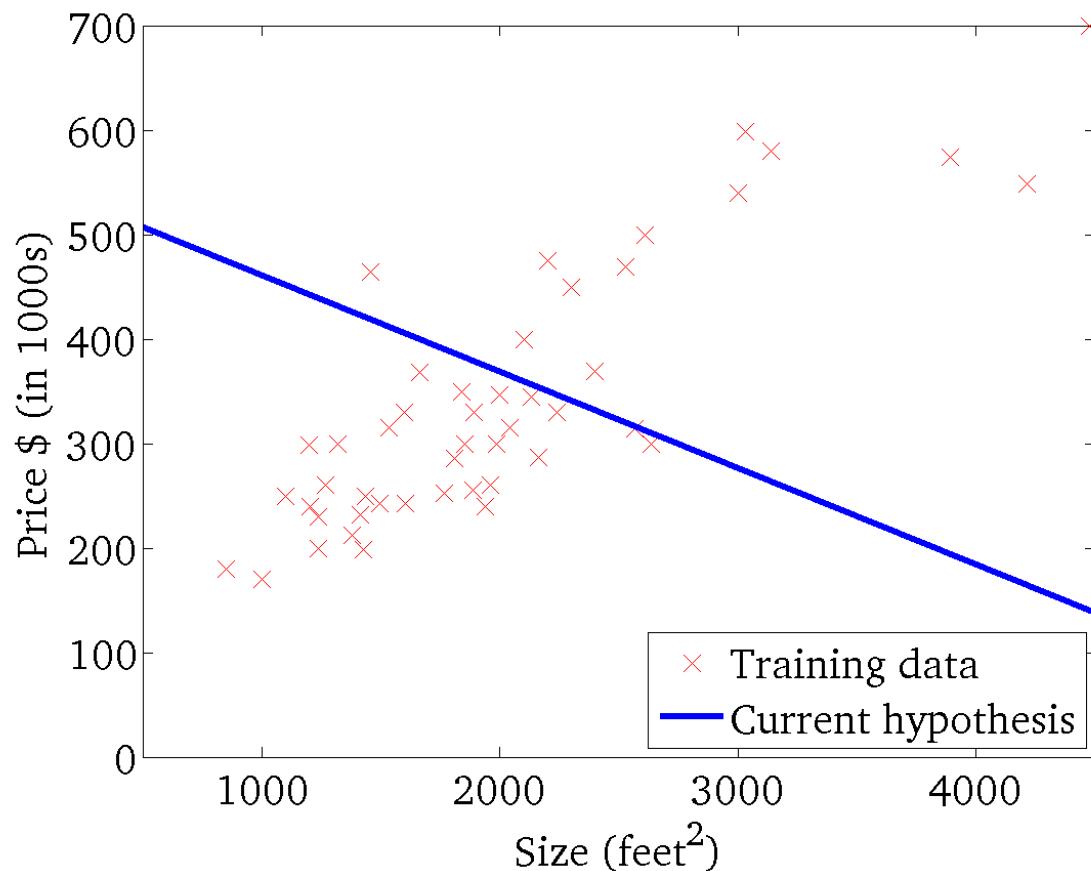


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

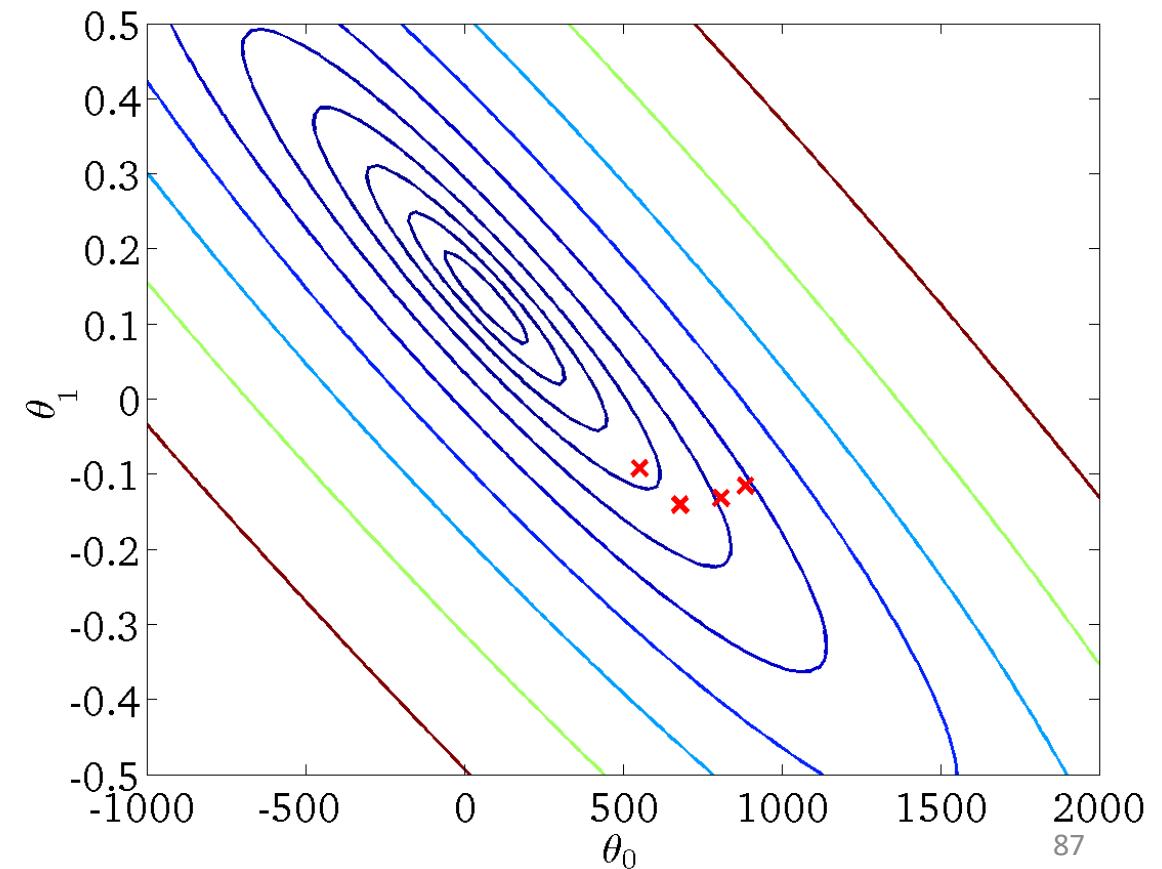


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

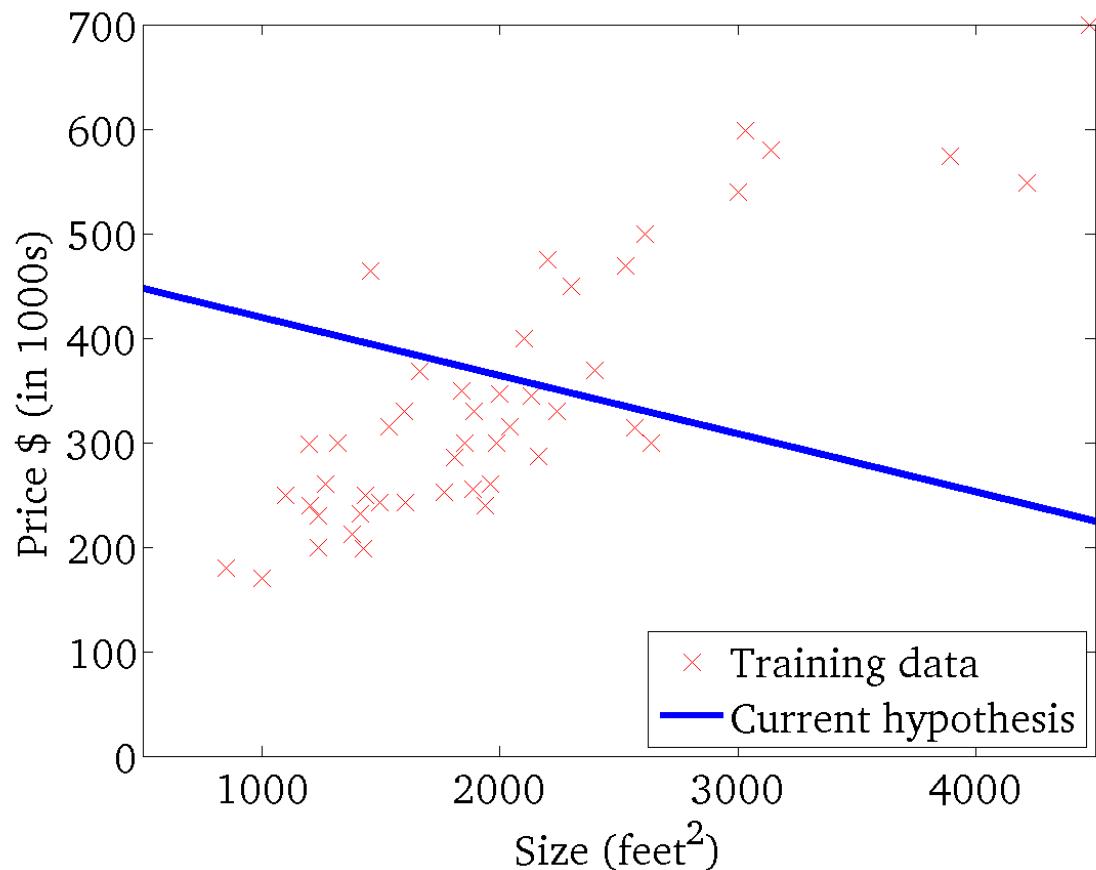


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

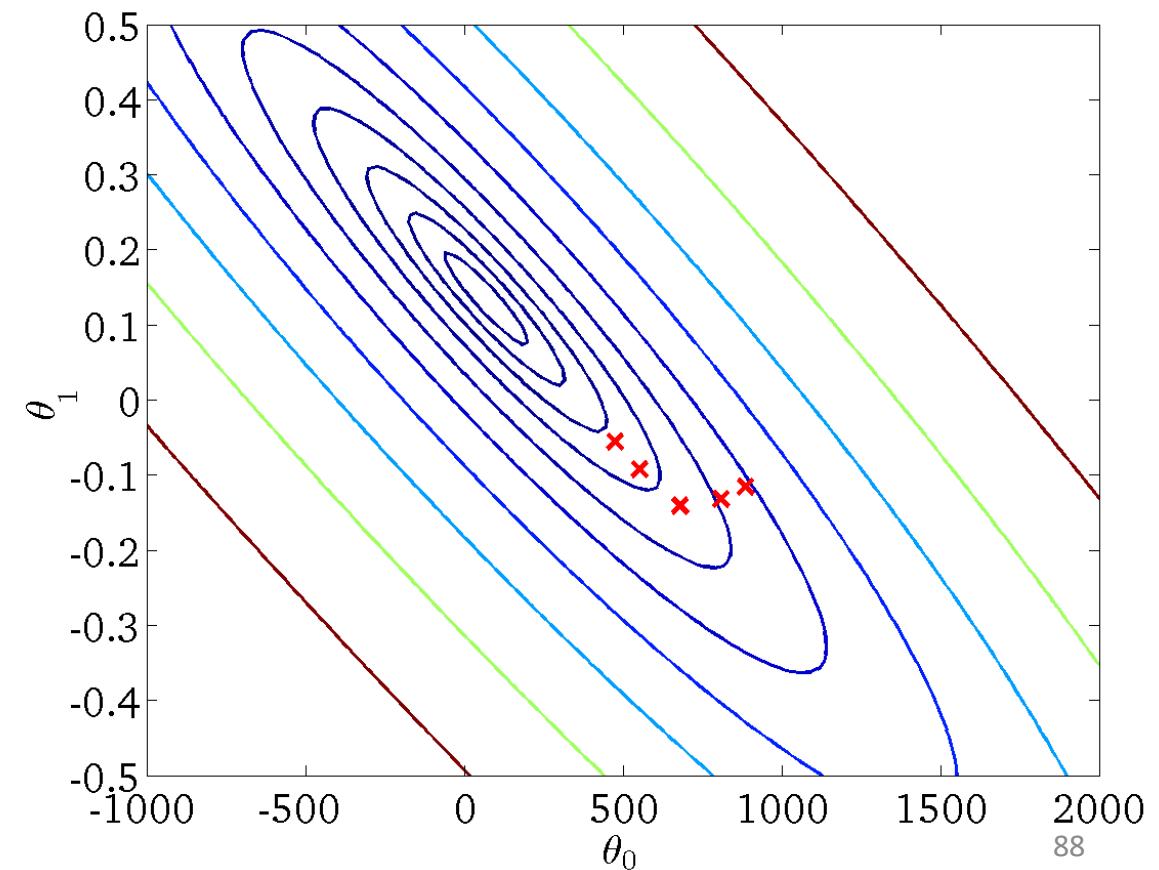


La décente du gradient

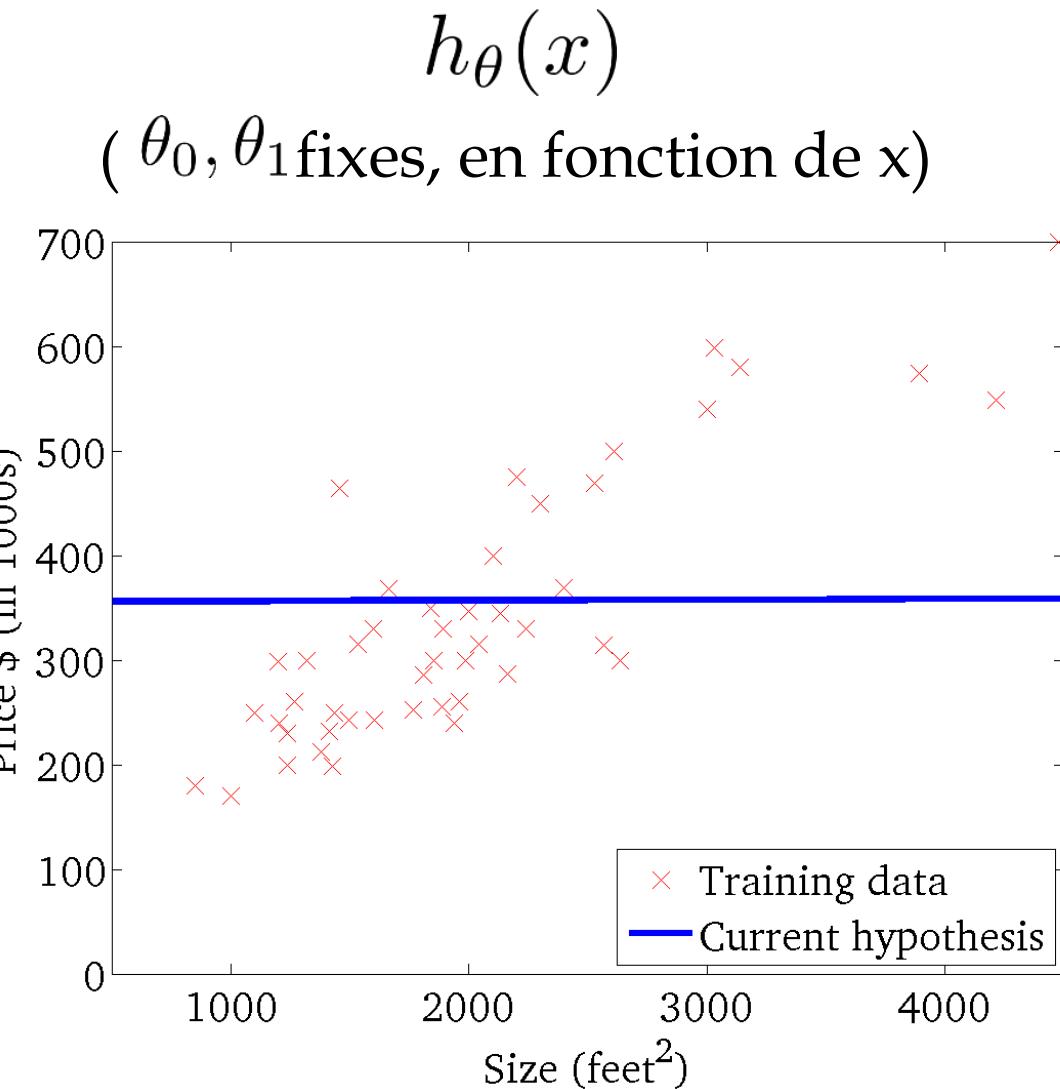
$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)



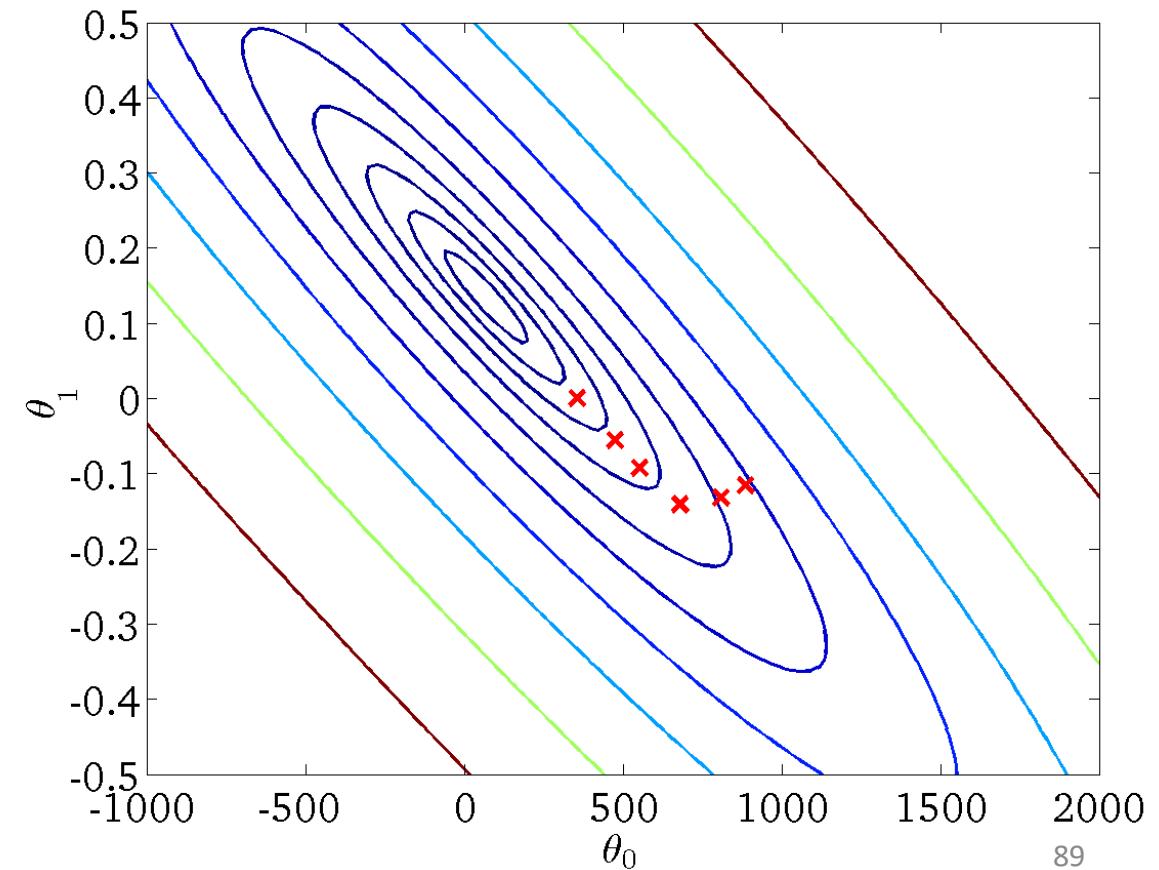
$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)



La décente du gradient

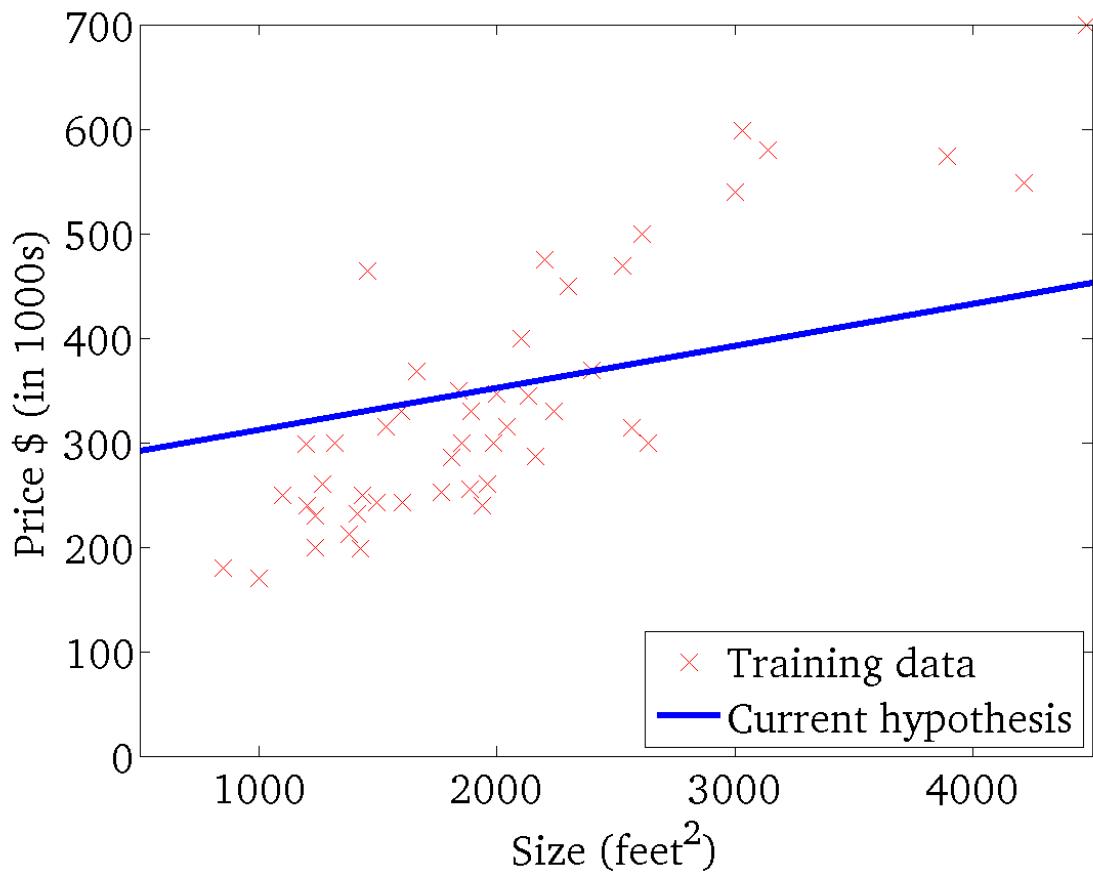


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

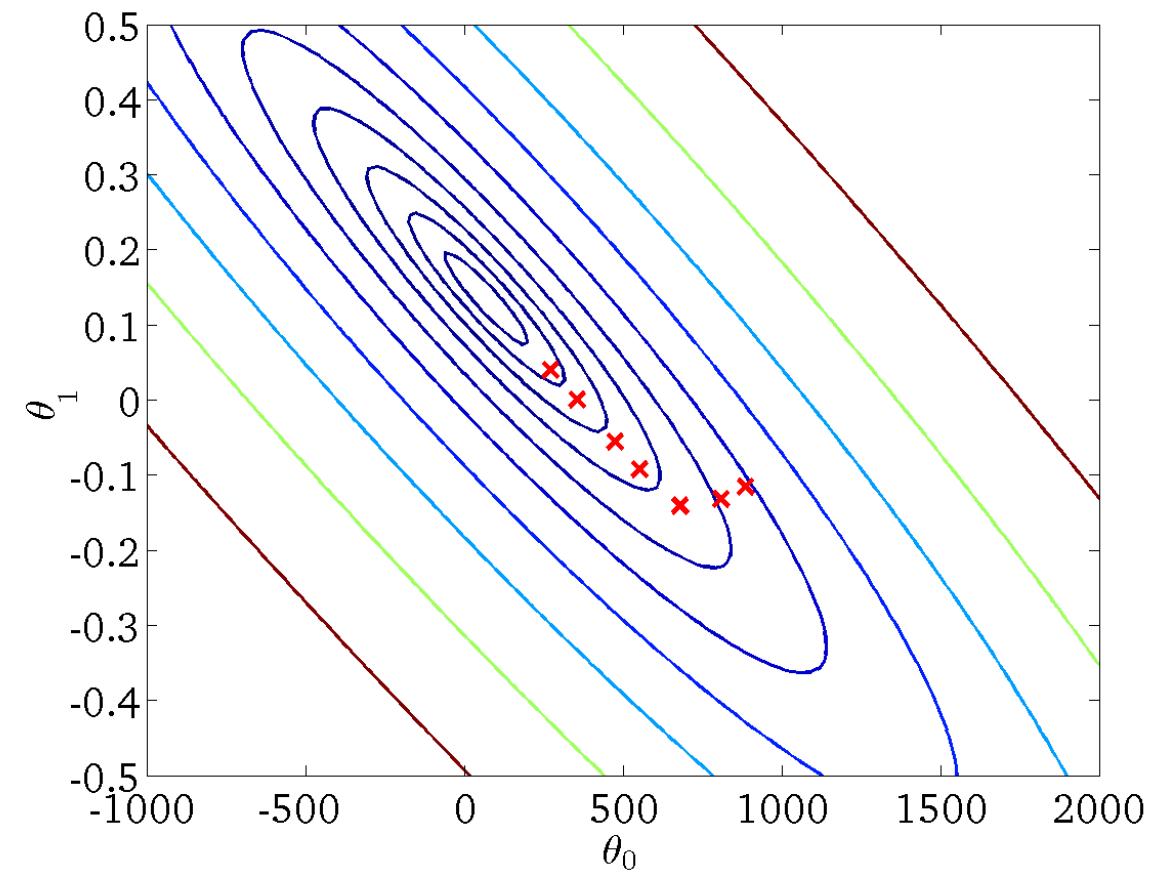


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

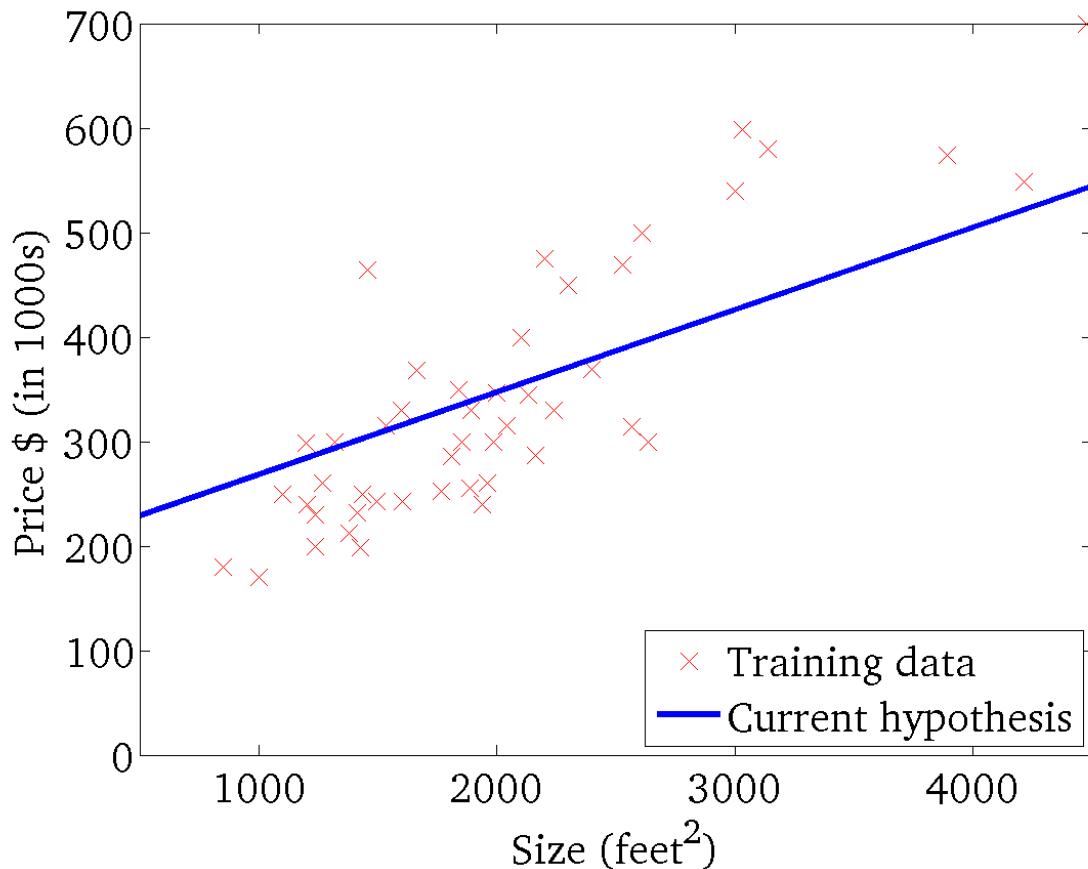


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

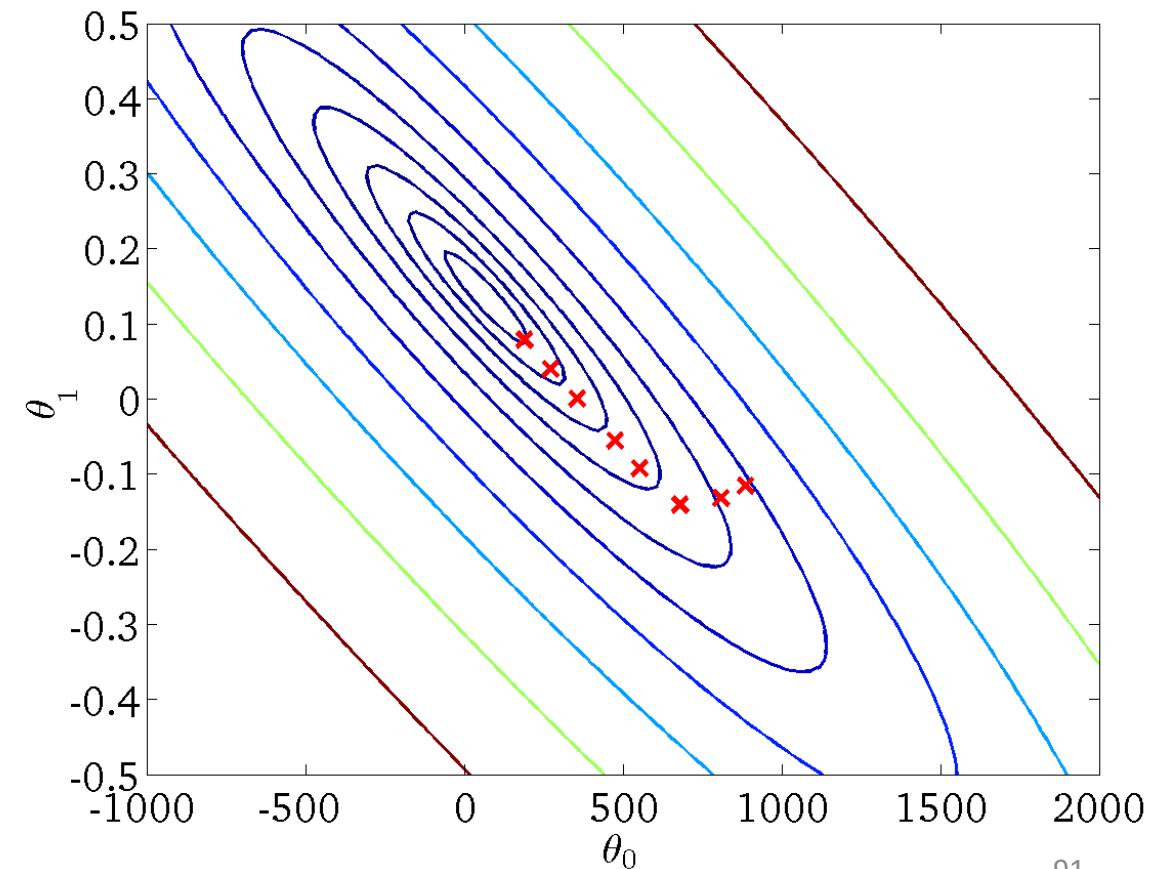


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)

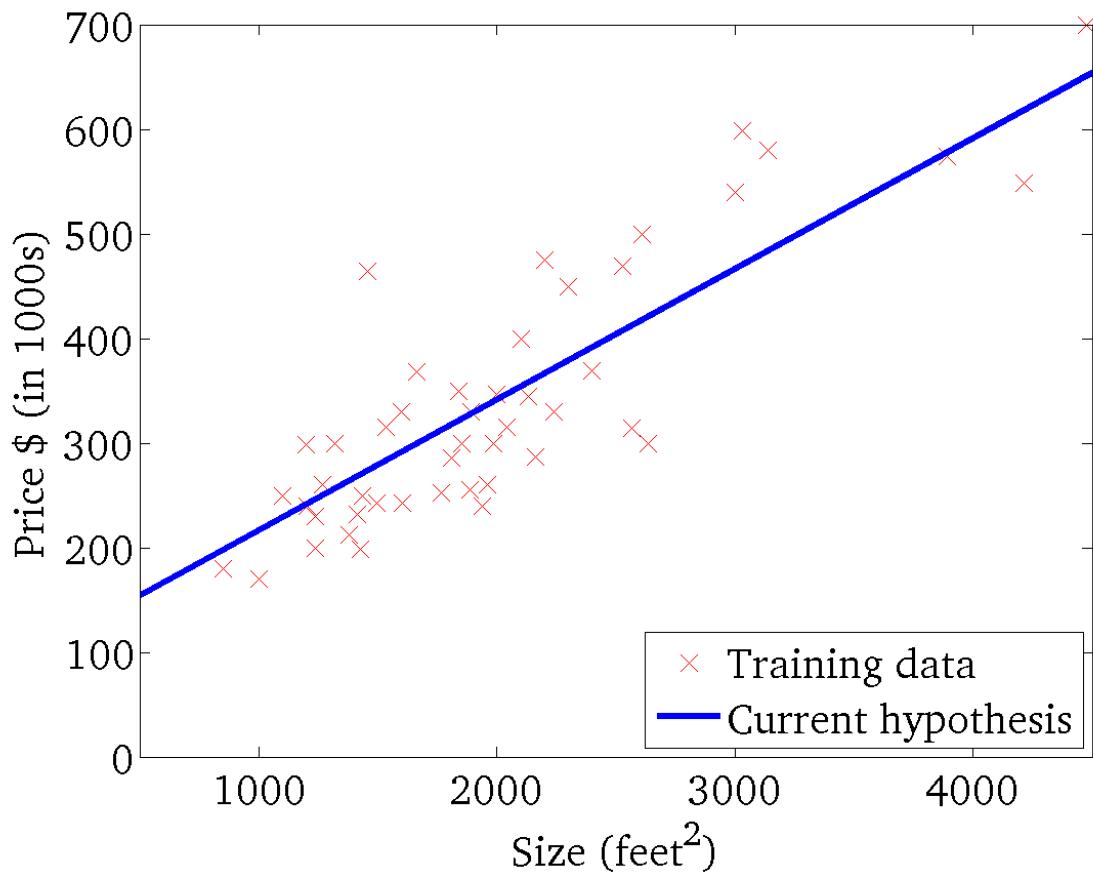


$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)

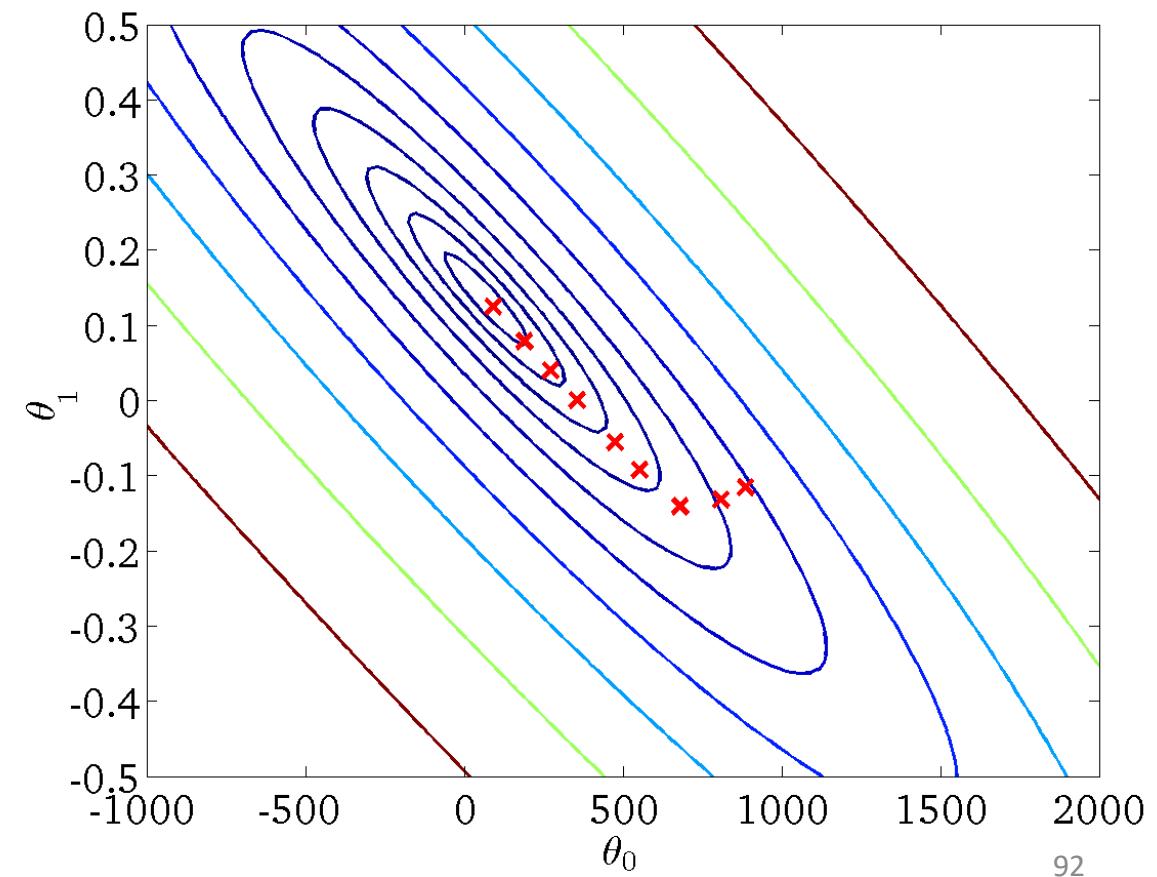


La décente du gradient

$h_{\theta}(x)$
(θ_0, θ_1 fixes, en fonction de x)



$J(\theta_0, \theta_1)$
(en fonction des paramètres θ_0, θ_1)



Variantes de la décente du gradient

- **Batch Gradient Descent** (BGD) utilise les échantillons dans tous les ensembles de données pour mettre à jour le paramètre de poids en fonction de la valeur du gradient au point actuel.

$$w_{k+1} = w_k - \eta \frac{1}{m} \sum_{i=1}^m \nabla f_{w_k}(x^i)$$

- **La descente de gradient stochastique** (SGD) sélectionne de manière aléatoire un échantillon dans un ensemble de données pour mettre à jour le paramètre de pondération en fonction de la valeur du gradient au point actuel.

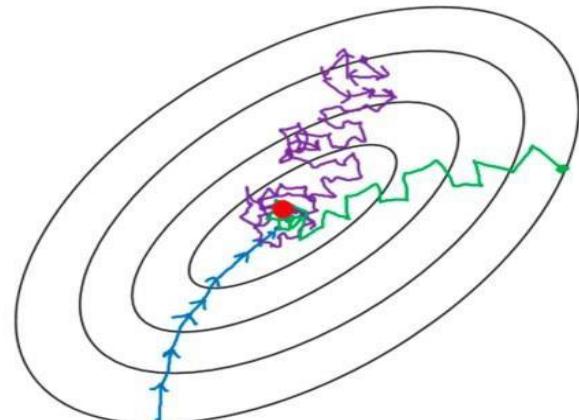
$$w_{k+1} = w_k - \eta \nabla f_{w_k}(x^i)$$

- **Mini-Batch Gradient Descent** (MBGD) combine les fonctionnalités de BGD et SGD et sélectionne les gradients de n échantillons dans un ensemble de données pour mettre à jour le paramètre de poids.

$$w_{k+1} = w_k - \eta \frac{1}{n} \sum_{i=t}^{\tau+n-1} \nabla f_{w_k}(x^i)$$

Comparaison des trois méthodes de descente de gradient

- Dans le SGD, les échantillons sélectionnés pour chaque apprentissage sont stochastiques. Une telle instabilité rend la fonction de perte instable ou même provoque un déplacement inverse lorsque la fonction de perte diminue jusqu'au point le plus bas.
- BGD a la stabilité la plus élevée mais consomme trop de ressources informatiques.
- MBGD est une méthode qui équilibre SGD et BGD.



BGD

Uses **all** training samples for training each time.

SGD

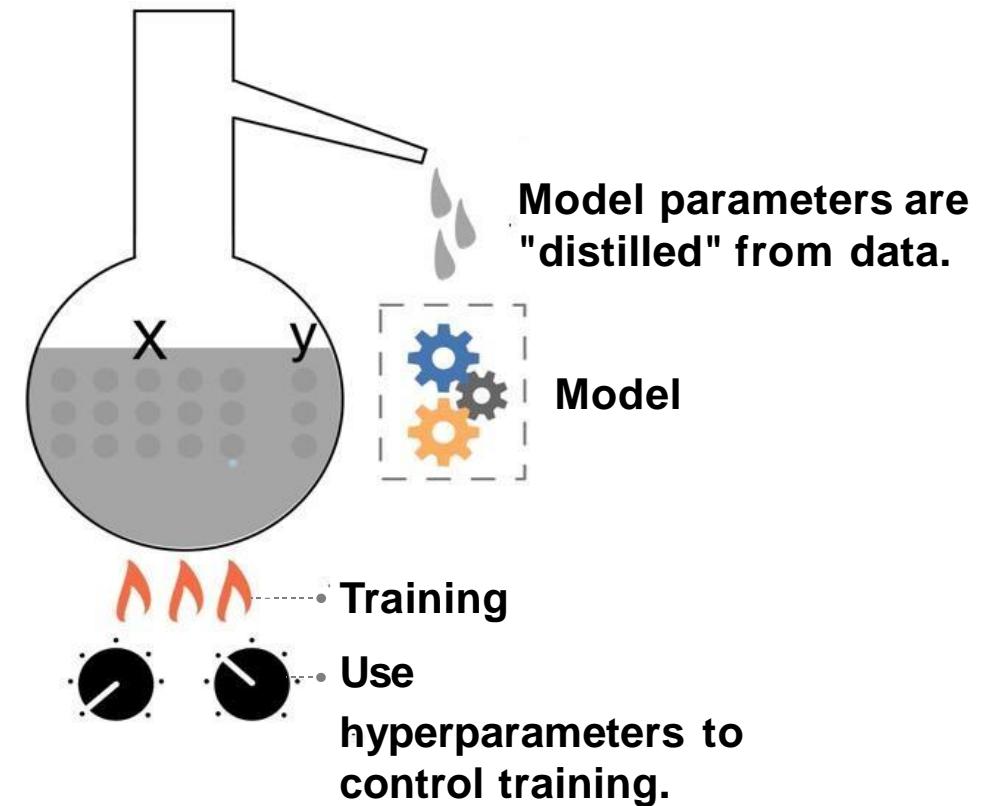
Uses **one** training sample for training each time.

MBGD

Uses a certain number of training samples for training each time.

Paramètres et hyperparamètres dans les modèles

- Le modèle contient non seulement des paramètres mais aussi des hyperparamètres.
- Le but est de permettre au modèle d'apprendre les paramètres optimaux.
 - Les paramètres sont automatiquement appris par les modèles.
 - Les hyperparamètres sont définis manuellement.



Hyperparamètres d'un modèle

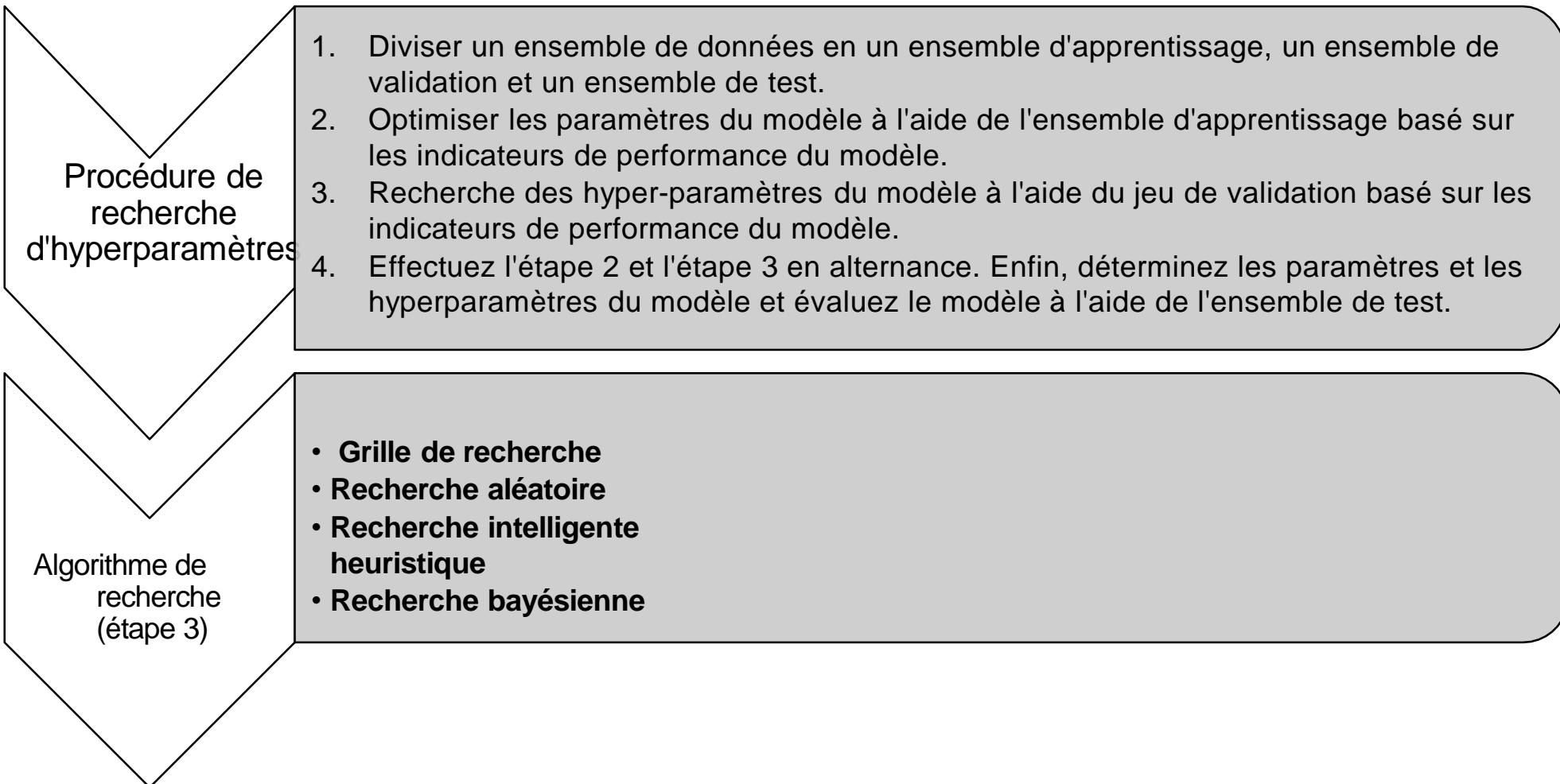
- Souvent utilisé dans les processus d'estimation des paramètres du modèle.
- Souvent spécifié par le praticien.
- Peut souvent être défini à l'aide d'heuristiques.
- Souvent réglé pour un problème de modélisation prédictive donné.

Les hyperparamètres de modèle sont des configurations externes de modèles.

- λ pendant la régression Lasso/Ridge
- Taux d'apprentissage pour l'entraînement d'un réseau de neurones, nombre d'itérations, taille du lot, fonction d'activation et nombre de neurones
- C et dans le vecteur support machines (SVM)
- K dans k plus proche voisin (KNN)
- Nombre d'arbres dans une forêt aléatoire

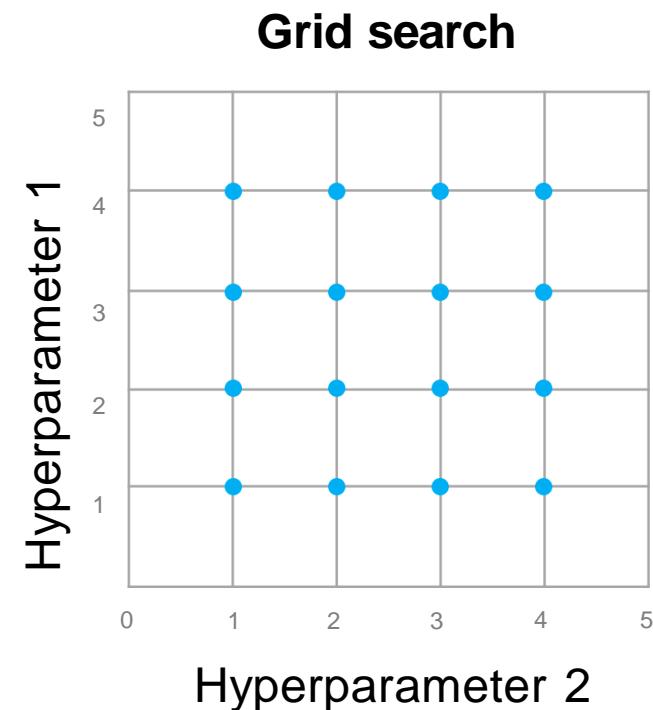
Hyperparamètre de modèle communs

Procédure et méthode de recherche d'hyperparamètres



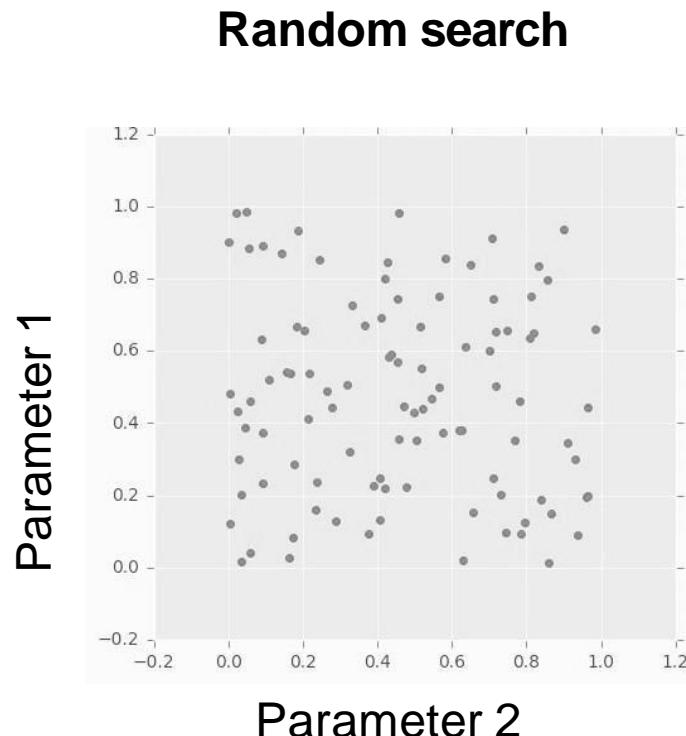
Méthode de recherche d'hyperparamètres - Recherche par grille

- La recherche par grille tente de rechercher de manière exhaustive toutes les combinaisons d'hyperparamètres possibles pour former une grille de valeurs d'hyperparamètres.
- En pratique, la plage de valeurs d'hyperparamètres à rechercher est spécifiée manuellement.
 - Cette méthode fonctionne bien lorsque le nombre d'hyperparamètres est relativement petit. Par conséquent, il est applicable aux algorithmes d'apprentissage automatique en général mais inapplicable aux réseaux de neurones



Méthode de recherche d'hyperparamètres - Recherche aléatoire

- Lorsque l'espace de recherche d'hyperparamètres est grand, la recherche aléatoire est meilleure que la recherche par grille.
- Dans la recherche aléatoire, chaque paramètre est échantillonné à partir de la distribution des valeurs de paramètres possibles, dans le but de trouver le meilleur sous-ensemble d'hyperparamètres.
 - La recherche est effectuée dans une plage grossière, qui sera ensuite réduite en fonction de l'endroit où le meilleur résultat apparaît
 - Certains hyperparamètres sont plus importants que d'autres et l'écart de recherche sera affecté lors de la recherche aléatoire.



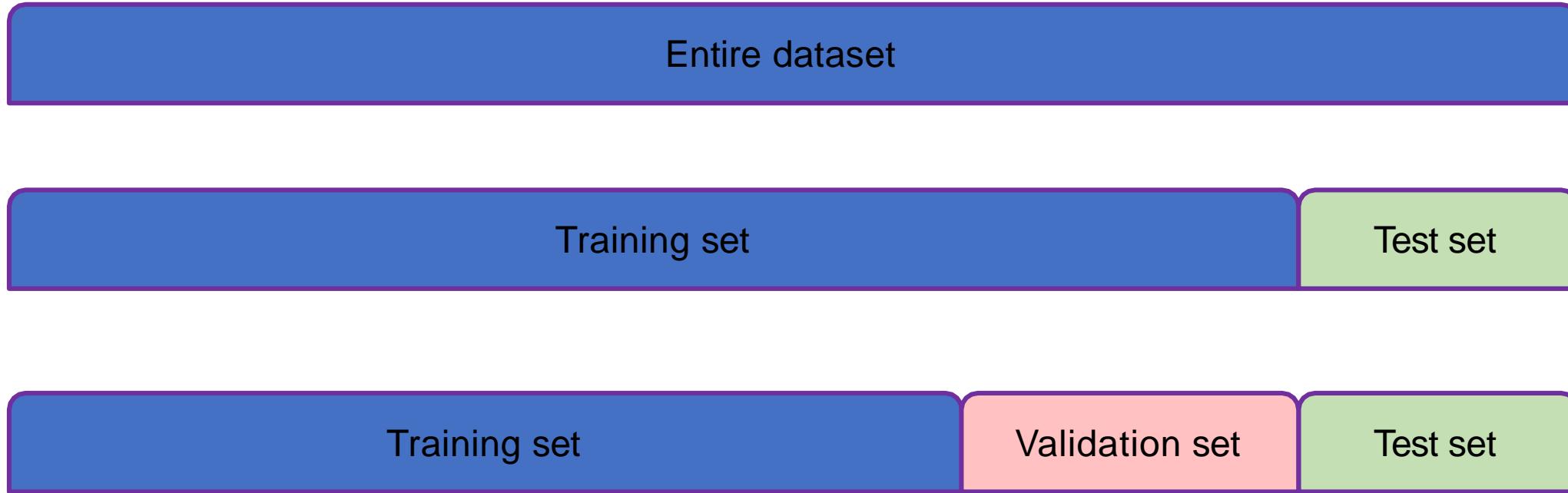
Cross Validation (1)

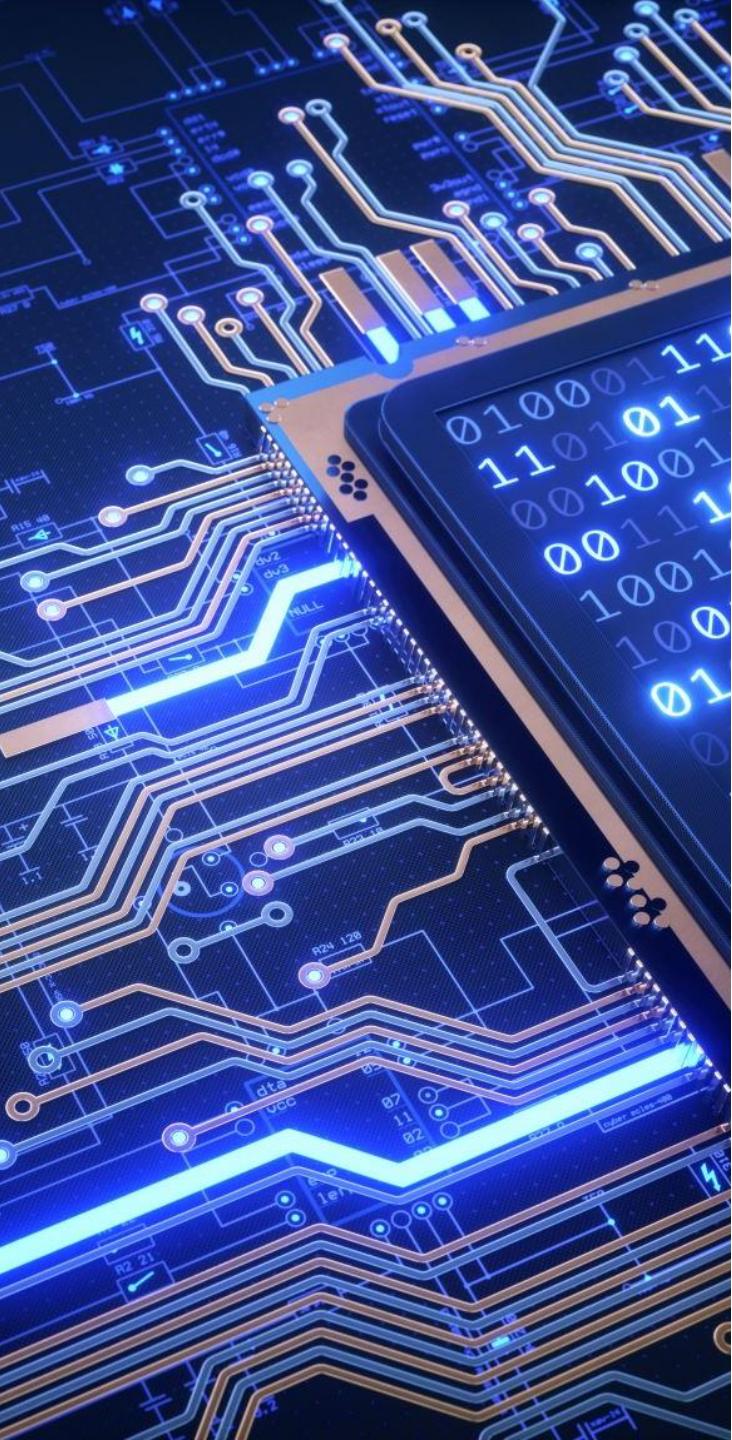
Validation croisée : Il s'agit d'une méthode d'analyse statistique utilisée pour valider les performances d'un classificateur. L'idée de base est de diviser l'ensemble de données d'origine en deux parties : l'ensemble d'apprentissage et l'ensemble de validation. Entraînez le classificateur à l'aide de l'ensemble d'apprentissage et testez le modèle à l'aide de l'ensemble de validation pour vérifier les performances du classificateur.

k-fold cross validation ($K - CV$):

- Divisez les données brutes en k groupes (généralement répartis de manière égale).
- Utilisez chaque sous-ensemble comme ensemble de validation et utilisez les autres sous-ensembles – 1 comme ensemble d'apprentissage. Un total de k modèles peut être obtenu.
- Utilisez la précision de classification moyenne des ensembles de validation finaux des modèles comme indicateur de performance du classificateur $K - CV$.

Cross Validation (2)

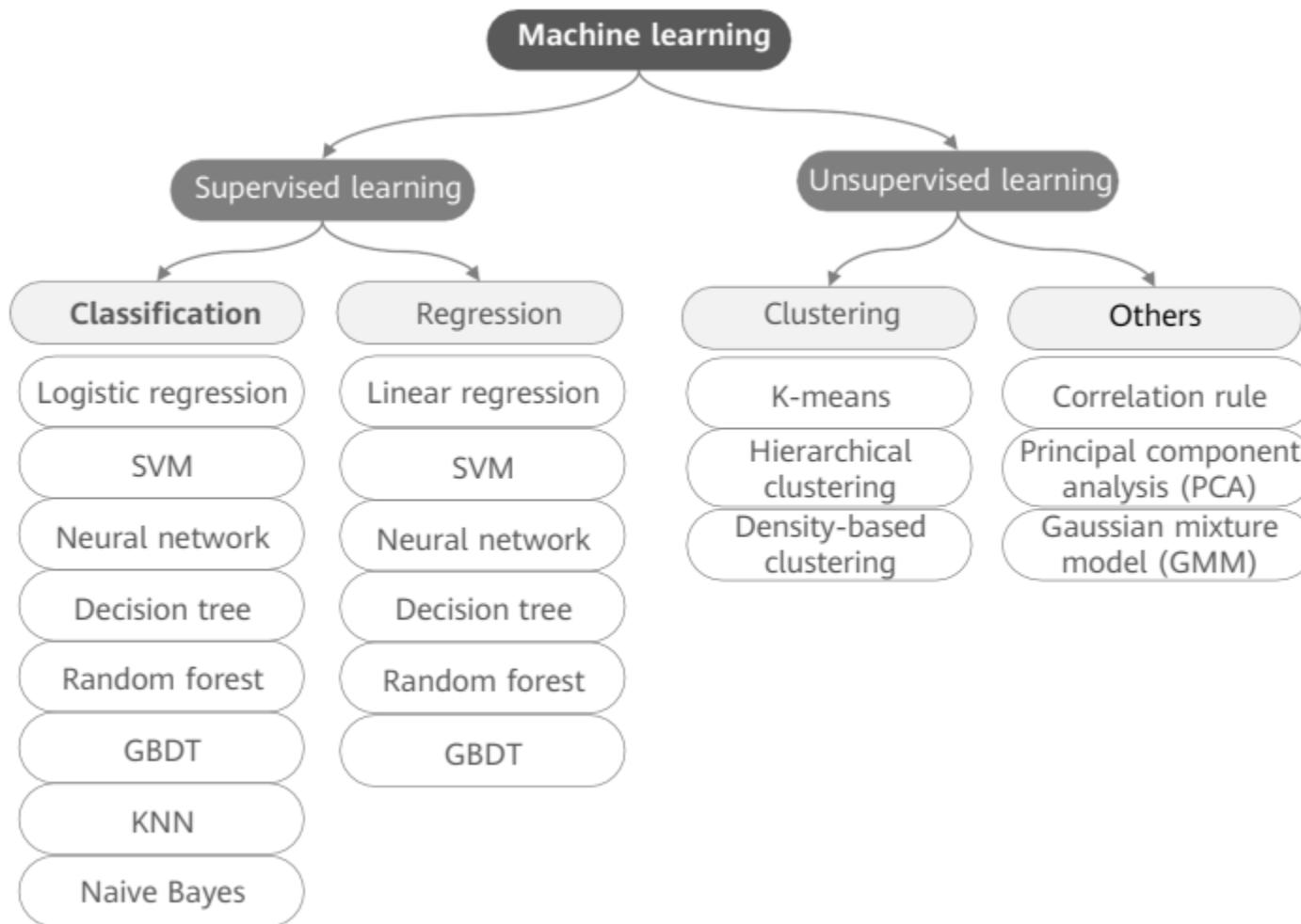




Plan

- Définition de l'apprentissage automatique
- Types d'apprentissage automatique
- Processus d'apprentissage automatique
- Autres méthodes clés d'apprentissage automatique
- **Algorithmes courants d'apprentissage automatique**

Présentation de l'algorithme d'apprentissage automatique



Formulation

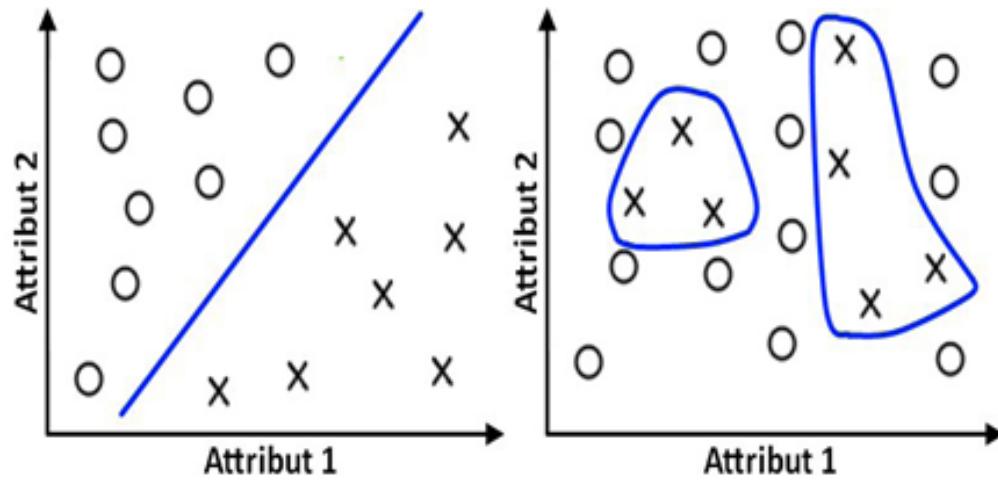
- On appelle fonction d'apprentissage la fonction notée : $I : X \rightarrow Y$ qui associe un résultat (valeur) supervisé à chaque vecteur d'entrée.
- Le but d'un algorithme d'apprentissage supervisé sera donc d'approcher cette fonction I , uniquement à partir des exemples d'apprentissage.
- En fonction du résultat (comportement) supervisé que l'on veut obtenir, on peut distinguer deux types problèmes :
 - **Régression** : lorsque le résultat supervisé que l'on cherche à estimer est une valeur dans un ensemble continu de réels.
 - **Classification** : lorsque l'ensemble des valeurs de sortie est discret. Ceci revient à attribuer une classe (aussi appelée étiquette ou label) pour chaque vecteur d'entrée.
- Nous nous plaçons souvent dans le cas de problème classification à deux classes (2-classe) qui peut être facilement étendu à N-classe.

Méthodes de classification supervisée

- Les méthodes de classification supervisée peuvent être basées sur
 - des **hypothèses probabilistes** (cas du classifieur naïf bayésien),
 - des **notions de proximité** (exemple, k plus proches voisins) ou
 - des **recherches dans des espaces d'hypothèses** (exemple, arbres de décisions).

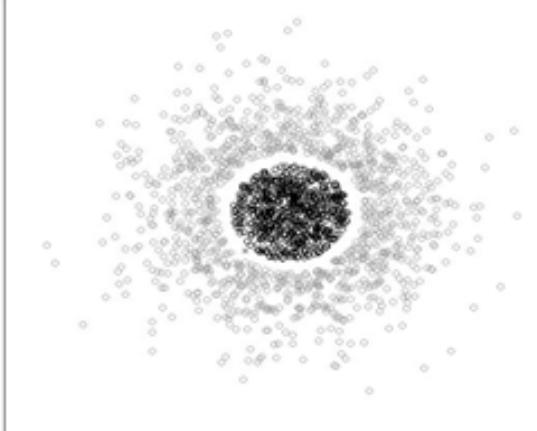
Problème Linéaire et Non-Linéaire

- On dit qu'un problème est **linéairement séparable** si les exemples de classes différentes sont complètement séparables par un hyperplan
- Un problème peut être **non séparable de manière linéaire**. Dans ce cas, il faut utiliser d'autres types de classifieurs, souvent plus longs à paramétrier, mais qui obtiennent des résultats plus précis.

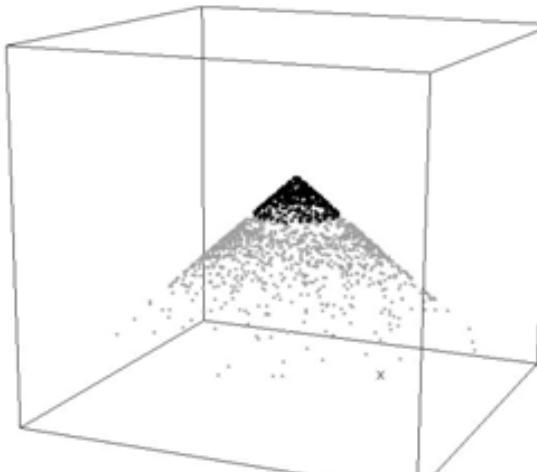


Problème Linéaire et Non-Linéaire

- Un problème, initialement, non linéairement séparable peut s'avérer séparable avec l'ajout d'un nouvel attribut.
- D'où l'intérêt d'un choix judicieux de ces attributs.
- C'est ce principe qui est utilisé par le classifieur Support Vector Machine (SVM)



Problème non linéairement séparable
(2 attributs)



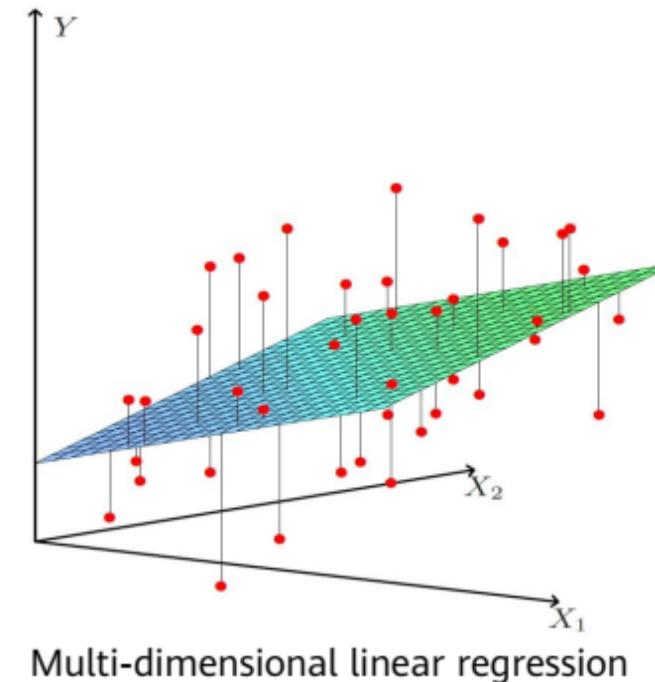
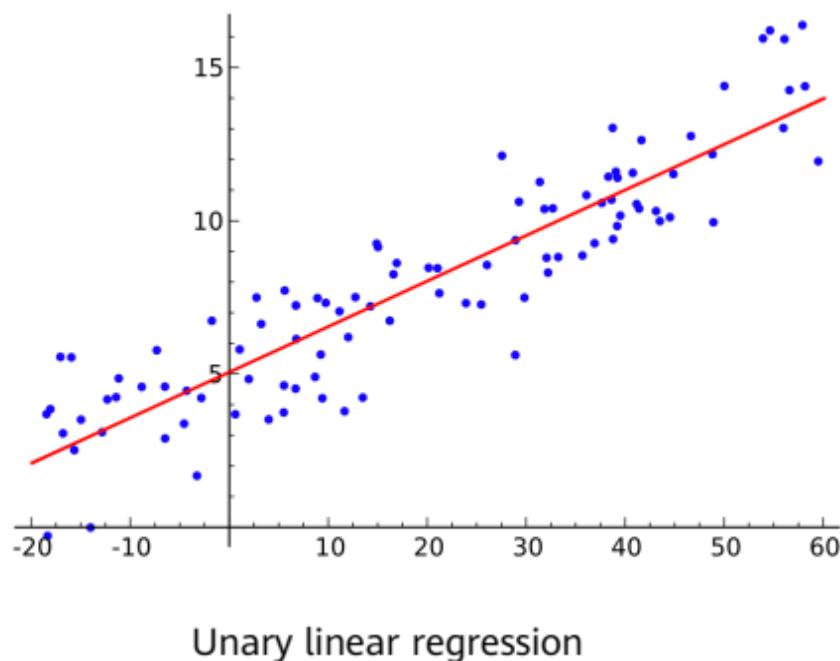
Même problème, linéairement séparable
(un attribut supplémentaire est considéré)

Classificateurs à mémoire

- L'intérêt de ces classificateurs est qu'ils ne nécessitent aucune phase d'apprentissage ou d'entraînement
- Ils permettent de déduire directement la classe d'un nouvel exemple à partir de l'ensemble d'apprentissage.

Régression linéaire (1)

- Une méthode d'analyse statistique pour déterminer les relations quantitatives entre deux ou plusieurs variables par l'analyse de régression par les statistiques.



Régression linéaire (2)

- La fonction de modèle de la régression linéaire est la suivante, w indique le paramètre de pondération, b indique le biais et x indique l'attribut de l'échantillon.

$$h_w(x) = w^T x + b$$

- La relation entre la valeur prédite par le modèle et la valeur réelle est la suivante, y indique la valeur réelle et ε indique l'erreur.

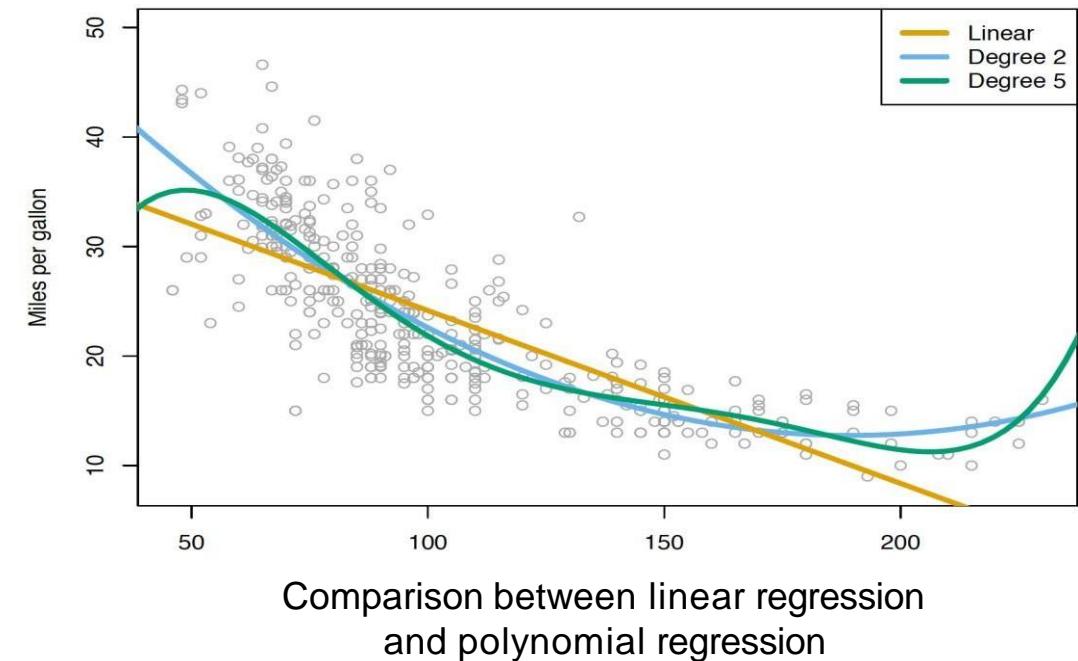
$$y = w^T x + b + \varepsilon$$

- L'erreur ε est influencée par de nombreux facteurs indépendamment. Selon la fonction de distribution normale et l'estimation du maximum de vraisemblance (*maximum likelihood estimation*) la fonction de perte de la régression linéaire est la suivante :

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2$$

Extension de régression linéaire - Régression polynomiale

- Généralement, la complexité d'un ensemble de données dépasse la possibilité d'ajustement par une ligne droite. C'est-à-dire qu'un sous-ajustement évident se produit si le modèle de régression linéaire d'origine est utilisé. La solution est d'utiliser la régression polynomiale.



$$h_w(x) = w_1x + w_2x^2 + \dots + w_nx^n + b$$

Régression linéaire et prévention du surapprentissage

- Des termes de régularisation peuvent être utilisés pour réduire le surapprentissage. La valeur de λ ne peut pas être trop grande ou trop petite dans l'espace échantillon. Vous pouvez ajouter une perte de somme carrée sur la fonction cible.

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2 + \lambda \sum \|w\|_2^2$$

- Termes de régularisation (norme) : Le terme de régularisation est ici appelé norme L2. La régression linéaire qui utilise cette fonction de perte est également appelée Ridge regression.

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2 + \lambda \sum \|w\|_1$$

- La régression linéaire avec perte absolue est appelée régression de Lasso.

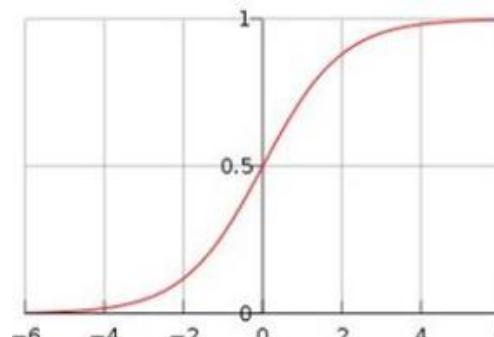
Régression logistique

- Le modèle de régression logistique est utilisé pour résoudre les problèmes de classification. Le modèle est défini comme suit :

$$P(Y = 1|x) = \frac{e^{wx+b}}{1 + e^{wx+b}}$$

$$P(Y = 0|x) = \frac{1}{1 + e^{wx+b}}$$

- où w indique le poids, b indique le biais et $wx + b$ est considéré comme la fonction linéaire de x . Comparez les deux valeurs de probabilité précédentes. La classe avec une valeur de probabilité plus élevée est la classe de x .



Régression logistique

- Le modèle de régression logistique et le modèle de régression linéaire sont tous deux des modèles linéaires généralisés. La régression logistique introduit des facteurs non linéaires (la fonction sigmoïde) basés sur la régression linéaire et définit des seuils, afin de pouvoir traiter les problèmes de classification binaire.
- Selon la fonction modèle de la régression logistique, la fonction de perte de la régression logistique peut être estimée comme suit en utilisant l'estimation du maximum de vraisemblance :

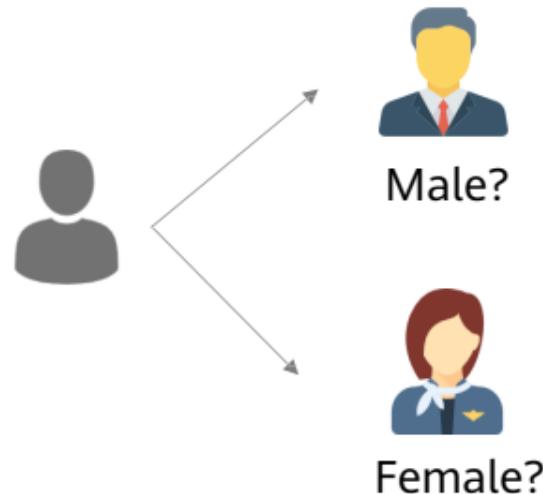
$$J(w) = -\frac{1}{m} \sum (y \ln h_w(x) + (1-y) \ln(1-h_w(x)))$$

- où w indique le paramètre de poids, m indique le nombre d'échantillons, x indique l'échantillon et y indique la valeur réelle. Les valeurs de tous les paramètres de poids peuvent également être obtenues grâce à l'algorithme de descente de gradient.

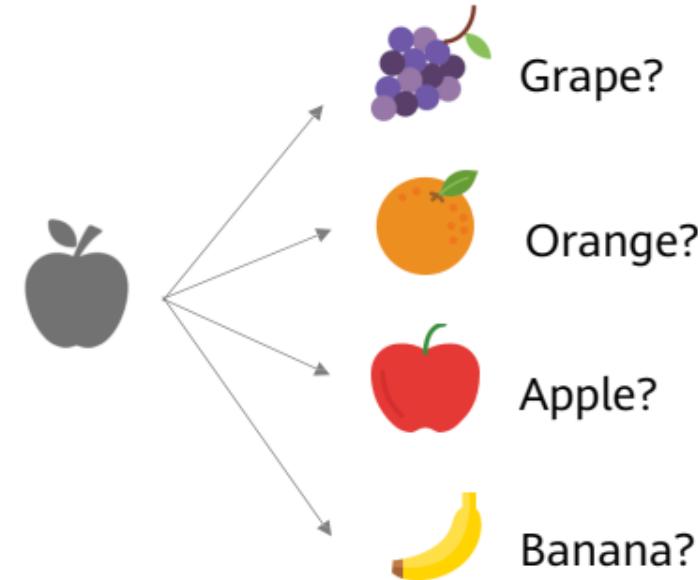
Extension de régression logistique - Fonction Softmax

- La régression logistique s'applique principalement aux problèmes de classification binaire. Pour les problèmes de classification multi-classes, utilisez la fonction **Softmax**.

Binary classification problem



Multi-class classification problem



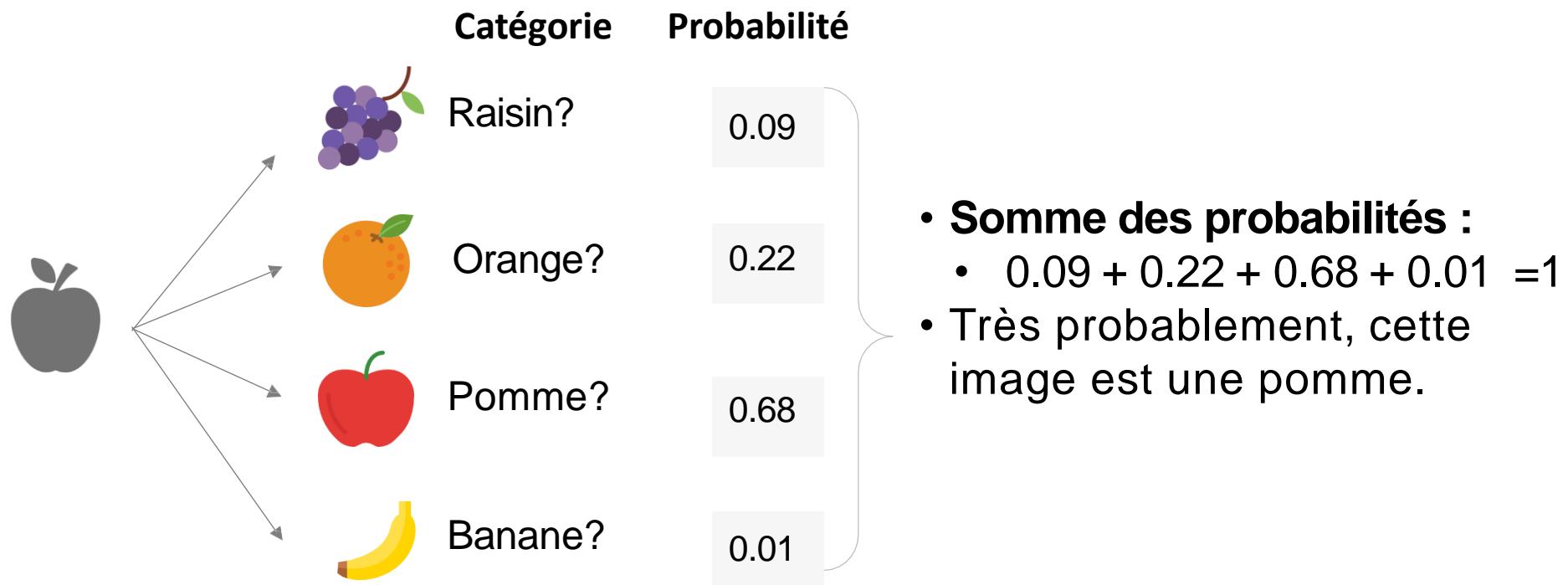
Extension de régression logistique - Fonction Softmax

- La fonction Softmax est utilisée pour mapper un vecteur K-dimensions de valeurs réelles arbitraires à un autre vecteur K-dimensions de valeurs réelles, où chaque élément est dans l'intervalle (0, 1).
- La fonction de probabilité de régression de Softmax est la suivante :

$$p(y = k \mid x; w) = \frac{e^{w_k^T x}}{\sum_{l=1}^K e^{w_l^T x}}, k = 1, 2, \dots, K$$

Extension de régression logistique - Fonction Softmax

- Softmax attribue une probabilité à chaque classe dans un problème multi-classes. Ces probabilités doivent totaliser 1.

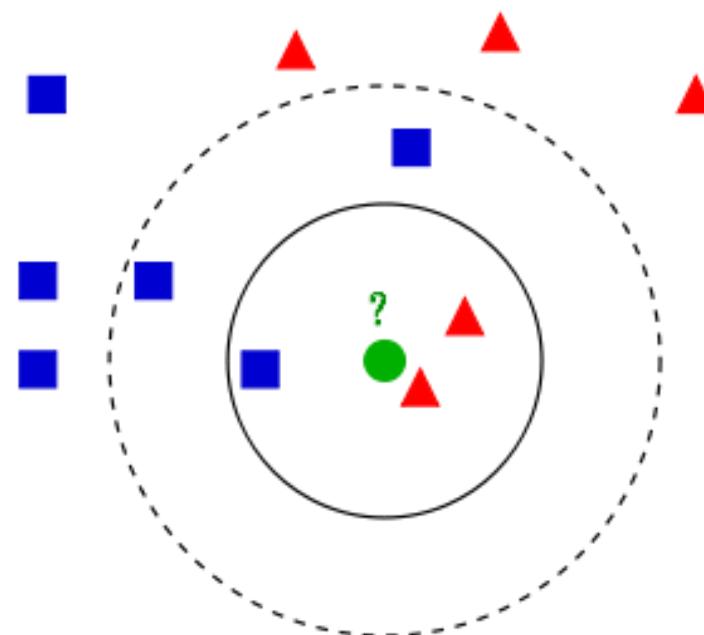


K-plus proches Voisins

- Le classifieur des k plus proches voisins ou **k -ppv** (k -Nearest Neighbor ou **k -NN**, en anglais) est l'un des algorithmes de classification les plus simples.
- **Principe** : Un exemple est classifié par vote majoritaire de ses k "voisins" (par mesure de distance),
 - c'est-à-dire qu'il est prédit de classe C si la classe la plus représentée parmi ses k voisins est la classe C.
 - L'opérateur de distance le plus souvent utilisé est la distance Euclidienne.
 - Un cas particulier est le cas où $k = 1$, l'exemple est alors affecté à la classe de son plus proche voisin.

K-plus proches Voisins

- Le choix du k est très important pour la classification.
- On s'abstient de choisir des valeurs paires de k, pour éviter les cas d'égalité.



Pseudo-Algorithme K-ppv

Déclarations

- M : nombre de classes d'apprentissage $C = \{c_1, \dots, c_M\}$;
- N : nombre d'exemples d'entraînement $E = \{e_1, \dots, e_N\}$;
- $Ent = \{(e_i, c_k)\}$: ensemble d'apprentissage formé par les couples (e_i, c_k) ; /* e_i est l'exemple d'apprentissage et c_k sa classe d'appartenance */
- ex : exemple test /*dont on cherche la classe d'appartenance*/

Début

< On cherche à classer e_x ? > ;

Pour Chaque exemple $(e_i, w) \in Ent$ **Faire**

< Calculer la distance $D(e_i, e_x)$ entre e_i et e_x > ;

FPour

< Trier les échantillons e_i par ordre croissant des distances > ;

Pour les k plus proches e_i de e_x (les k premières – ayant les plus petites- $D(e_i, e_x)$) **Faire**

< Compter le nombre d'occurrences de chaque classe > ;

FPour

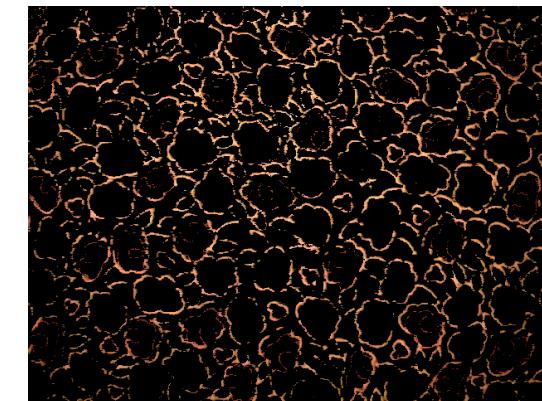
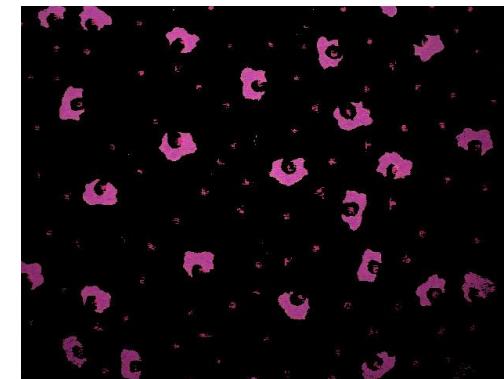
< Attribuer à e_x la classe c_j la plus fréquente > ; /*Celle qui apparaît le plus souvent*/

Fin.

Segmentation d'image par k-ppv

- On suppose qu'on dispose d'une base d'exemple étiquetés (dans ce cas un ensemble de pixels dont on connaît leurs appartennances aux régions (classes) dans l'image)
- Pour un nouveau pixel x_i , prédire sa classe d'appartenance y_i
- Pour chaque pixel x_i à classer :
 - Calculer les distances avec les pixels des régions étiquetées : $\{d_j\}$
 - Si $k = 1 \Rightarrow$ la classe y_i de x_i est celle de l'exemple le plus proche de x_i
 - si $k > 1 \Rightarrow$ la classe y_i de x_i est la classe majoritaire des k exemples les plus proches au sens de la distance choisie

Segmentation d'image par k-ppv



Classifieur Naïf Bayésien

- La **classification naïve bayésienne** repose sur l'hypothèse que les attributs sont fortement (ou naïvement) **indépendants**.
- Elle est basée sur le théorème de Bayes qui ne s'applique que sous cette hypothèse.
- Etant donné un objet O, la méthode consiste à calculer la probabilité d'appartenance de O à chaque classe, puis choisir celle qui maximise cette valeur

Théorème de Bayes

- Soit l'ensemble d'apprentissage D , **la probabilité a posteriori** de l'hypothèse h , $P(h|D)$ suit le théorème de Bayes :

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- MAP (maximum posteriori) hypothesis

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{Max}} P(h \mid D) = \underset{h \in H}{\operatorname{Max}} P(D \mid h)P(h).$$

- Difficulté pratique: on a besoin de connaître initialement plusieurs probabilités et un temps de calcul non négligeable
- On suppose que les attributs sont indépendants:

$$P(c_j \mid V) \approx P(c_j) \prod_{i=1}^n P(v_i \mid c_j)$$

Exemple

- Etant donné des données d'entraînement, on peut calculer les probabilités.
P:jouer au tennis et N: ne pas jouer au tennis

Temps	Temperature	Humidite	Vent	Class				
soleil	chaud	élevé	faux	N	temps	P	N	
soleil	chaud	élevé	VRAI	N	soleil	2/9	3/5	humidité
couvert	chaud	élevé	faux	P	couvert	4/9	0	P
pluie	tiede	élevé	faux	P	pluie	3/9	2/5	N
pluie	froid	normal	faux	P				
pluie	froid	normal		N	température			
couvert	froid	normal	VRAI	P	chaud	2/9	2/5	vent
soleil	tiede	élevé	faux	N	tiède	4/9	2/5	VRAI
soleil	froid	normal	faux	P	froid	3/9	1/5	FAUX
pluie	tiede	normal	faux	P				
soleil	tiede	normal	VRAI	P				
couvert	tiede	élevé	VRAI	P				
couvert	chaud	normal	faux	P				
pluie	tiede	élevé	VRAI	N				

Exemple

- Le problème de classification peut être formalisé en utilisant les probabilités a-posteriori:
 - $P(C|X)$ = prob. que $X=\langle x_1, \dots, x_k \rangle$ soit de la classe C.
 - Ex. $P(\text{classe}=N \mid \text{temps}=soleil, \text{vent}=vrai, \dots)$
 - Affecter à X la classe C tel que $P(C|X)$ est maximal
 - Hypothèse: indépendance des attributs
 - $P(x_1, \dots, x_k | C) = P(x_1 | C) \cdot \dots \cdot P(x_k | C)$

Exemple estimer $P(x_i | C)$

Temps	Temperature	Humidite	Vent	Class
soleil	chaud	élevé	faux	N
soleil	chaud	élevé	VRAI	N
couvert	chaud	élevé	faux	P
pluie	tiede	élevé	faux	P
pluie	froid	normal	faux	P
pluie	froid	normal		N
couvert	froid	normal	VRAI	P
soleil	tiede	élevé	faux	N
soleil	froid	normal	faux	P
pluie	tiede	normal	faux	P
soleil	tiede	normal	VRAI	P
couvert	tiede	élevé	VRAI	P
couvert	chaud	normal	faux	P
pluie	tiede	élevé	VRAI	N

$$P(p) = 9/14$$

$$P(n) = 5/14$$

X = <pluie, chaud, élevée, faux>

Temps	
$P(\text{soleil} P) = 2/9$	$P(\text{soleil} N) = 3/5$
$P(\text{couvert} P) = 4/9$	$P(\text{couvert} N) = 0$
$P(\text{pluie} P) = 3/9$	$P(\text{pluie} N) = 2/5$
Température	
$P(\text{chaud} P) = 2/9$	$P(\text{chaud} N) = 2/5$
$P(\text{tiède} P) = 4/9$	$P(\text{tiède} N) = 2/5$
$P(\text{froid} P) = 3/9$	$P(\text{froid} N) = 1/5$
Humidité	
$P(\text{élevée} P) = 3/9$	$P(\text{élevée} N) = 4/5$
$P(\text{normale} P) = 6/9$	$P(\text{normale} N) = 2/5$
Vent	
$P(\text{Vrai} P) = 3/9$	$P(\text{vrai} N) = 3/5$
$P(\text{faux} P) = 6/9$	$P(\text{faux} N) = 2/5$

Exemple : test

- Soit $X = \langle \text{pluie}, \text{chaud}, \text{élevée}, \text{faux} \rangle$
- $P(X|p) \cdot P(p) =$
 $P(\text{pluie}|p) \cdot P(\text{chaud}|p) \cdot P(\text{élevée}|p) \cdot P(\text{faux}|p) \cdot P(p) = 3/9 \cdot 2/9 \cdot 3/9 \cdot 6/9 \cdot 9/14 = 0.010582$
- $P(X|n) \cdot P(n) =$
 $P(\text{pluie}|n) \cdot P(\text{chaud}|n) \cdot P(\text{élevée}|n) \cdot P(\text{faux}|n) \cdot P(n) = 2/5 \cdot 2/5 \cdot 4/5 \cdot 2/5 \cdot 5/14 = \mathbf{0.018286}$
- X est classifié en N (ne pas jouer au tennis)

Exercice

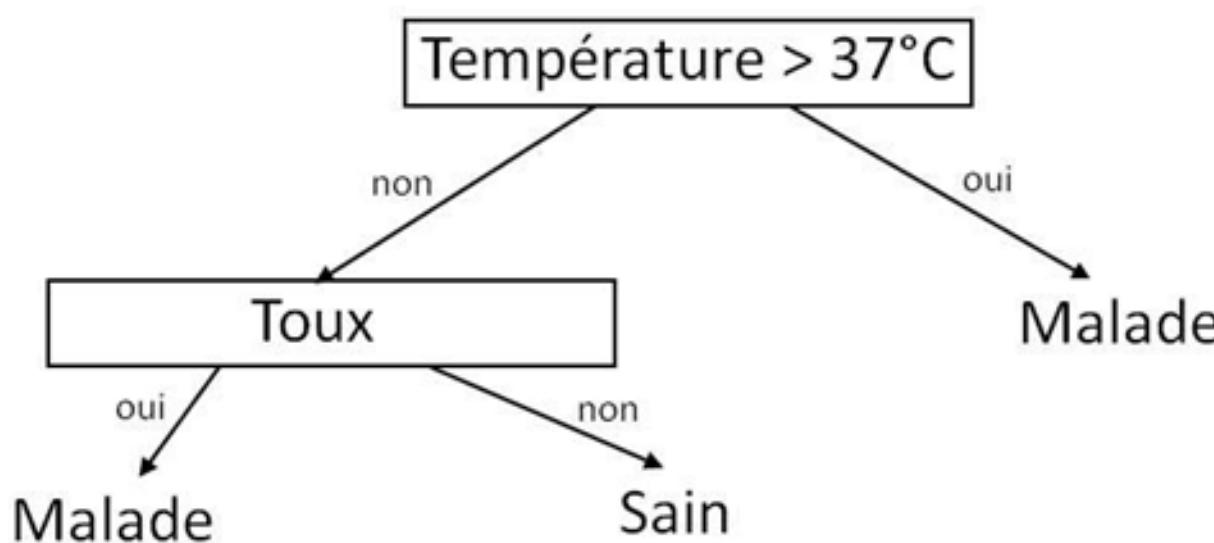
Fievre	Douleur	Toux	Maladie
oui	Abdomen	non	Appendicite
non	Abdomen	oui	Appendicite
oui	gorge	non	rhume
oui	gorge	oui	rhume
non	gorge	oui	mal de gorge
oui	non	non	aucune
oui	non	oui	rhume
non	non	oui	refroidissement
non	non	non	aucune

Arbres de Décision

- Les **arbres de décision** sont un outil très populaire de classification. Leur principe repose sur la construction d'un arbre de taille limitée.
- La racine constitue le point de départ de l'arbre et représente l'ensemble des données d'apprentissage. Puis ces données sont segmentées en plusieurs sous-groupes, en fonction d'une variable **discriminante** (un des attributs).
- Une fois l'arbre construit à partir des données d'apprentissage, on peut prédire un nouveau cas en le faisant descendre le long de l'arbre, jusqu'à une feuille.
- Comme la feuille correspond à une classe, l'exemple sera prédict comme faisant partie de cette classe.

Exemple

- Une personne qui a une température $< 37^{\circ}\text{C}$ et qui a de la toux est prédite comme malade, tandis qu'une personne qui a une température $< 37^{\circ}\text{C}$ mais pas de toux est considérée comme saine.



Construction de l'arbre

- Lors de la création de l'arbre, la première question qui vient à l'esprit est le choix de la variable de segmentation sur un sommet.
- Pourquoi par exemple avons-nous choisi la variable "température" à la racine de l'arbre ?
- Il nous faut donc une mesure afin d'évaluer la qualité d'une segmentation et sélectionner la meilleure variable sur chaque sommet.
- Ces algorithmes s'appuient notamment sur les techniques issues de la théorie de l'information, et notamment la théorie de Shannon

Idée et Propriétés Générales

- Diviser récursivement et le plus efficacement possible les individus de l'ensemble d'apprentissage par des tests définis à l'aide des variables jusqu'à ce que l'on obtienne des sous-ensembles d'individus ne contenant (presque) que des exemples appartenant à une même classe
- A la base, trois opérations seront nécessaires :
 - **Décider si un nœud est terminal** (tous les individus sont dans la même classe).
 - **Sélectionner un test à associer à un nœud** (utiliser critères statistiques).
 - **Affecter une classe à une feuille** (nœud terminal)- la classe majoritaire

Classification et Règles

- L'Arbre de Décision (AD) permet de classer un nouvel exemple : (37.2, oui), c'est-à-dire, Température=37.2 et Gorge-Irritéeoui, comme appartenant à la classe malade.
- L'AD peut être traduit en un système de règles ; lesquelles pouvant être considérées comme le pseudo-code ou l'algorithme de l'AD :
 - Si (Temp. < 37.5) et (Gorge-irritée) Alors malade.
 - Si (Temp. < 37.5) et Non (Gorge-irritée) Alors Sain.
 - Si (Temp. > 37.5) Alors malade.

Inférence d'Arbres de Décision

- Objectif : Inférer (déduire et aussi au sens de construire) un arbre de décision à partir d'exemples.
- Pour ce faire, on a besoin :
 - de comprendre la répartition de la population (ex., de patients) dans l'arbre. Ainsi, il est intéressant de savoir mesurer le degré de mélange d'une population.
 - de la définition d'une méthode d'inférence, en saisissant :
 - Comment sélectionner le test à effectuer à un nœud ?
 - Comment décider si un nœud est terminal ?
 - Quelle classe associée à une feuille ?
 - Enfin, de comment tout écrire mathématiquement ?

Mélange et Degré de Mélange

- Le calcul du degré de mélange des classes dans la population vient du besoin de comparer les différents choix possibles.
- Ainsi, de ce besoin, on introduit des fonctions qui permettent de mesurer le degré de mélange d'une population dans les différentes classes.
- Les propriétés de ces fonctions devraient être de la sorte :
 - Le minimum est atteint lorsque tous les nœuds sont « purs » (si tous les individus associés au nœud appartiennent à la même classe). Ainsi, le mélange sera minimal (sinon nul).
 - Le maximum est atteint lorsque les individus sont équirépartis entre les classes (mélange maximal).

Exemples de Fonctions Mélanges

- **Fonction d'Entropie**

$$Entropie(p) = - \sum_{k=1}^c P(k|p) \ln P(k|p)$$

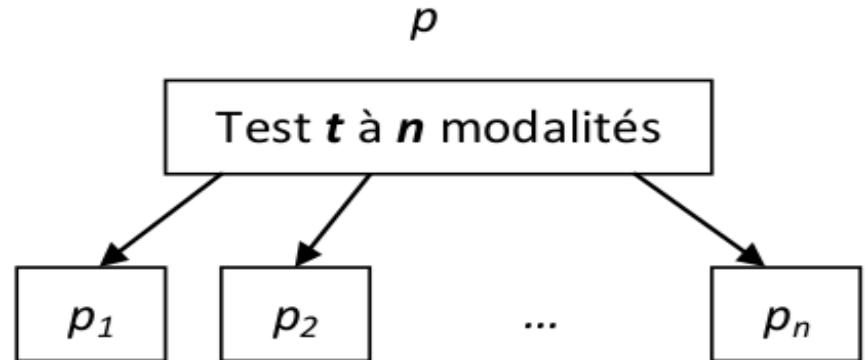
Avec, classe k et nœud p .

- **Fonction de Gini**

$$Gini(p) = 1 - \sum_{k=1}^c P^2(k|p) = 2 \sum_{k < k'} P(k|p)P(k'|p)$$

Notion de Gain

- t : le test (la variable).
- n : le nombre de modalités de t .
- i : la fonction pour mesurer le degré de mélange.



On introduit la fonction de gain :

$$Gain(p, t) = i(p) - \sum_{j=1}^n p_j i(p_j)$$

Avec,

- p_j : la proportion des individus de la position (nœud) p qui vont en position p_j .
- La position p est fixée !
- Le but est de **chercher le test qui maximise le gain** !

Algorithme de Construction d'un Arbre de Décision

Cet algorithme d'apprentissage et de classification correspond au classifieur CART (Classification And Regression Tree).

- **Entrée**

- n individus,
- p variables continues ou discrètes,
- Une variable supplémentaire contenant la classe de chaque individu (c classes).

- **Sortie**

- L'arbre de décision \mathbf{T} construit.

Soient,

- $N(p)$: le nombre d'individus associés à la position (nœud) \mathbf{p} .
- $N(k|p)$: le nombre d'individus appartenant à la classe k en sachant qu'ils sont associés à la position \mathbf{p} .
- $P(k|\mathbf{p}) = \frac{N(k|\mathbf{p})}{N(\mathbf{p})}$: proportion des individus appartenant à la classe k parmi ceux de la position \mathbf{p} .

Jeu de Données

- Soit le tableau suivant récapitulant l'ensemble des clients d'une compagnie d'assurance.

Id Client	Montant (M)	Age (A)	Résidence (R)	Etudes (E)	Internet (I)
1	Moyen	moyen	Village	Oui	Oui
2	élevé	moyen	Bourg	Non	Non
3	Faible	âgé	Bourg	Non	Non
4	Faible	moyen	Bourg	Oui	Oui
5	Moyen	jeune	Ville	Oui	Oui
6	élevé	âgé	Ville	Oui	Non
7	Moyen	âgé	Ville	Oui	Non
8	Faible	moyen	village	non	non

- M : salaire ou moyenne des montants sur le compte.
- A : âge du client.
- R : lieu de résidence du client.
- E : le client a fait des études supérieures ou non ?
- I : le client consulte ses comptes sur internet ou non ? (classe) 140

Mélange Initial

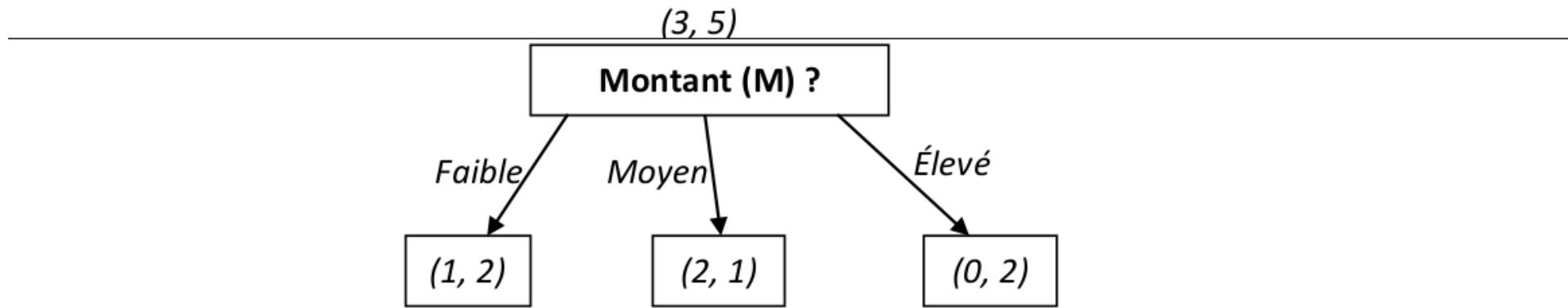
- Avec 8 clients dont : 3 (oui) ont Internet (classe 1 : oui) et 5 non (classe 2 : non), le mélange initial (selon Gini) :

$$\text{Mélange initial} = \text{Mélange(Gini)} = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2 = \frac{15}{32} = 0,46875$$

- La construction est descendante : on commence par tester les candidats à la racine.
- Au début, tous les individus sont regroupés (au niveau 0, la racine de l'arbre).
- Ainsi, quatre (04) constructions sont possibles, suivant les variables : Montant (M), âge (A), résidence (R) et études (E).

Tester les Candidates à la Racine

a. Construction selon la variable M (Montant)



$1 \text{ oui}, 2 \text{ non}$

$2 \text{ oui}, 1 \text{ non}$

$0 \text{ oui}, 2 \text{ non}$

$$\text{Mélange}(Faible) = Gini = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = \frac{4}{9}$$

$$\text{Mélange}(élevé) = Gini = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

On calcule le Gain selon la variable M :

$$\text{Mélange}(Moyen) = Gini = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$Gain(M) = \text{mélange}(initial) - \frac{n_{Faible}}{n} \times \text{mélange}(F) - \frac{n_{Moyen}}{n} \times \text{mélange}(M) - \frac{n_{élevé}}{n} \times \text{mélange}(E)$$

$$Gain(M) = \frac{15}{32} - \frac{3}{8} \cdot \frac{4}{9} - \frac{3}{8} \cdot \frac{4}{9} - \frac{2}{8} \cdot 0 = \frac{13}{96} = 0,135$$

Tester les Candidats à la Racine

b. Construction selon la variable A (âge)

Après avoir calculé les mélanges selon Gini, on calcule le Gain selon la variable A :

$$Gain(A) = \text{mélange}(initial) - \frac{n_{Jeune}}{n} \times \text{mélange}(J) - \frac{n_{Moyen}}{n} \times \text{mélange}(M) - \frac{n_{âgé}}{n} \times \text{mélange}(Ä)$$

$$Gain(A) = \frac{15}{32} - \frac{1}{8} \cdot 0 - \frac{4}{8} \cdot \frac{1}{2} - \frac{3}{8} \cdot 0 = \frac{7}{32} = 0,219$$

c. Construction selon la variable R (Résidence)

On calcule le Gain selon la variable R :

$$Gain(R) = \text{mélange}(initial) - \frac{n_{Village}}{n} \times \text{mélange}(V) - \frac{n_{Bourg}}{n} \times \text{mélange}(B) - \frac{n_{ville}}{n} \times \text{mélange}(Vi)$$

$$Gain(R) = \frac{15}{32} - \frac{2}{8} \cdot \frac{1}{2} - \frac{3}{8} \cdot \frac{4}{9} - \frac{3}{8} \cdot \frac{4}{9} = \frac{1}{96} = 0,010$$

Tester les Candidats à la Racine

d. Construction selon la variable E (étude)

Après avoir calculé les mélanges selon Gini, on calcule le Gain par rapport la variable E :

$$Gain(E) = mélange(initial) - \frac{n_{oui}}{n} \times mélange(O) - \frac{n_{non}}{n} \times mélange(N)$$

$$Gain(E) = \frac{15}{32} - \frac{5}{8} \cdot \frac{12}{25} - \frac{3}{8} \cdot 0 = \frac{27}{160} = 0,169$$

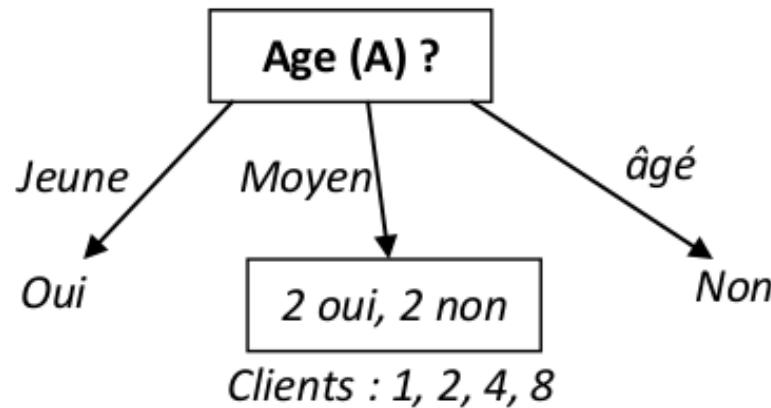
Quel Test Choisir ?

Variable Test	Composition nœuds	Gain
Montant (M)	(1, 2) ; (2, 1) ; (0, 2)	0,135
Age (A)	(1, 0) ; (2, 2) ; (0, 3)	0,219
Résidence (R)	(1, 2) ; (1, 2) ; (1, 1)	0,010
Etudes (E)	(3, 2) ; (0, 3)	0,169

Remarque,

- Sur la *variable R*, aucune discrimination sur aucune branche, ainsi on ne gagne rien avec ce test !
- Sur la *variable A*, deux nœuds sur trois sont “**purs**”, ce qui semble intéressant !

Le Premier Niveau de l'Arbre Appris



L'étape suivante consiste à **ignorer les valeurs** (les supprimer du tableau de valeurs) pour laquelle “Age = jeune” et “Age = âgé” (pour les lignes : 3, 5, 6, 7) et **ne pas prendre en considération** la variable Age A (retirer la colonne Age).

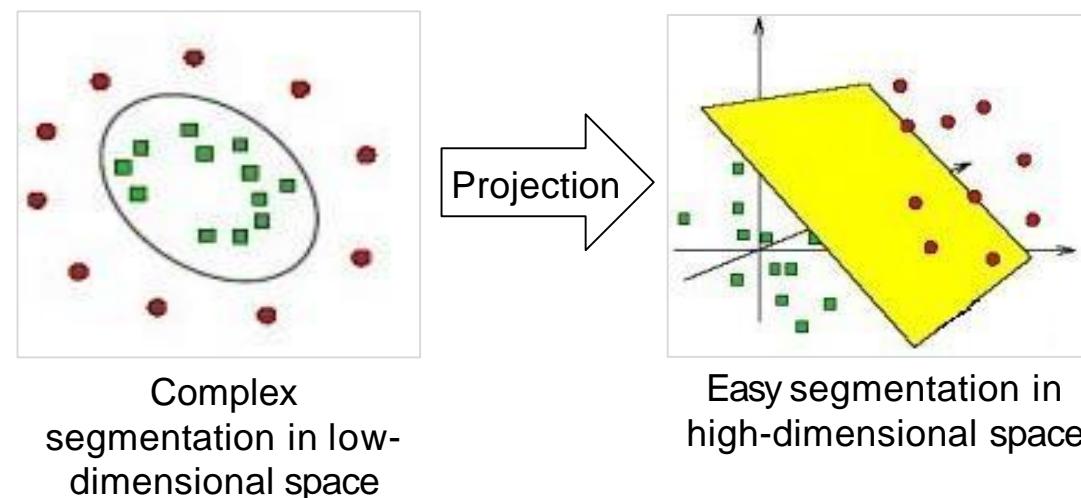
Puis, continuer la construction des autres niveaux selon les variables restantes, à savoir : **M**, **R** et **E**.

Exercice

N°	Pif	Temp	Humid	Vent	Golf	la classe
1	soleil	chaud	haute	faux	NePasJouer	
2	soleil	chaud	haute	vrai	NePasJouer	
3	couvert	chaud	haute	faux	Jouer	
4	pluie	bon	haute	faux	Jouer	
5	pluie	frais	normale	faux	Jouer	
6	pluie	frais	normale	vrai	NePasJouer	
7	couvert	frais	normale	vrai	Jouer	
8	soleil	bon	haute	faux	NePasJouer	
9	soleil	frais	normale	faux	Jouer	
10	pluie	bon	normale	faux	Jouer	
11	soleil	bon	normale	vrai	Jouer	
12	couvert	bon	haute	vrai	Jouer	
13	couvert	chaud	normale	faux	Jouer	
14	pluie	bon	haute	vrai	NePasJouer	

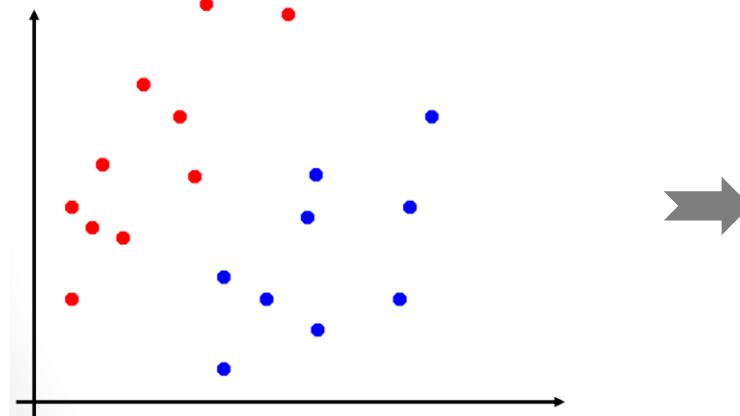
SVM (Support Vector Machine)

- SVM est un modèle de classification binaire dont le modèle de base est un classificateur linéaire défini dans l'espace propre avec le plus grand intervalle. Les SVM incluent également des astuces de noyau qui en font des classificateurs non linéaires. L'algorithme d'apprentissage SVM est la solution optimale à la programmation quadratique convexe.

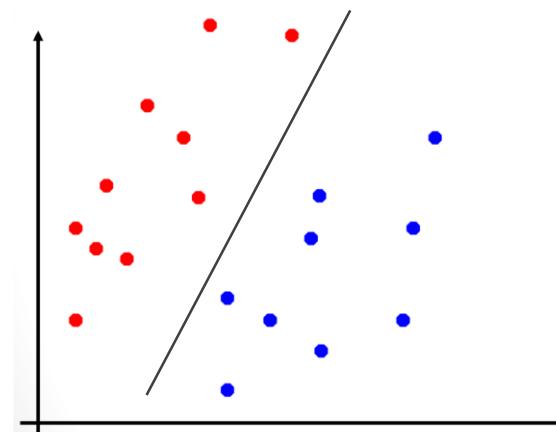


SVM linéaire

- Comment diviser les jeux de données rouge et bleu par une ligne droite ?

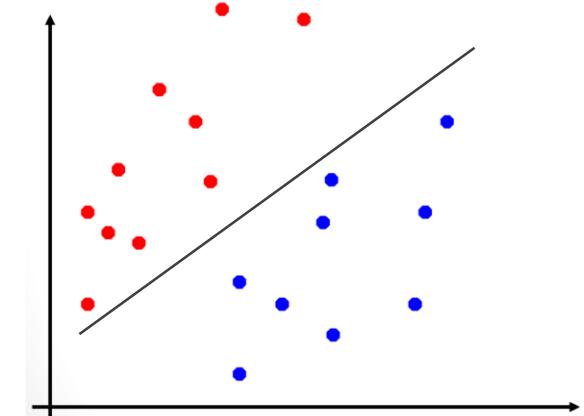


With binary classification
Two-dimensional dataset



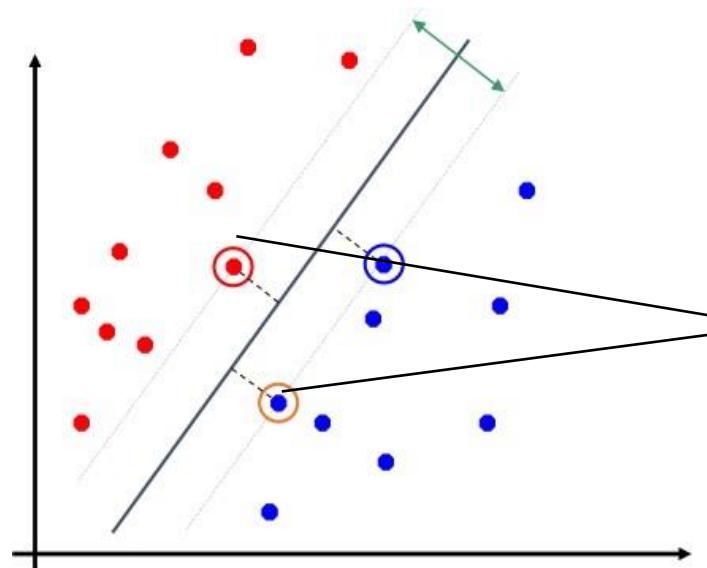
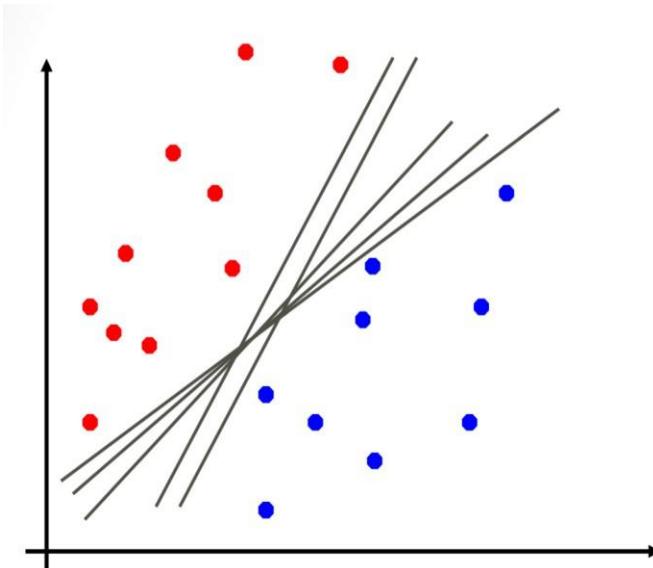
Both the left and right methods can be used to divide datasets. Which of them is correct?

or

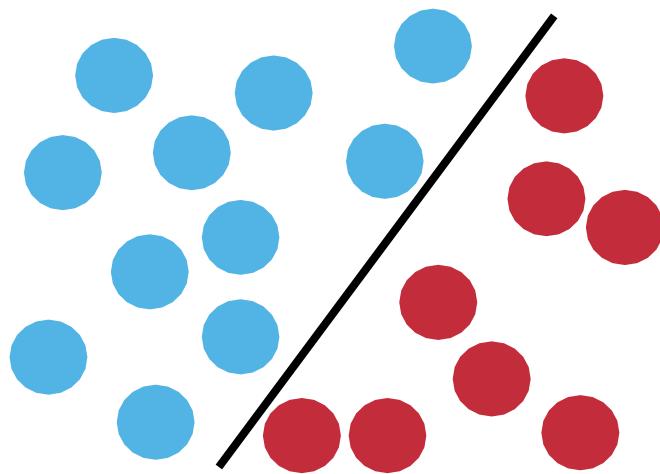


SVM linéaire

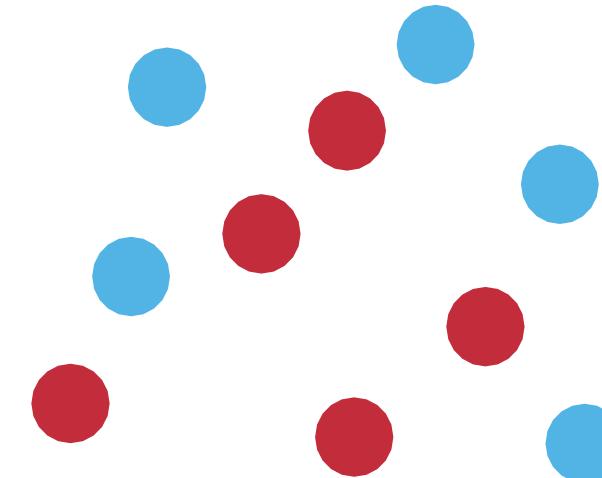
- Les lignes droites sont utilisées pour diviser les données en différentes classes. En fait, nous pouvons utiliser plusieurs lignes droites pour diviser les données. L'idée de base de la SVM est de trouver une ligne droite et de garder le point proche de la ligne droite aussi loin que possible de la ligne droite. Cela peut permettre une forte capacité de généralisation du modèle. Ces points sont appelés vecteurs supports.
- Dans l'espace à deux dimensions, nous utilisons des lignes droites pour la segmentation. Dans l'espace de grande dimension, nous utilisons des hyperplans pour la segmentation.



SVM non linéaire



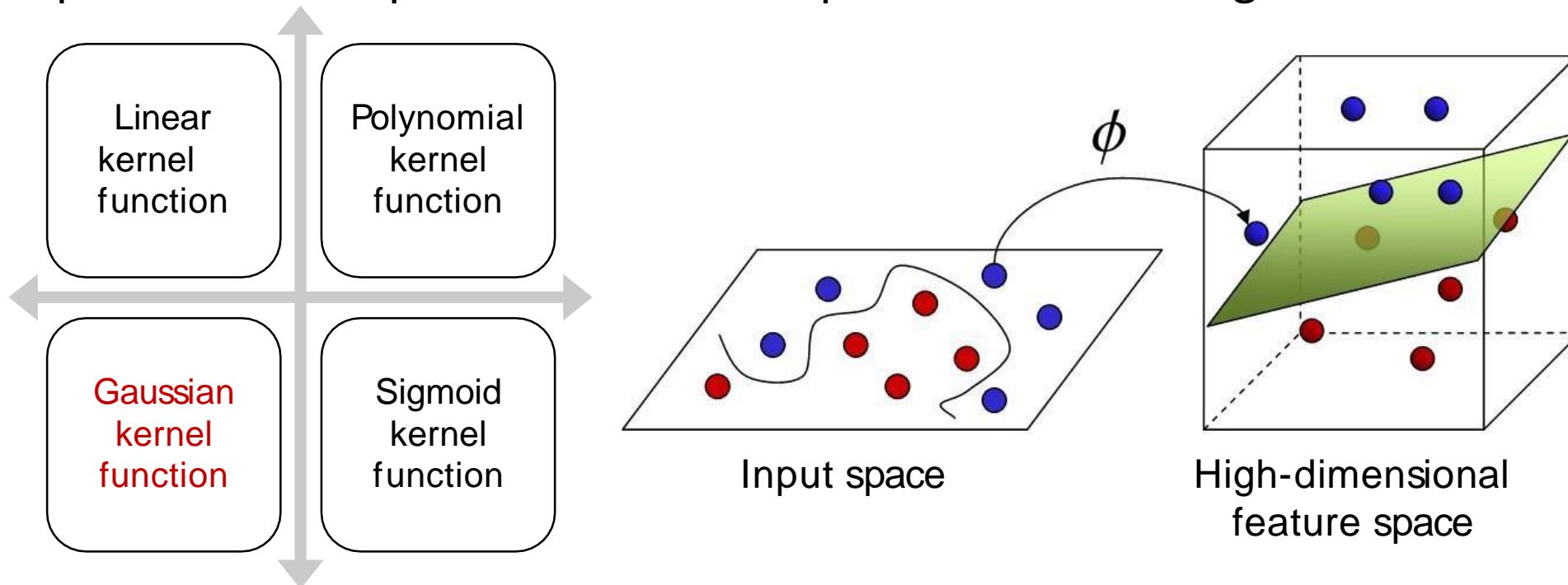
Le SVM linéaire peut bien fonctionner pour les ensembles de données séparables linéairement.



Les jeux de données non linéaires ne peuvent pas être fractionnés avec des lignes droites.

SVM non linéaire

- Les fonctions du noyau sont utilisées pour construire des SVM non linéaires.
- Les fonctions de noyau permettent aux algorithmes de s'adapter au plus grand hyperplan dans un espace de caractéristiques transformé de grande dimension.

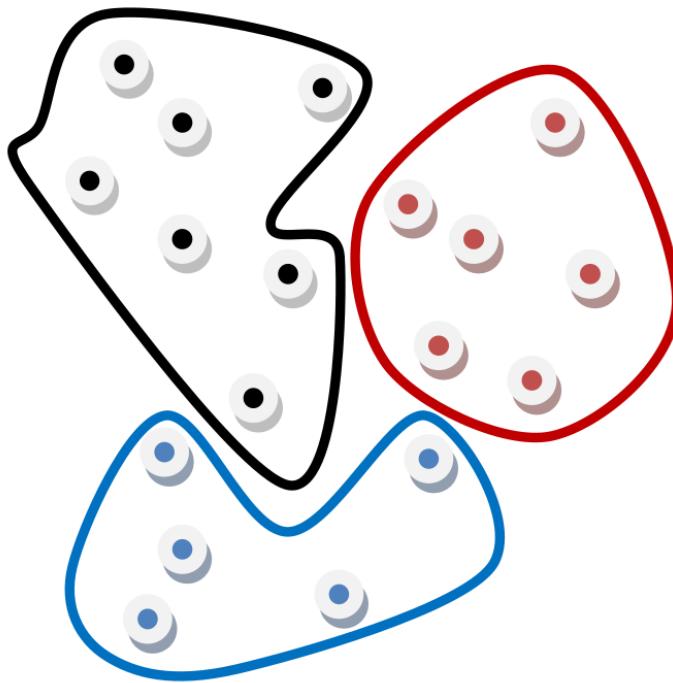


Méthodes de classification non supervisées

- Il s'agit d'une tâche principale en intelligence artificielle et dans la fouille exploratoire de données.
- C'est une technique d'analyse statistique des données très utilisée dans de nombreux domaines y compris l'apprentissage automatique, la reconnaissance de formes, le traitement d'images, la recherche d'information, etc.
- L'idée est de **découvrir des groupes au sein des données, de façon automatique.**

Algorithme des Centres Mobiles

- **K-means** est un algorithme de minimisation alternée qui, étant donné un entier K , va chercher à séparer un ensemble de points en K **clusters** ou groupes



Combien de Classes ?

- Le nombre K représente le nombre de classes que l'algorithme doit former à partir des propriétés des échantillons ou exemples.
- Le nombre de groupes K peut être supposé fixe (donné par l'utilisateur) ou fixé par la nature du problème à traiter.
 - C'est le cas, si l'on s'intéresse à classer des images de chiffres manuscrits (nombre de classes = 10 : 0, ..., 9) ou de lettres manuscrites (nombre de classes = nombres de caractères de l'alphabet), etc.

Données, Classes et Métrique

- Considérons une image couleur. On peut donc représenter le $i^{\text{ème}}$ pixel par un vecteur x_i de dimension 3 : $x_i = (x_{i1}, x_{i2}, x_{i3}) \in \{0, 1, \dots, 255\}^3$.
- Ainsi, si le problème de classification consiste à répartir les données pixel en 3 classes : rouge, verte, bleue. L'idée revient à mesurer la proportion (métrique en taux) de chaque composante de couleur (R%, V% et B%) contenue dans chaque pixel.
- Mais, on peut aussi procéder de la manière suivante : Si l'on connaît les classes de certains pixels, on pourra prédire les classes des autres pixels en choisissant une mesure de **(dis)similarité** ou de **(dis)ressemblance**,

Exemple

- 7 objets représentés chacun par un descripteur à 2 paramètres
- On veut grouper ces données (selon leurs similarités) en deux ($k=2$) groupes : G 1 et G 2 .

Sujets	X	Y
1	2,0	2,0
2	3,0	4,0
3	6,0	8,0
4	10,0	14,0
5	7,0	10,0
6	9,0	10,0
7	7,0	9,0

Exemple

- On procèdera de la manière suivante :
 1. A partir des n données (x_1, x_2, \dots, x_n) , choisir (au hasard) les $k = 2$ centres $\mu = (\mu_1, \mu_2, \dots, \mu_k)$ des **k groupes** initiaux $G = (G_1, G_2, \dots, G_k)$ à générer.
 2. A partir de l’itération $t = 0$, et pour chaque objet x_i ($i = 1 \dots n$) :
 - a. Calculer sa distance $dist(x_i, \mu_j)$ de x_i à chaque centre μ_j ($j = 1 \dots k$),
 - b. Affecter ou réaffecter x_i au groupe G_l de centre μ_l (qu’est le plus proche à x_i), si $dist(x_i, \mu_l) = minimale$.
 3. Recalculer le centre μ_j (la moyenne) de chaque groupe G_j ($j = 1 \dots k$),

1^{ère} étape : Initialisation (**itération 0**) : Recherche d'une partition initiale.

Ça consiste à choisir les centres initiaux des 2 groupes. Prenant, par exemple, les objets 1 et 4 (les plus éloignés $-min$ et max , selon une distance ; optant pour la distance Euclidienne) comme centres de G_1 et G_2 .

Itération 1^{ère} :

2^{ème} étape : Les objets restants sont examinés un par un et localisés par rapport au plus proche cluster.

Ceci nous mènent à calculer d'abord les distances de chaque objet aux 2 groupes G_1 et G_2 , représentés respectivement par leurs centres ou centroïdes C_1 et C_2 ; ces distances sont données par le tableau suivant (les valeurs des distances sont arrondîtes, et on ne retient qu'un seul chiffre après la virgule) :

3^{ème} étape : Une fois que les distances sont connues, on réaffecte chaque objet au cluster G_i , si sa distance au centre C_i est minimale.

Puisque, on n'est pas sûr que chaque objet soit assigné au **bon cluster**, ceci nous oblige à **réitérer** une fois de plus sur l'**étape 2** : ce qui revient à recalculer les distances de chaque objet aux 2 nouveaux centres C_1 et C_2 . Puis, réaffecter chaque objet au groupe dont la distance à son centre est minimale.

Itération 2^{ème} :

Recalculons les distances de chaque objet aux 2 groupes G_1 et G_2 .

Une fois les distances recalculées, on réaffecte chaque objet au cluster G_i qui lui est le plus proche. Ici, seul l'objet 3 est reclassé dans G_2 . Puis, on recalcule les barycentres des nouveaux groupes.

Itération 3^{ème} :

Recalculons les distances de chaque objet aux 2 groupes G_1 et G_2 ; on obtient :

On recalcule les barycentres des nouveaux groupes, mais ce n'est pas nécessaire puisque les groupes n'ont pas changé, ainsi, il y a ***convergence***.

Condition de Convergence :

L'itération s'arrête lorsque chaque objet est plus proche de son propre moyen cluster que celui des autres groupes, et la solution finale des clusters sera ce dernier partitionnement.

Propriétés du K-means

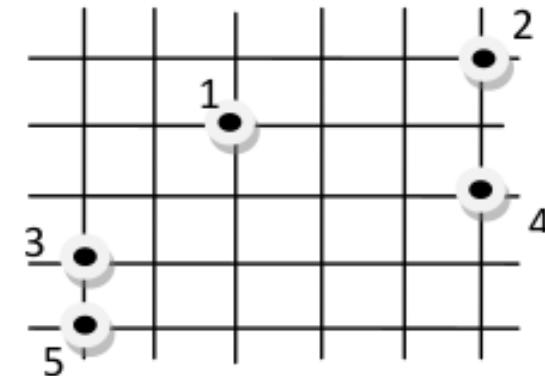
- C'est un algorithme de regroupement simple et rapide, mais aussi très utilisé.
- La méthode k-means minimise une mesure de dissemblance intra-classe pour les k-groupes.
- Chaque objet est affecté au cluster dont le centre (centroïde/barycentre) est le plus proche.
- Le centre d'un groupe est la moyenne de tous les points (éléments) de ce groupe.
- Son inconvénient est qu'il produit un résultat différent à chaque exécution (initialisation).

Classification Hiérarchique Ascendante

- La classification hiérarchique ou (re)groupement hiérarchique (ou clustering) est une méthode de classification automatique qui consiste à effectuer une suite de regroupements en agrégeant à chaque étape les objets (données ou descripteurs d'objets) ou les groupes d'objets les plus proches.
- On trouve des applications dans des domaines très divers tels que la biologie (classements par espèce, genre), l'archéologie, le traitement d'images, le traitement de requêtes, etc.

Exemple

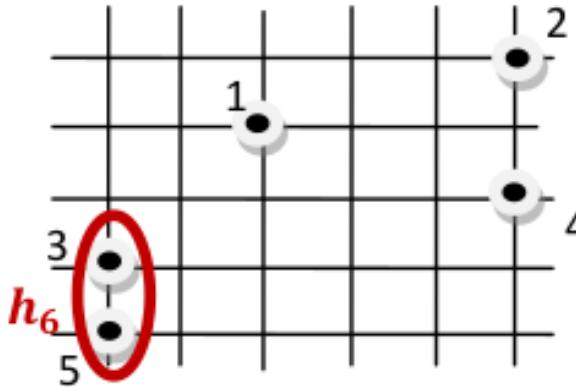
- Soit un ensemble d'objets représentés par des points numérotés de 1 à 5, dans un repère euclidien.
- Notons d la distance euclidienne mesurée entre les objets.



d	1	2	3	4	5
1	0	$\sqrt{10}$	$\sqrt{8}$	$\sqrt{10}$	$\sqrt{13}$
2		0	$\sqrt{34}$	2	$\sqrt{41}$
3			0	$\sqrt{26}$	1
4				0	$\sqrt{29}$
5					0

1^{ère} étape : Cette première étape de la méthode nous conduit à regrouper les **objets 3 et 5**, qui sont les points les **plus proches** (distance minimale), $d = 1$, et à former un groupe $h_6 = \{3, 5\}$.

A ce groupe est associé son **niveau**, ou **indice d'agrégation f**, qui est la distance entre ses deux sous-groupes $\{3\}$ et $\{5\}$, $f(h_6) = 1$.

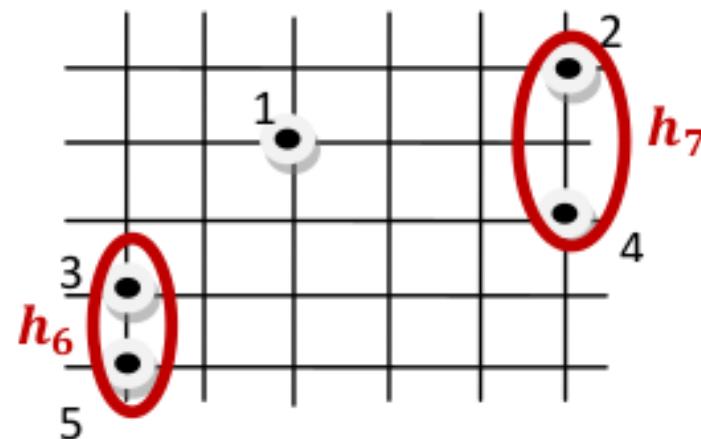


Plusieurs solutions sont possibles. Nous en proposons deux à titre d'exemples :

1. **Le saut minimal** : qui consiste à affecter à la distance entre deux groupes la distance entre leurs objets **les plus proches**, et
2. **Le diamètre maximal** : qui consiste à retenir la distance entre leurs objets **les plus éloignés**.

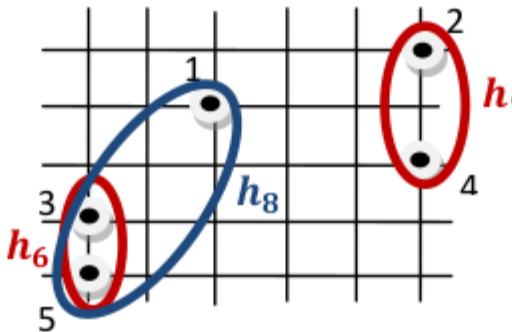
2^{ème} étape : A l'étape suivante, on a la même agrégation pour les deux distances $d = 2$ (même distance minimale), $h_7 = \{2, 4\}$, $f(h_7) = 2$:

		<i>Saut minimal</i>	
d	1	h_6	h_7
1	0	$\sqrt{8}$	$\sqrt{10}$
h_6		0	$\sqrt{26}$
h_7			0



3^{ème} étape : A la troisième étape, les deux hiérarchies deviennent différentes :

- **Saut minimal :** $h_8 = \{1\} \cup h_6 = \{1,3,5\}$, $f(h_8) = \sqrt{8}$



Saut minimal		
d	h_7	h_8
h_7	0	$\sqrt{10}$
h_8		0

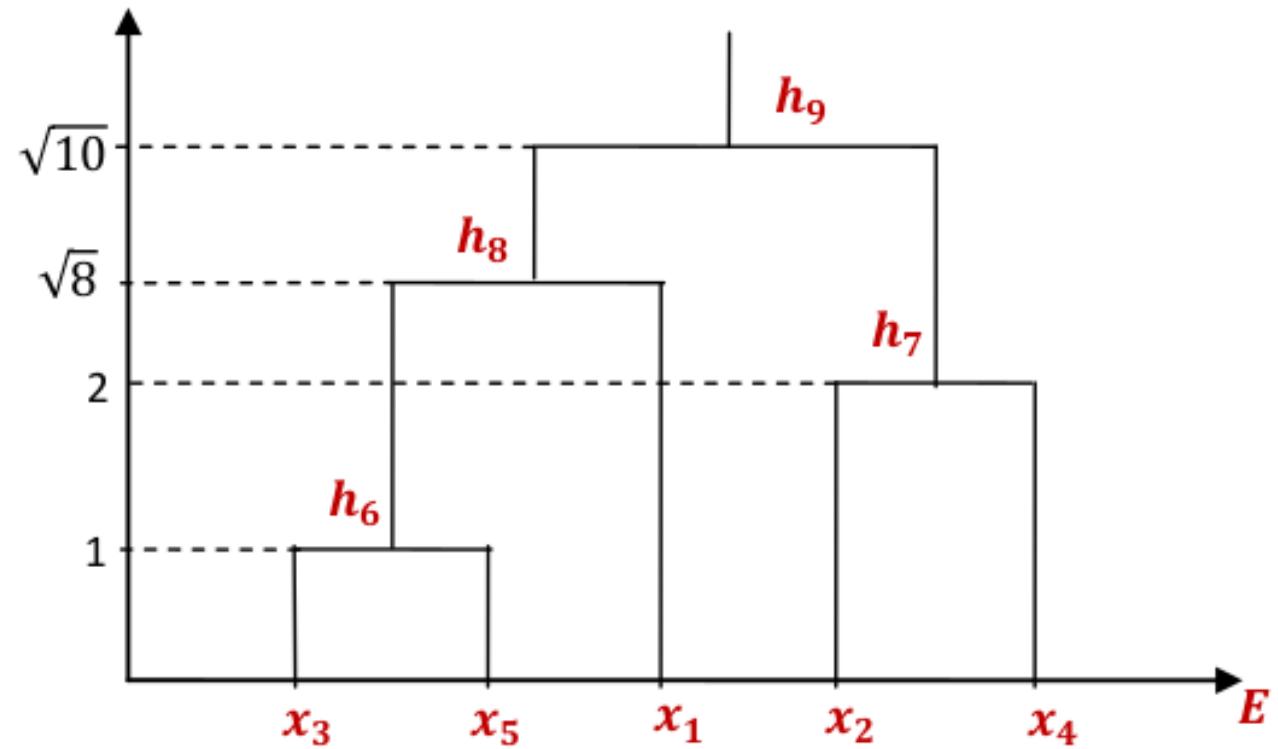
Si on continue à la dernière étape ($4^{\text{ème}}$ itération de regroupement), tous les objets sont regroupés :

- **Saut minimal :** $h_9 = h_7 \cup h_8 = \{1,2,3,4,5\}$, $f(h_9) = \sqrt{10}$

Remarque important : Le *choix de la distance* entre deux groupes influe sur les regroupements.

Dendrogramme

- Cette hiérarchie de regroupement, clustering, des objets peut être représentée par un diagramme dit : **dendrogramme**. C'est une représentation arborescente d'une hiérarchie.



Etapes du Regroupement Hiérarchique

12

D'une manière générale, l'algorithme de clustering ou (re)groupement hiérarchique ascendant peut comprendre plusieurs **itérations**, dont le nombre maximal d'itérations est donné par le **nombre** et/ou la **nature** des objets à regrouper.

Cependant, on peut décider de l'arrêt du regroupement (convergence) par un **seuil de regroupement** selon le **niveau d'agrégation désiré**.

Chaque itération de l'algorithme de classification par clustering hiérarchique comprend :

1. Calcul de la matrice des distances entre objets, pour la 1^{ère} étape (ou mise à jour de la matrice des distances entre objets, pour les étapes suivantes).
2. Déterminer la distance minimale (niveau d'agrégation) et repérer les groupes ou objets concernés.
3. Tester si le niveau d'agrégation n'a pas dépassé le seuil ou le nombre d'itérations est suffisant, sinon arrêt.
4. Regrouper les objets ou groupes concernés. Réitérer sur 1.

Recherche d'une Hiérarchie Indicée à partir d'une Ultramétrique

Le processus de construction de la hiérarchie à partir d'une ultramétrique u est récursif. Il fonctionne comme suit

1. On forme l'ensemble des parties à un élément : $\{x_1\}, \{x_2\}, \dots, \{x_n\}$.
2. On agrège les deux éléments h_i et h_j les plus proches au sens de u , la distance ultramétrique reste stable par cette agrégation (en effet, pour tout nœud h_k , $\{h_i, h_j, h_k\}$ forme un triangle isocèle dont la base est $\{h_i, h_j\}$).
3. On réitère l'étape 2 sur $\{h_1, h_2, \dots, \{h_i, h_j\}, \dots, h_p\}$ et ceci jusqu'à ce que tous les éléments soient regroupés en une seule classe.

Exemple

u	1	2	3	4	5
1	0	8	8	8	5
2		0	2	4	8
3			0	4	8
4				0	8
5					0

On agrège $\{2\}$ et $\{3\}$ et on recommence avec $\{\{1\}, \{2,3\}, \{4\}, \{5\}\}$.

Pour obtenir la distance entre les classes $\{1\}$ et $\{2,3\}$, on remarque que, u étant une ultramétrique et $\{2,3\}$ étant les points les plus proches, $\{1,2,3\}$ est un triangle isocèle de base $\{2,3\}$, donc $u(1, 2) = u(1, 3)$.

Il suffit ainsi de poser $u(1, \{2,3\}) = u(1, 2)$. Cette distance étendue aux nœuds reste une ultramétrique.

u	1	$\{2,3\}$	4	5
1	0	8	8	5
$\{2,3\}$		0	4	8
4			0	8
5				0

On agrège $\{2,3\}$ et $\{4\}$:

u	1	$\{2,3,4\}$	5
1	0	8	5
$\{2,3,4\}$		0	8
5			0

On agrège $\{1\}$ et $\{5\}$:

u	$\{1,5\}$	$\{2,3,4\}$
$\{1,5\}$	0	8
$\{2,3,4\}$		0

On agrège enfin $\{1,5\}$ et $\{2,3,4\}$. On obtient ainsi la hiérarchie

