

IFT3390/6390

# Fondements de l'apprentissage machine

<http://www.iro.umontreal.ca/~vincentp/ift3390>

Sixième cours:

**Distribution Gaussienne multivariée.**  
**Évaluation de la performance de généralisation.**  
**Courbes d'apprentissage.**

Professeur: Pascal Vincent

# Au programme aujourd'hui

- Opérations sur les Distributions
- Gaussienne multivariée.
- Évaluation de performance de généralisation.
- Courbes d'apprentissage

Première partie

# Distributions

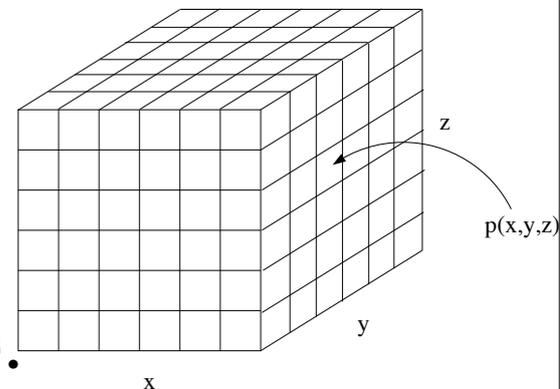
# Distributions

Voir <http://www.techno-science.net/?onglet=glossaire&definition=6395>  
et le rappel de proba de Balazs

- On associe naturellement une **loi de probabilité (ou distribution)** à une variable aléatoire pour décrire la *répartition des valeurs qu'elle peut prendre*.
- La distribution d'une variable aléatoire peut se caractériser par sa **fonction de distribution cumulative (c.d.f.)**:

$$F_X(x) = P(X \leq x) = P(X_1 \leq x_1, \dots, X_n \leq x_n)$$

- La loi d'une **variable discrète** est déterminée par la **probabilité de chacune des valeurs** qu'elle peut prendre. => table de probabilités (qui doivent sommer à 1).



- La loi d'une **variable continue** peut être donnée sous la forme d'une fonction de **densité de probabilité (p.d.f.)** qui est la dérivée de la c.d.f. *La probabilité qu'un tirage de la variable tombe dans une certaine région de l'espace est l'intégrale de la densité sur cette région.*

# Opérations avec les distributions

Avec une distribution, on peut vouloir:

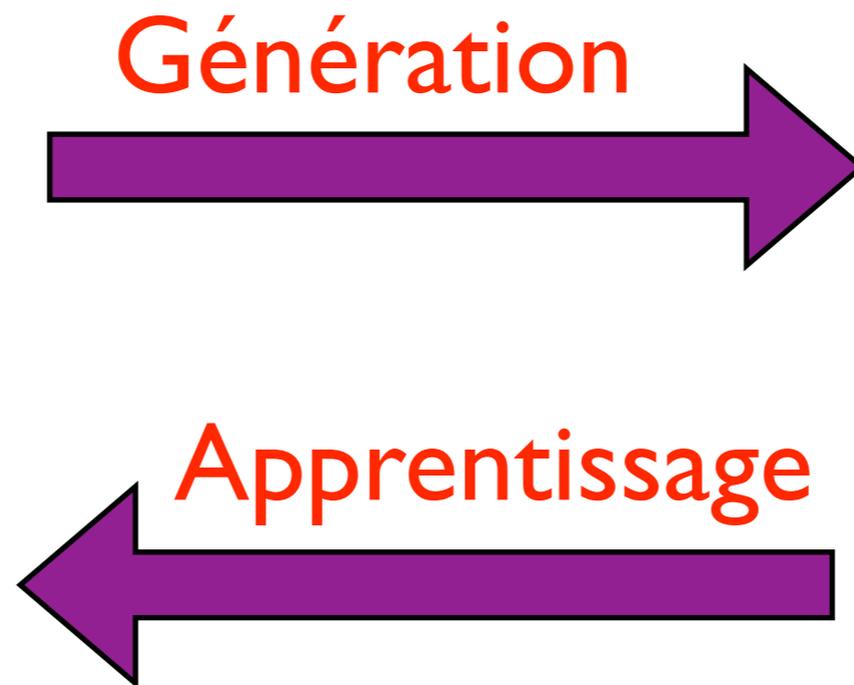
- **Générer des données**, c.a.d. tirer des échantillons selon la distribution.
- **Calculer la (log) probabilité d'une configuration** (sachant la valeur de certaines des variables et ayant marginalisé celles dont on ne connaît pas la valeur).
- **Inférence**: *inférer* la valeur la plus probable ou la valeur espérée d'un sous ensemble de variables sachant la valeur des autres.
- **Apprentissage** des paramètres de la distribution **à partir d'un ensemble de données** (de sorte à maximiser la probabilité que les données soient générées par cette distribution avec ces paramètres: principe du *maximum de vraisemblance*).

# Ex. variable scalaire discrète $X$

Table de probabilité:

$x$	$P(X=x)$
1	0.10
2	0.80
3	0.10

Ensemble de données:



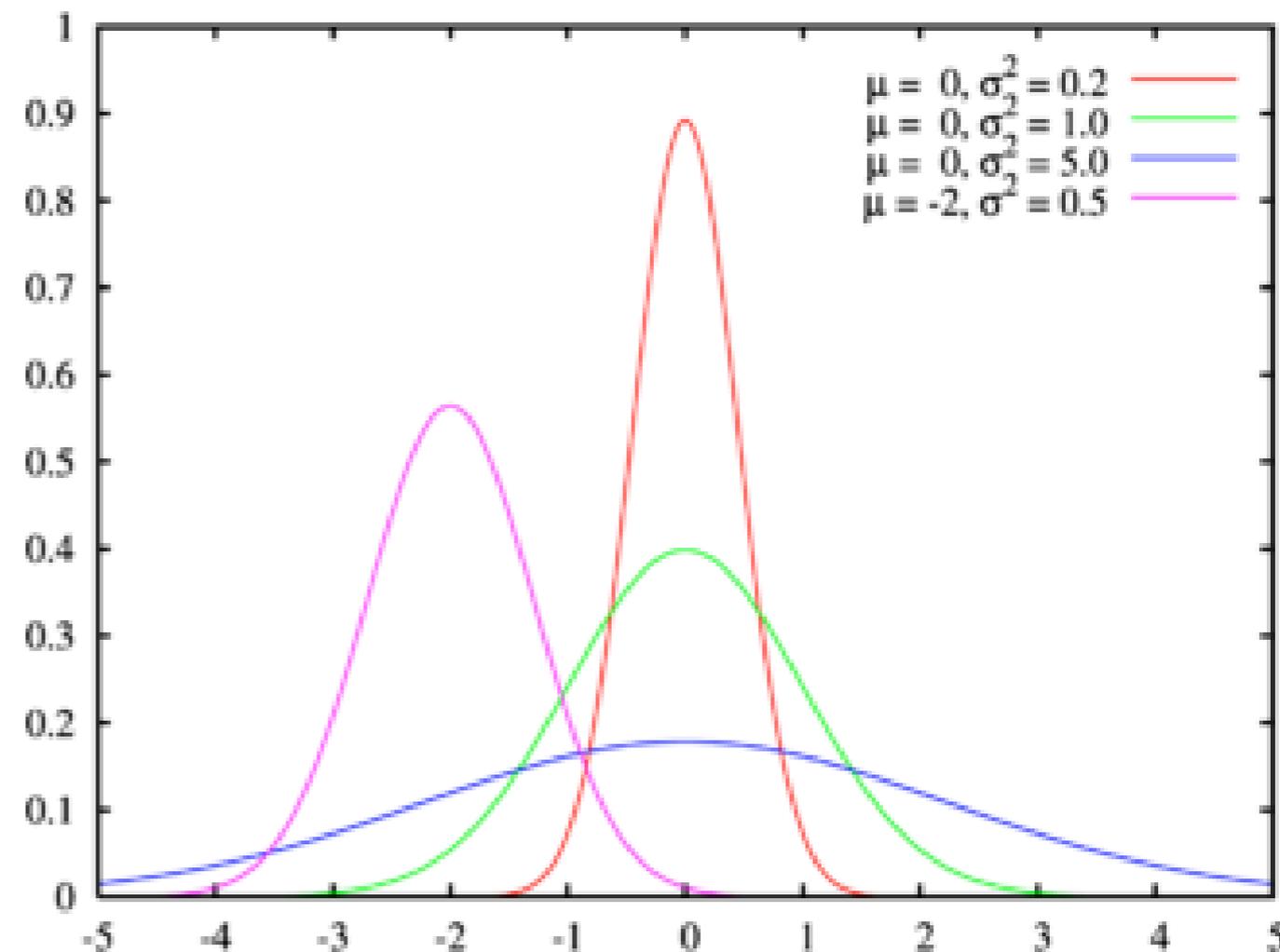
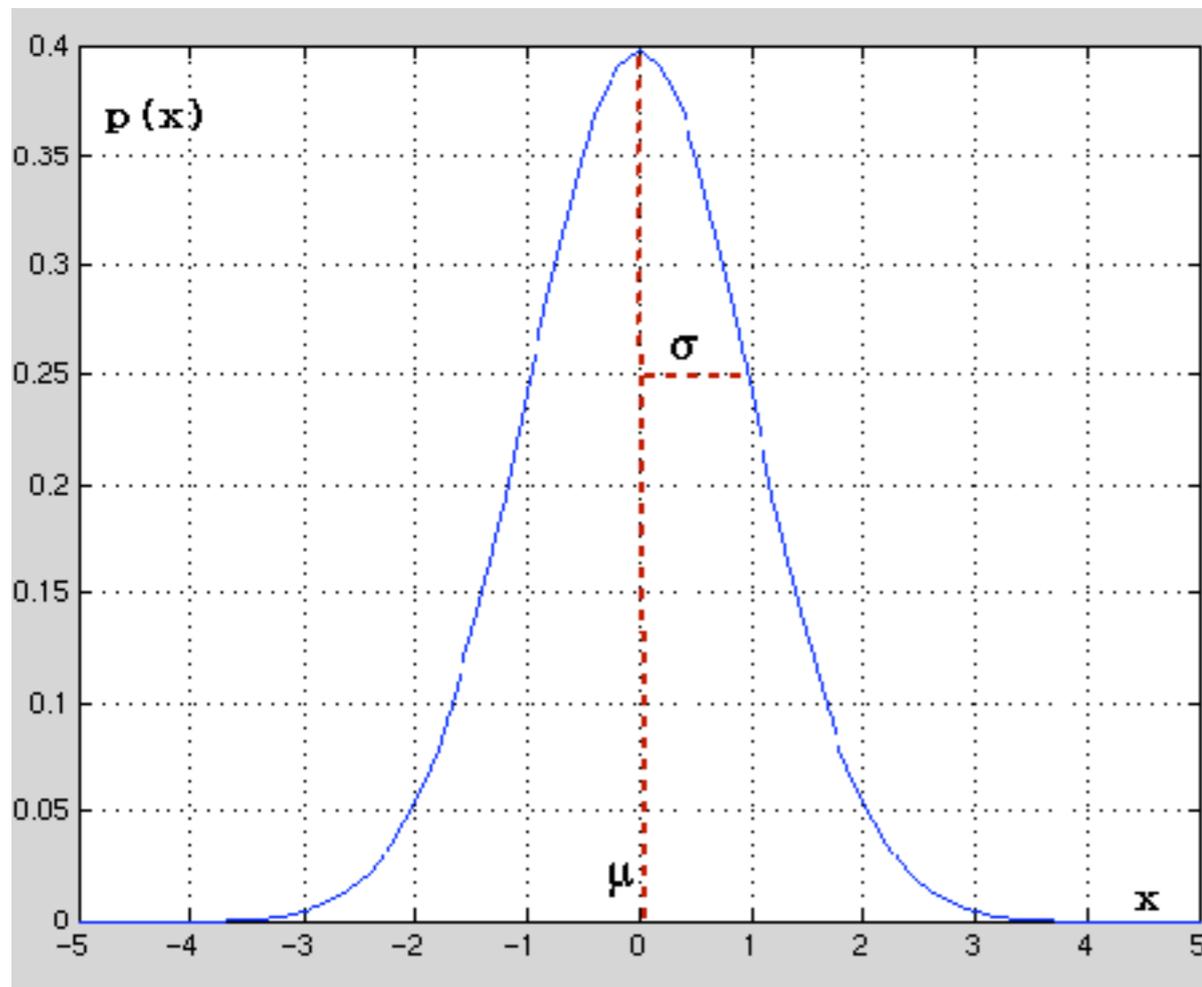
2  
2  
2  
1  
2  
2  
2  
2  
3  
2  
2  
⋮

# La densité Gaussienne univariée

(distribution dite Gaussienne ou Normale)

Densité Gaussienne univariée (c.a.d. en dimension 1) de moyenne  $\mu$  et de variance  $\sigma^2$  (écart type  $\sigma$ ).

$$p(x) = \mathcal{N}_{\mu, \sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

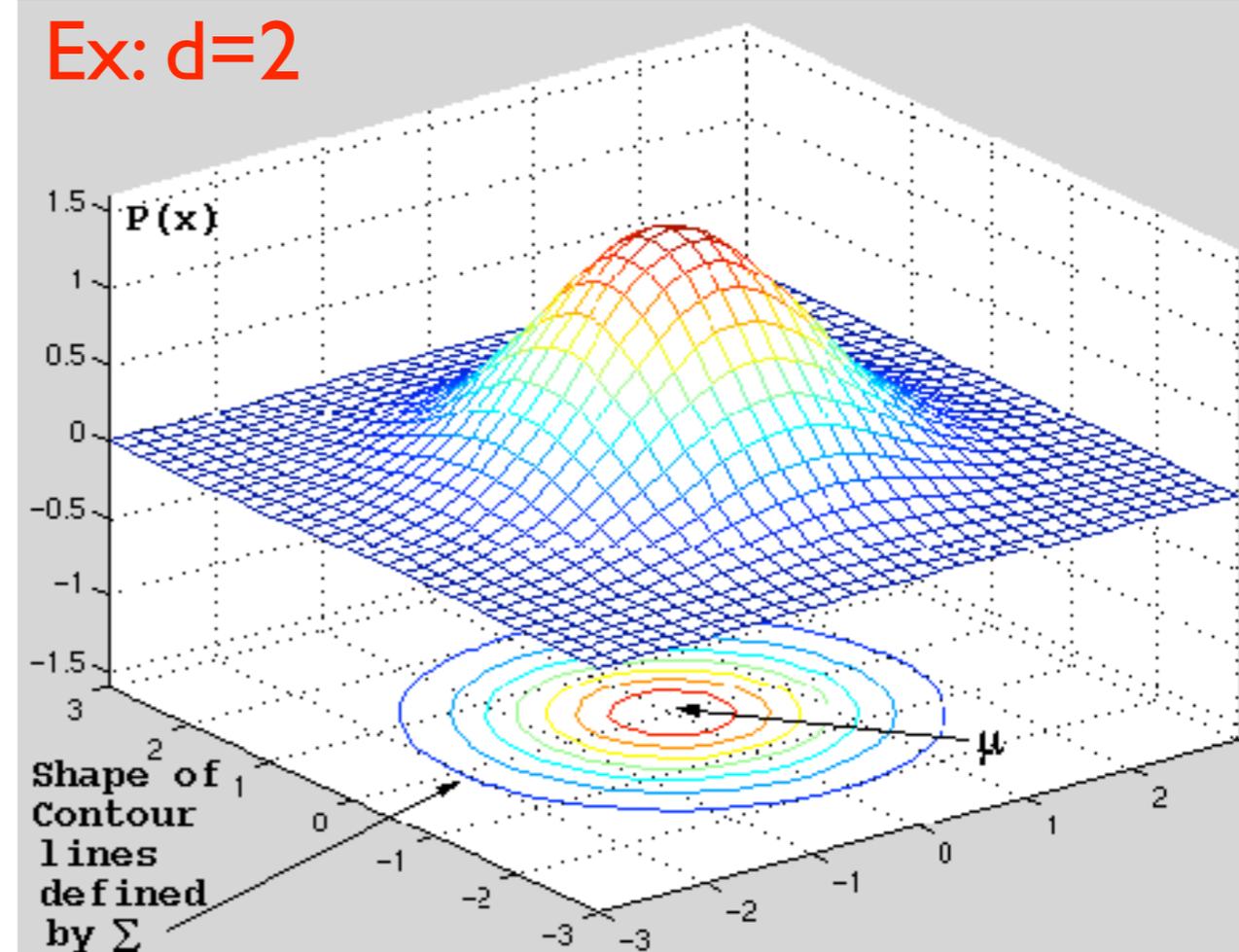


# La densité Gaussienne multivariée

Gaussienne isotropique  
("sphérique") en dimension d:

$$p(x) = \mathcal{N}_{\mu, \sigma^2}(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} e^{-\frac{1}{2} \frac{\|x - \mu\|^2}{\sigma^2}}$$

Il s'agit d'une "bosse" Gaussienne  
**"centrée"** en  $\mu$  et de **"largeur"** , la  
même dans toutes les directions.



Gaussienne générale en dimension d, de moyenne  $\mu$  et de  
matrice de covariance  $\Sigma$ .

$$p(x) = \mathcal{N}_{\mu, \Sigma}(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{|\Sigma|}} e^{-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)}$$

$$\mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$

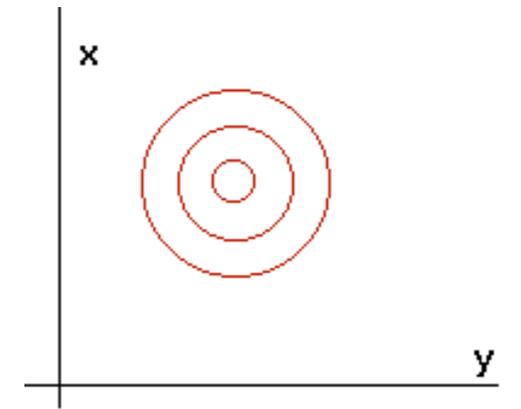
} déterminant de  $\Sigma$

*Note: le dénominateur est une simple normalisation qui assure que la densité intègre à 1, mais ne change rien à sa forme*

# La densité Gaussienne multivariée

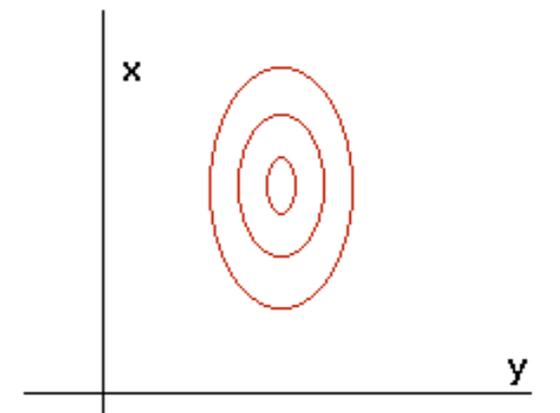
## *matrices de covariance particulières*

- Gaussienne isotropique ou sphérique:  $\Sigma = \sigma^2 I$   
(avec  $I$  la matrice identité)



- Gaussienne diagonale:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & & \mathbf{0} \\ & \sigma_2^2 & \\ \mathbf{0} & & \ddots \\ & & & \sigma_d^2 \end{pmatrix}$$

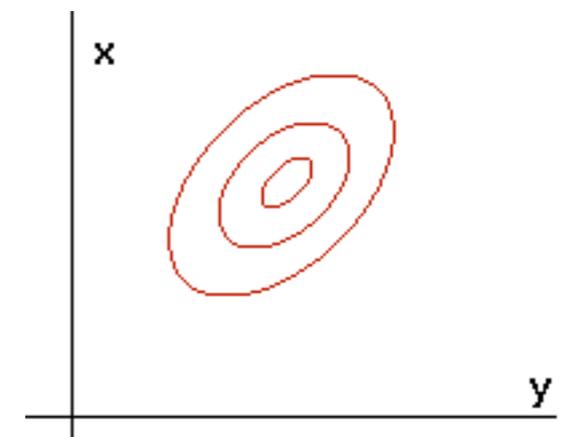


- Décomposition en valeurs propres/vecteurs propres:

$$\Sigma = \sum_{i=1}^d \lambda_i v_i v_i^T$$

*Les vecteurs propres indiquent les directions des axes de l'ellipsoïde associée, les valeurs propres leur "longueur".*

*Le déterminant  $|\Sigma| = \lambda_1 \lambda_2 \dots \lambda_d$  indique la "taille" de l'ellipsoïde.*



# La densité Gaussienne multivariée

## *apprentissage des paramètres*

- On peut **apprendre les paramètres** d'une Gaussienne à partir d'un ensemble de données de manière simple:
- Pour  $\mu$  on calcule la **moyenne empirique** des points (le "centroïde").
- Pour  $\Sigma$  on calcule la **matrice de covariance empirique** des données.
- On verra comment dériver ces résultats dans un prochain cours (*principe du maximum de vraisemblance*).

Deuxième partie

# Evaluation de la performance de généralisation

# Estimation de l'erreur de généralisation

## **Le problème:**

- Principe de **minimisation du risque empirique**: on entraîne un modèle (on adapte ses paramètres) de manière à ce qu'il fasse un **minimum d'erreurs sur l'ensemble d'entraînement**.
- **MAIS** ce qui nous intéresse vraiment, c'est de **bien généraliser sur de nouveaux exemples**.
- Puisque les paramètres du modèle sont choisis, spécialisés, pour minimiser **l'erreur sur les exemples d'entraînement**, celle-ci **sous-estime l'erreur de généralisation**.
- Ainsi **l'erreur d'entraînement n'est pas un bon estimé de l'erreur de généralisation** (c'est un estimé **biaisé**).

# Estimation d'erreur de généralisation: Validation simple

Ensemble de toutes les données étiquetées dont on dispose:

$D =$

$(x_1, y_1)$

$(x_2, y_2)$

⋮

$(x_N, y_N)$

On sépare nos données en **deux**

(attention il peut s'avérer nécessaire de d'abord mélanger les rangées de données)

Ensemble d'entraînement (taille  $n$ )

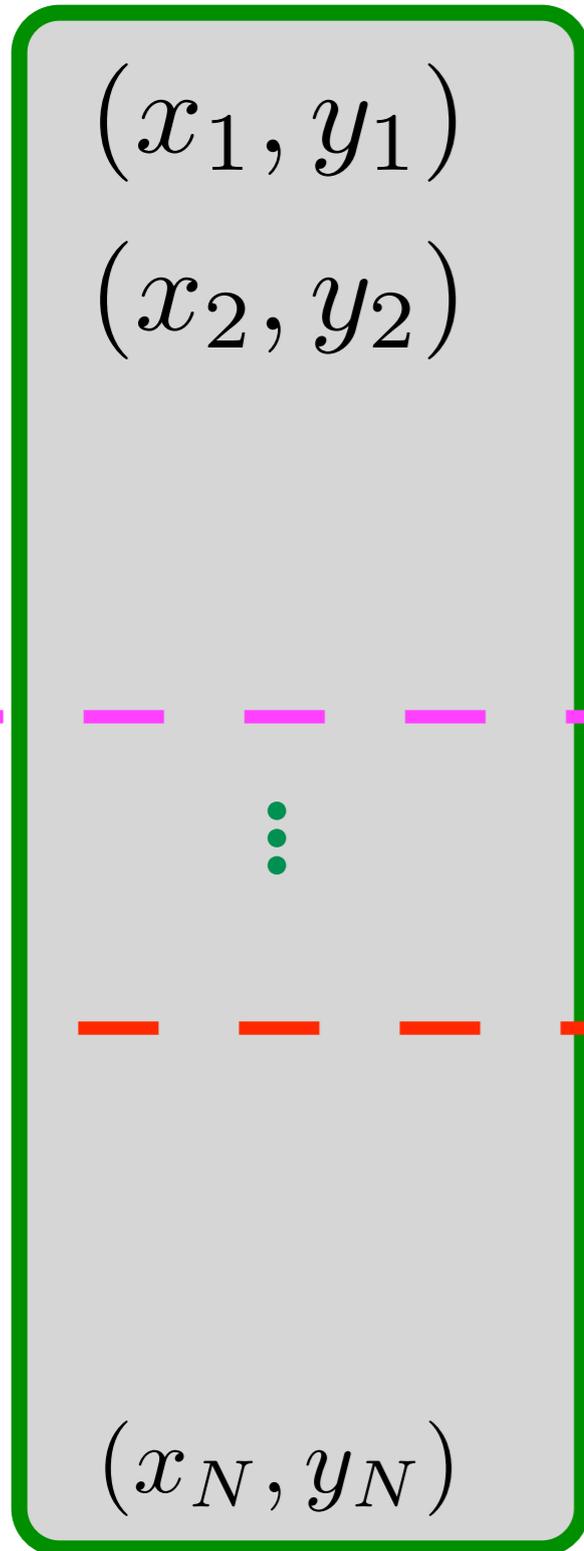
On entraîne le modèle pour minimiser l'erreur sur l'ensemble d'entraînement

Ensemble de test (taille  $m$ )

On évalue la performance de généralisation en mesurant les erreurs sur l'ensemble de test qu'on n'a jamais regardé pendant l'entraînement. (mesure de performance "hors échantillon").

# Sélection des hyper-paramètres

$D =$



On sépare nos données en **trois**

Ensemble d'entraînement (taille n)

Ensemble de validation (taille n')

Ensemble de test (taille m)

Ex: choisir le  $k$  des  $k$ -PPV

Pour chaque valeur d'hyper-paramètres considérée:

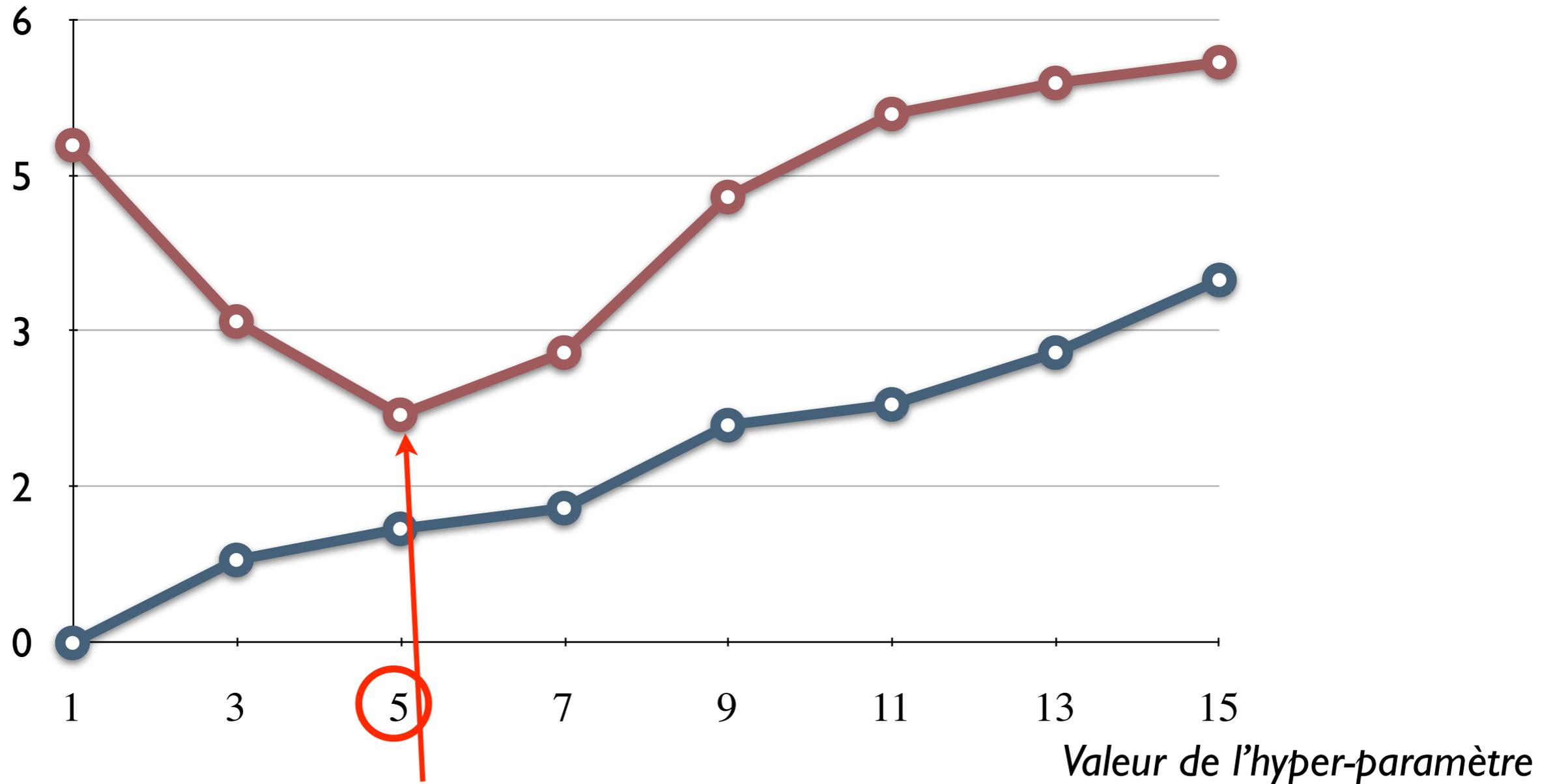
- 1) On **entraîne le modèle** pour minimiser l'erreur sur l'ensemble d'entraînement.
- 2) On évalue l'erreur sur **l'ensemble de validation**.

On utilise finalement la valeur des hyper-paramètres donnant l'erreur la plus basse sur l'ensemble de validation. (possibilité de réentraîner sur les  $n+n'$  exemples ou non).

On **évalue la performance de généralisation** en mesurant les erreurs sur l'ensemble de test qu'on n'a jamais regardé pendant l'entraînement / validation (mesure de performance "hors échantillon").

# Courbes d'apprentissage

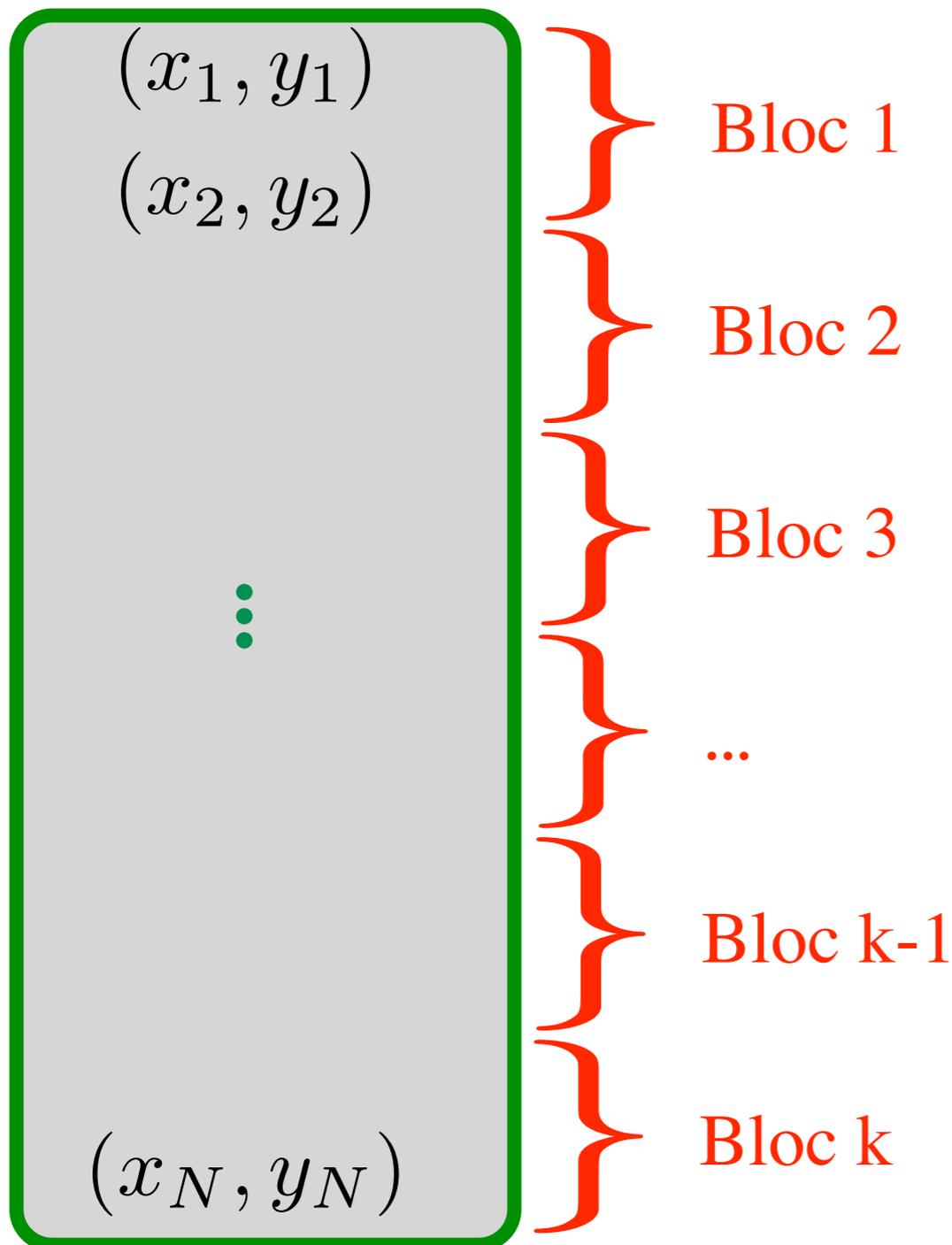
- Erreur d'apprentissage
- Erreur de validation



La valeur de l'hyper-paramètre donnant l'erreur minimale sur l'ensemble de validation est 5 (alors que c'est 1 pour l'ensemble d'apprentissage)

# Si on n'a pas assez de données: validation croisée (en $k$ blocs)

$D =$



Idée simple: on répète plusieurs fois la procédure entraînement/test en divisant différemment l'ensemble de données.

<i>Entraînement sur</i>	<i>Calcul de l'erreur (test) sur</i>
$D \setminus \text{Bloc 1}$	Bloc 1
$D \setminus \text{Bloc 2}$	Bloc 2
...	...
$D \setminus \text{Bloc } k$	Bloc $k$

Notre estimé de l'erreur de généralisation de l'algorithme sur ce problème est déduit de la somme des erreurs obtenues sur tous les blocs.

En Anglais on appelle cela *k-fold cross validation*.  
Le cas où  $k=N$  est appelé *leave-one-out* ou *jackknife*.

# Double validation croisée

- Si on a peu de données et qu'on veut faire de la sélection d'hyper-paramètre ET obtenir un estimé non biaisé de l'erreur de généralisation de la procédure, on peut faire une double validation croisée:
- Un premier niveau de validation croisée est utilisé pour obtenir des paires d'ensembles entraînement/test
- Un deuxième niveau (imbriqué) de validation croisée où l'on divise l'ensemble d'entraînement sert à sélectionner les hyper-paramètres.