

# **Apprentissage automatique**

Classification linéaire - fonction discriminante

# TYPES D'APPRENTISSAGE

**Sujets:** apprentissage supervisé, classification, régression

**RAPPEL**

- L'apprentissage supervisé est lorsqu'on a une cible à prédire
  - **classification** : la cible est un indice de classe  $t \in \{1, \dots, K\}$ 
    - exemple : reconnaissance de caractères
      - ✓  $x$  : vecteur des intensités de tous les pixels de l'image
      - ✓  $t$  : identité du caractère
  - **régression** : la cible est un nombre réel  $t \in \mathbb{R}$ 
    - exemple : prédiction de la valeur d'une action à la bourse
      - ✓  $x$  : vecteur contenant l'information sur l'activité économique de la journée
      - ✓  $t$  : valeur d'une action à la bourse le lendemain

# TYPES D'APPRENTISSAGE

**Sujets:** apprentissage supervisé, classification, régression

**RAPPEL**

- L'apprentissage supervisé est lorsqu'on a une cible à prédire

‣ **classification** : la cible est un indice de classe  $t \in \{1, \dots, K\}$

- exemple : reconnaissance de caractères
  - ✓  $x$  : vecteur des intensités de tous les pixels de l'image
  - ✓  $t$  : identité du caractère

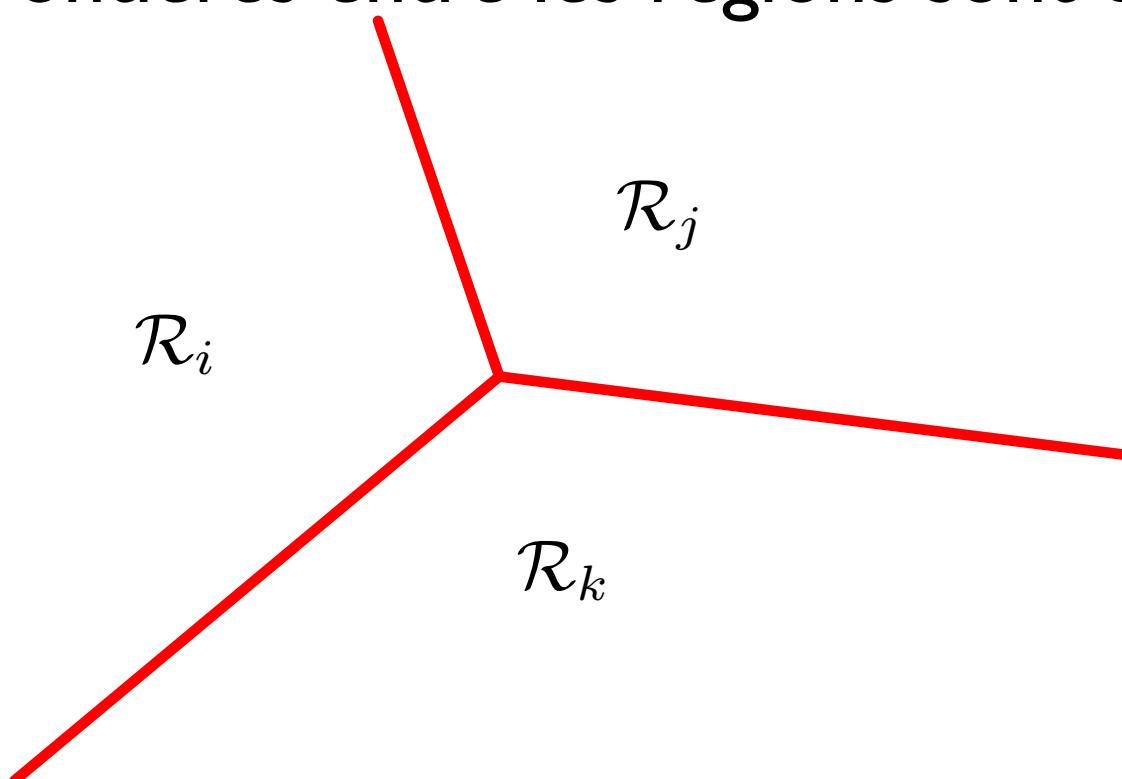
‣ **régression** : la cible est un nombre réel  $t \in \mathbb{R}$

- exemple : prédiction de la valeur d'une action à la bourse
  - ✓  $x$  : vecteur contenant l'information sur l'activité économique de la journée
  - ✓  $t$  : valeur d'une action à la bourse le lendemain

# CLASSIFICATION

**Sujets:** surface de décision, région de décision

- On cherche à diviser l'espace des entrées  $x$  en différentes **régions de décision**
  - chaque région de décision  $\mathcal{R}_k$  est associée à une classe  $\mathcal{C}_k$
  - les frontières entre les régions sont des **surfaces de décision**

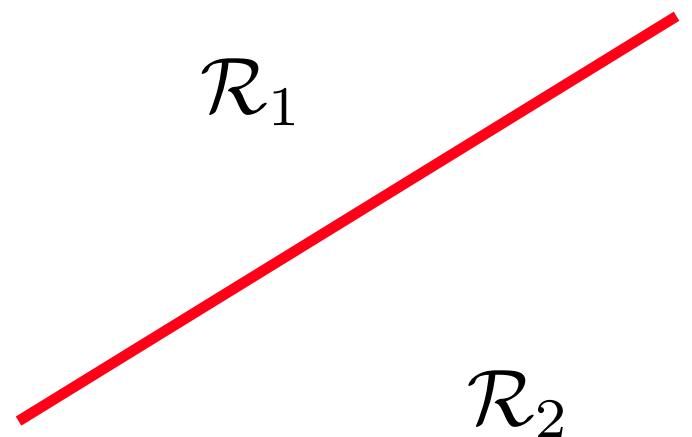


# CLASSIFICATION

**Sujets:** classification binaire, séparabilité linéaire

- Cas spécial : classification binaire

- classe  $\mathcal{C}_1$  correspond à  $t = 1$
- classe  $\mathcal{C}_2$  correspond à  $t = 0$  (ou  $t = -1$ )



- Cas spécial : classification linéaire

- la surface de décision entre chaque paire de régions de décision est linéaire, i.e. un hyperplan (droite pour  $D=2$ )
- on dit qu'un problème est **linéairement séparable** si une surface linéaire permet de classifier parfaitement

# FONCTION DISCRIMINANTE

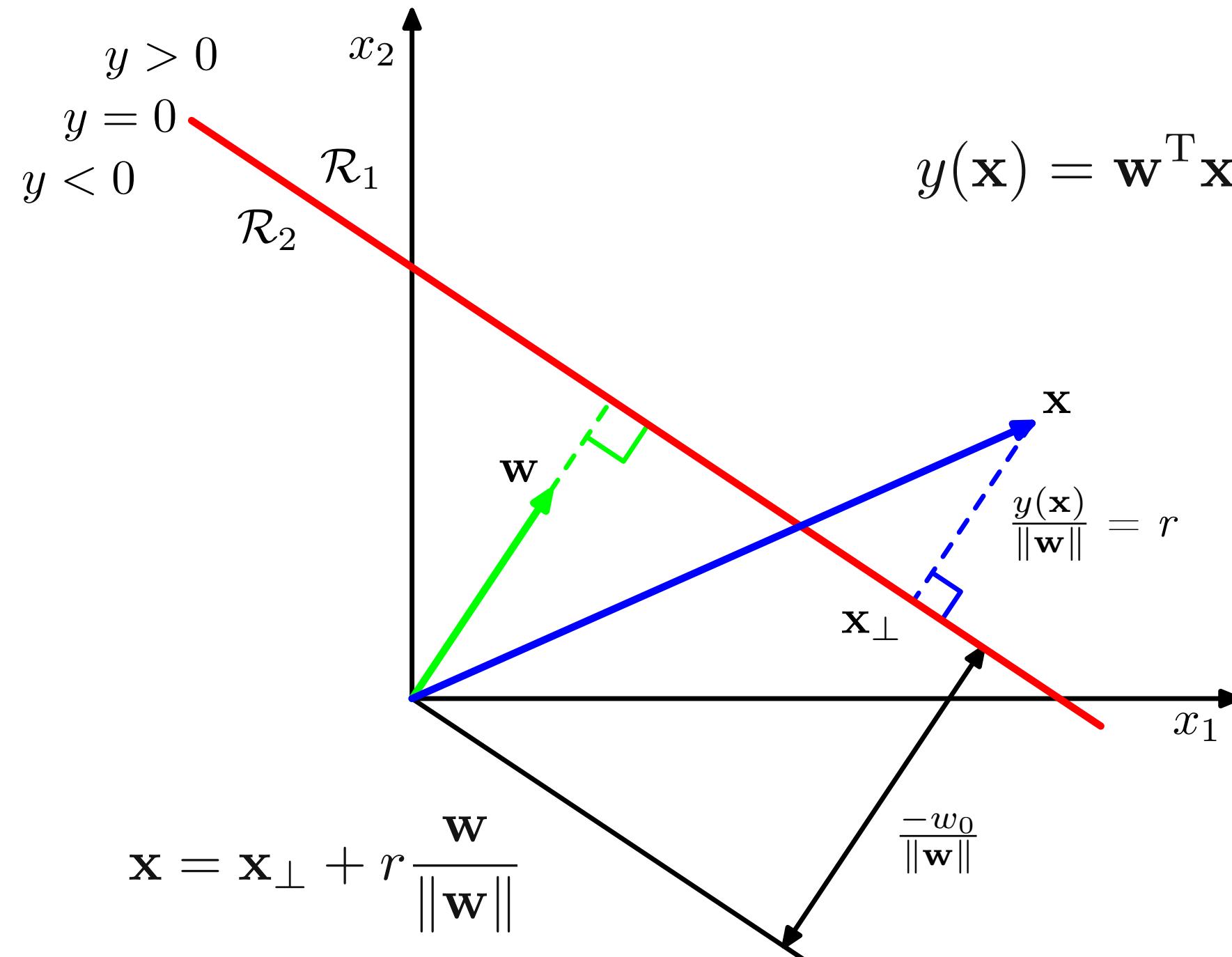
# Sujets: fonction discriminante, vecteur de poids, biais

- On souhaite apprendre une **fonction discriminante** qui prend  $x$  en entrée et donne sa classe  $\mathcal{C}_k$  en sortie
  - Dans le cas binaire, on va s'intéresser aux fonctions discriminantes qui :
    - I. calculent une transformation linéaire de l'entrée

2. retourne  $\mathcal{C}_1$  si  $y(\mathbf{x}) \geq 0$  ou retourne  $\mathcal{C}_2$  sinon

# FONCTION DISCRIMINANTE

**Sujets:** fonction discriminante, vecteur de poids, biais



# SÉPARABILITÉ LINÉAIRE

**Sujets:** séparabilité linéaire

- Est-ce que l'hypothèse de séparabilité linéaire est raisonnable ?
  - en haute dimensionnalité (grande valeur de  $D$ ), possiblement !
- **Théorème :** soit  $D+1$  entrées  $\mathbf{x}_n$ , on peut toujours les séparer linéairement en 2 classes, quelque soit la valeur de leurs cibles  $t_n$
- On peut également utiliser une représentation  $\phi(\mathbf{x})$  qui elle est non-linéaire

**Condition :**  
chaque sous-ensemble de  
 $D$  entrées est  
linéairement indépendant

# FONCTION DISCRIMINANTE

**Sujets:** entraînement

- Idéalement, on voudrait entraîner  $y(\mathbf{x})$  en minimisant directement le taux d'erreur de classification sur l'ensemble d'entraînement
  - malheureusement, on peut démontrer que c'est un problème NP-difficile
- On va donc devoir attaquer le problème indirectement
  - ceci va donner lieu à différents algorithmes d'apprentissage

# **Apprentissage automatique**

Classification linéaire - méthode des moindres carrés

# MÉTHODE DES MOINDRES CARRÉS

**Sujets:** méthode des moindres carrés

- On va traiter la classification comme un problème de régression
  - on pourrait prédire directement la valeur de la cible ( $t = 1$  vs.  $t = -1$ )
  - si  $y(\mathbf{x}) \geq 0$  on classifie dans  $\mathcal{C}_1$  sinon  $\mathcal{C}_2$
- On parle de **moindres carrés** puisque la régression minimise la différence au carré entre  $t$  et  $y(\mathbf{x})$

# RÉGULARISATION

**Sujets:** régularisation, weight decay, régression de Ridge

**RAPPEL**

- On peut montrer que la solution (maximum a posteriori) est alors :

$$\mathbf{w} = (\lambda \mathbf{I} + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

- dans le cas  $\lambda = 0$ , on retrouve la solution tu maximum de vraisemblance
- si  $\lambda > 0$ , permet également d'avoir une solution plus stable numériquement (si  $\Phi^T \Phi$  n'est pas inversible)

# MÉTHODE DES MOINDRES CARRÉS

**Sujets:** méthode des moindres carrés

- Pour le cas à plus de deux classes, on va traiter la classification comme un problème de régression à prédiction multiple
  - la cible va être un vecteur binaire indiquant à quelle classe appartient l'entrée
  - exemple : s'il y a  $K=5$  classes et qu'une entrée est de la classe  $\mathcal{C}_2$

$$\mathbf{t} = (0, 1, 0, 0, 0)^T$$

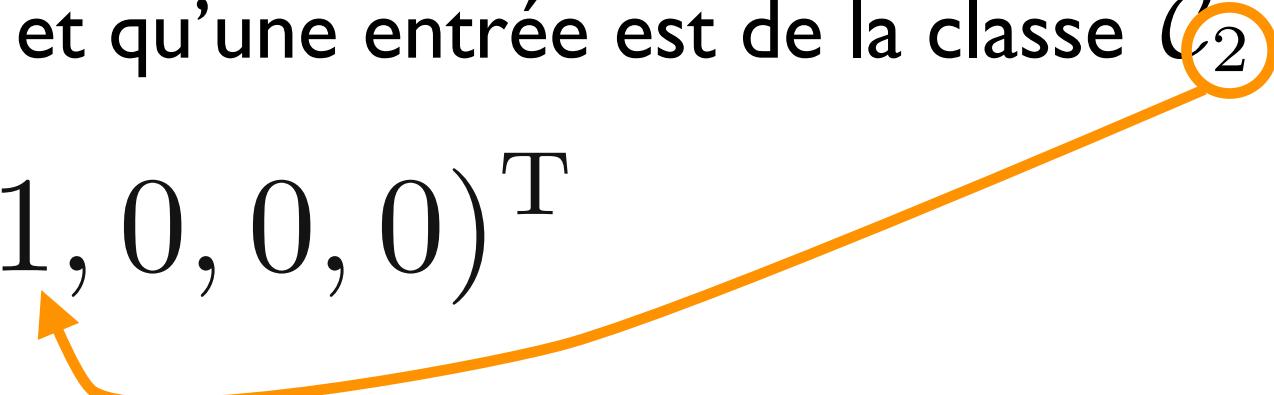
- on classifie dans la classe  $\mathcal{C}_k$  dont la valeur de  $y(\mathbf{x})_k$  est la plus élevée

# MÉTHODE DES MOINDRES CARRÉS

**Sujets:** méthode des moindres carrés

- Pour le cas à plus de deux classes, on va traiter la classification comme un problème de régression à prédiction multiple
  - la cible va être un vecteur binaire indiquant à quelle classe appartient l'entrée
  - exemple : s'il y a  $K=5$  classes et qu'une entrée est de la classe  $\mathcal{C}_2$

$$\mathbf{t} = (0, 1, 0, 0, 0)^T$$



- on classe dans la classe  $\mathcal{C}_k$  dont la valeur de  $y(\mathbf{x})_k$  est la plus élevée

# PRÉDICTIONS MULTIPLES

**Sujets:** modèle pour prédictions multiples

**RAPPEL**

- Le modèle doit maintenant prédire un vecteur :

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \boldsymbol{\phi}(\mathbf{x})$$

où  $\mathbf{W}$  est une matrice  $M \times K$

- Chaque colonne de  $\mathbf{W}$  peut être vu comme le vecteur  $\mathbf{w}_k$  du modèle  $y(\mathbf{x}, \mathbf{w}_k)$  pour la  $k^e$  cible

# PRÉDICTIONS MULTIPLES

**Sujets:** modèle pour prédictions multiples

**RAPPEL**

- Le modèle doit maintenant prédire un vecteur :

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \boldsymbol{\phi}(\mathbf{x})$$

où  $\mathbf{W}$  est une matrice  $M \times K$

- Chaque colonne de  $\mathbf{W}$  peut être vu comme le vecteur  $\mathbf{w}_k$  du modèle  $y(\mathbf{x}, \mathbf{w}_k)$  pour la  $k^e$  cible classe

# MAXIMUM DE VRAISEMBLANCE

**Sujets:** formulation probabiliste pour prédictions multiples

**RAPPEL**

- On peut démontrer que le maximum de vraisemblance est :

$$\mathbf{W}_{\text{ML}} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{T}$$

- On peut voir le résultat comme la concaténation (colonne par colonne) des solutions pour chaque tâche

$$\mathbf{w}_k = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}_k$$

où  $\mathbf{t}_k = (t_{1,k}, \dots, t_{N,k})^T$

# **Apprentissage automatique**

Classification linéaire - analyse discriminante linéaire

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** analyse discriminante linéaire

- En classification binaire, on cherche en fait une projection

$$y = \mathbf{w}^T \mathbf{x}$$

telle que le seuil  $y \geq -w_0$  sépare bien le plus d'entrées projetées possible

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** analyse discriminante linéaire

- **L'analyse discriminante linéaire** cherche plutôt à bien séparer la projection de moyennes, i.e. on maximise :

$$\mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1)$$

où  $m_k = \mathbf{w}^T \mathbf{m}_k$  et

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n,$$

$$\mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$$

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** variance intra-classe

- Jusqu'à maintenant, le problème est mal posé
  - il suffit d'augmenter  $w$  infiniment pour maximiser
  - on pourrait imposer que  $w$  soit de norme 1, mais ceci n'est pas entièrement satisfaisant
- En plus, on va tenter de réduire les **variances intra-classe** des entrées projetées

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** variance intra-classe, inter-classe

- On combine ces idées en maximisant plutôt

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

- On peut montrer que la solution est telle que

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

où la matrice de covariance intra-classe est

$$\mathbf{S}_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** variance intra-classe, inter-classe

- On combine ces idées en maximisant plutôt

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \quad \xrightarrow{\text{équivalent à une variance inter-classe}}$$

- On peut montrer que la solution est telle que

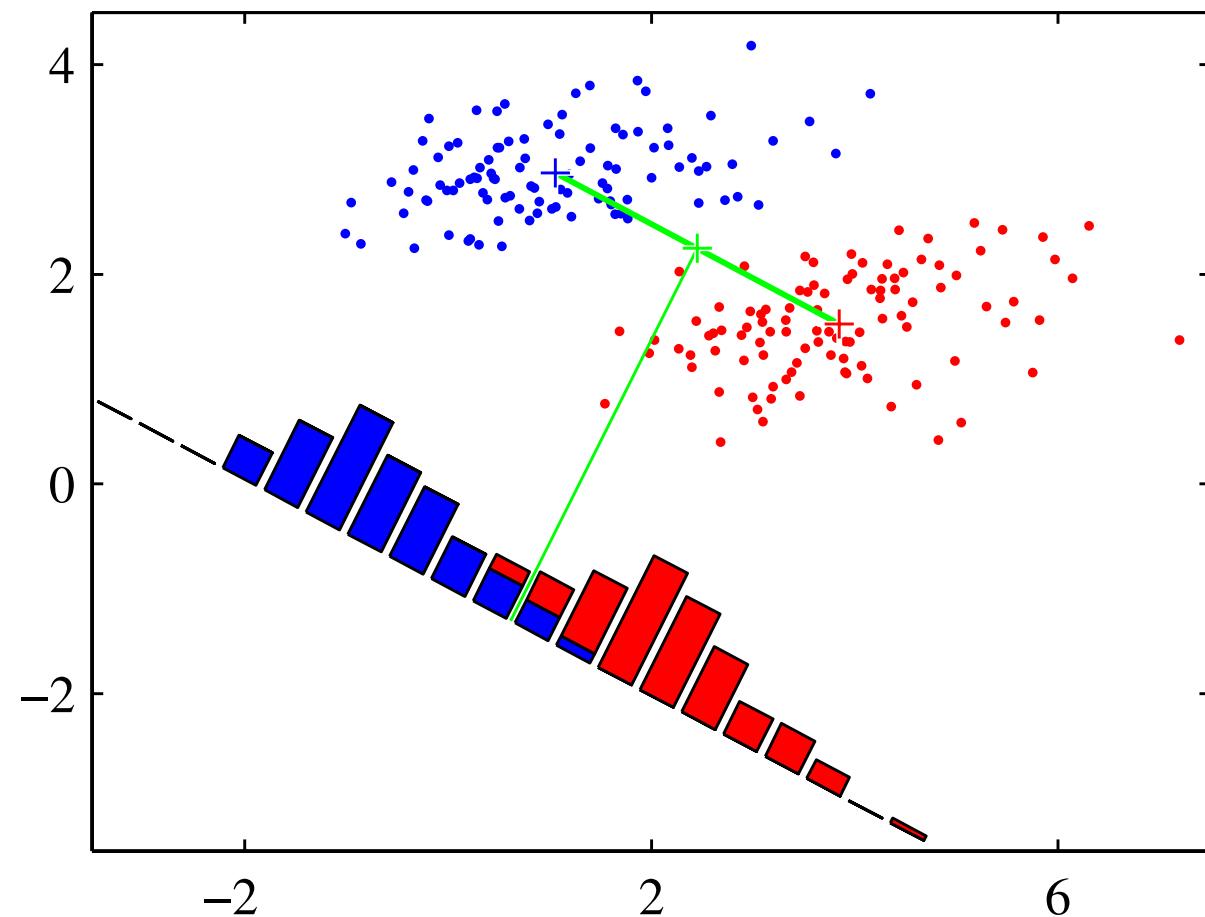
$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

où la matrice de covariance intra-classe est

$$\mathbf{S}_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

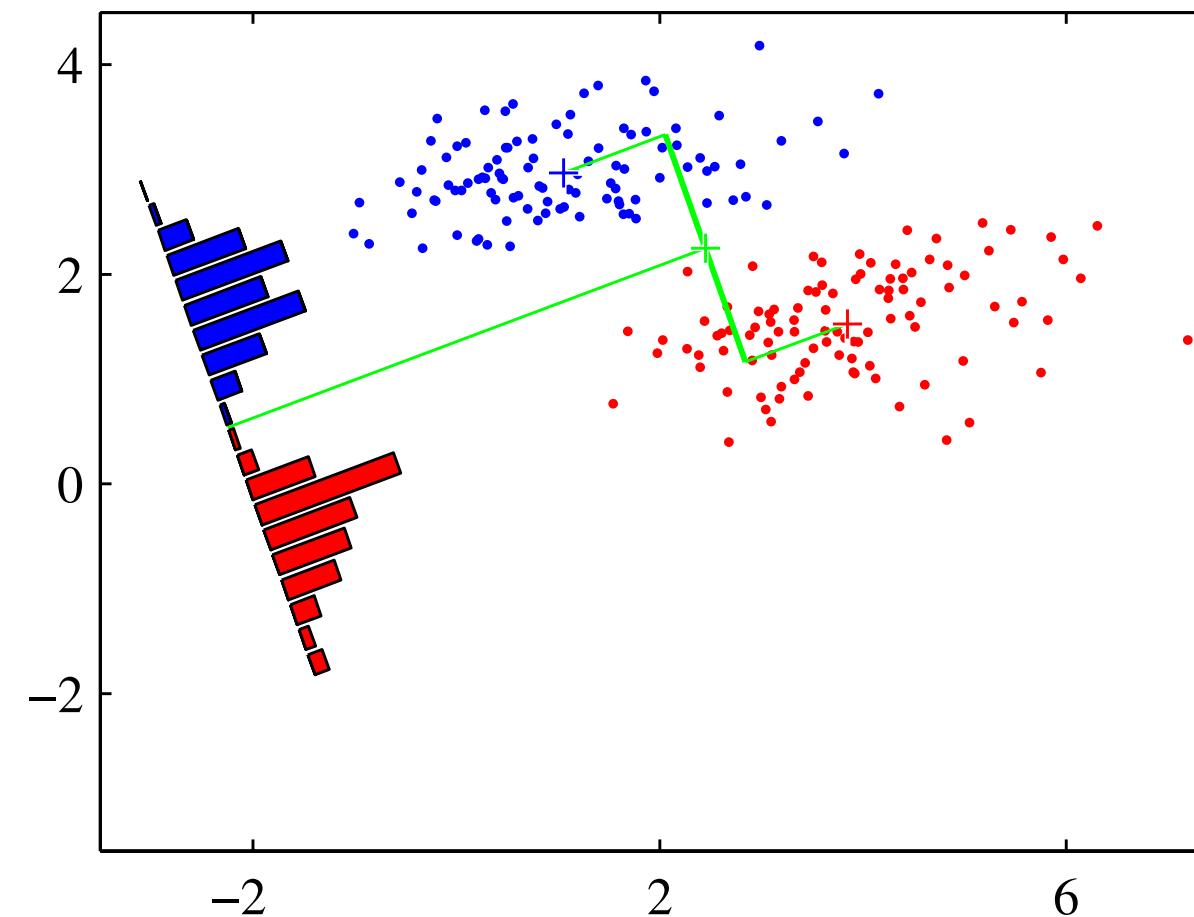
# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** analyse discriminante linéaire



Sans minimisation  
intra-classe

$$w \propto (m_1 - m_2)$$



Avec minimisation  
intra-classe

$$w \propto S_W^{-1}(m_1 - m_2)$$

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** analyse discriminante linéaire

- Une fois  $w$  calculé, il suffit de trouver un seuil de classification
  - un choix possible est  $(w^T m_1 + w^T m_2)/2$
- On peut voir l'analyse discriminante linéaire comme un cas particulier des moindres carrés
  - voir section 4.1.5
- Il est possible de généraliser au cas à plus de 2 classes
  - voir section 4.1.6

# Apprentissage automatique

Classification linéaire - approche probabiliste générative

# APPROCHE PROBABILISTE

**Sujets:** approche probabiliste

- Prenons plutôt une approche probabiliste
  - on suppose que nos données ont été générées d'un modèle probabiliste donné
  - on cherche les paramètres de ce modèle qui maximisent la vraisemblance des données d'entraînement
- Deux options :
  - **approche générative** : on choisit un modèle pour  $p(\mathbf{x}, t)$
  - **approche discriminante** : on choisit un modèle pour  $p(t|\mathbf{x})$

# APPROCHE PROBABILISTE

**Sujets:** approche probabiliste

- Prenons plutôt une approche probabiliste
  - on suppose que nos données ont été générées d'un modèle probabiliste donné
  - on cherche les paramètres de ce modèle qui maximisent la vraisemblance des données d'entraînement
- Deux options :
  - **approche générative** : on choisit un modèle pour  $p(\mathbf{x}, t)$
  - **approche discriminante** : on choisit un modèle pour  $p(t|\mathbf{x})$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** approche probabiliste générative

- On va supposer que les données ont été générées selon le processus suivant (cas binaire) :
  - pour  $n = 1 \dots N$ 
    - assigne  $t_n=1$  avec probabilité  $p(\mathcal{C}_1) = \pi$  et  $t_n=0$  avec probabilité  $p(\mathcal{C}_2) = 1 - \pi$
    - si  $t_n=1$ , génère  $\mathbf{x}_n$  de la loi de probabilité  $p(\mathbf{x}_n|\mathcal{C}_1) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$
    - sinon ( $t_n=0$ ), génère  $\mathbf{x}_n$  de la loi de probabilité  $p(\mathbf{x}_n|\mathcal{C}_2) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$
  - En mots : les entrées sont des échantillons d'une loi gaussienne, mais de moyennes différentes pour les différentes classes

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** maximum de vraisemblance

- La probabilité des données d'entraînement devient

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

- Pour entraîner le classifieur, on cherche les paramètres  $(\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$  maximise la (log-)vraisemblance

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** maximum de vraisemblance

- La probabilité des données d'entraînement devient

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

- Cas  $\pi$ : on prend le logarithme et garde les termes avec  $\pi$

$$\sum_{n=1}^N \{t_n \ln \pi + (1 - t_n) \ln(1 - \pi)\}$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

# Sujets: maximum de vraisemblance

- La probabilité des données d'entraînement devient

$$p(\mathbf{t}|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \mu_1, \Sigma)]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \mu_2, \Sigma)]^{1-t_n}$$

- Cas  $\pi$ : puis cherche le maximum en annulant la dérivée

$$\pi = \frac{1}{N} \sum_{n=1}^N t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

nb. de la classe  $\mathcal{C}_1$       nb. de la classe  $\mathcal{C}_2$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** maximum de vraisemblance

- La probabilité des données d'entraînement devient

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

- Cas  $\boldsymbol{\mu}_1$ : on prend le logarithme et garde les termes avec  $\boldsymbol{\mu}_1$

$$\sum_{n=1}^N t_n \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2} \sum_{n=1}^N t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{const}$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** maximum de vraisemblance

- La probabilité des données d'entraînement devient

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

- Cas  $\boldsymbol{\mu}_1$ : puis cherche le maximum en annulant la dérivée

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^N t_n \mathbf{x}_n$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** maximum de vraisemblance

- La probabilité des données d'entraînement devient

$$p(\mathbf{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

- Cas  $\boldsymbol{\mu}_2$ : de façon similaire, on obtient pour  $\boldsymbol{\mu}_2$

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^N (1 - t_n) \mathbf{x}_n$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** maximum de vraisemblance

- Cas  $\Sigma$  : plus compliqué à démontrer, mais on obtient

$$\Sigma = \frac{N_1}{N} \mathbf{S}_1 + \frac{N_2}{N} \mathbf{S}_2$$

covariance empirique de classe  $\mathcal{C}_1$

$$\mathbf{S}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^T$$

covariance empirique de classe  $\mathcal{C}_2$

$$\mathbf{S}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^T$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** règle de Bayes

- Une fois  $\pi, \mu_1, \mu_2, \Sigma$  calculés, on peut classifier de nouvelles entrées à l'aide de la règle de Bayes

$$p(\mathcal{C}_1 | \mathbf{x}) = \frac{p(\mathbf{x} | \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} | \mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x} | \mathcal{C}_2)p(\mathcal{C}_2)}$$

- Si  $p(\mathcal{C}_1 | \mathbf{x}) \geq 0.5$ , on classifie dans la classe  $\mathcal{C}_1$ , sinon on classifie dans la classe  $\mathcal{C}_2$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** fonction sigmoïde

- On peut aussi écrire

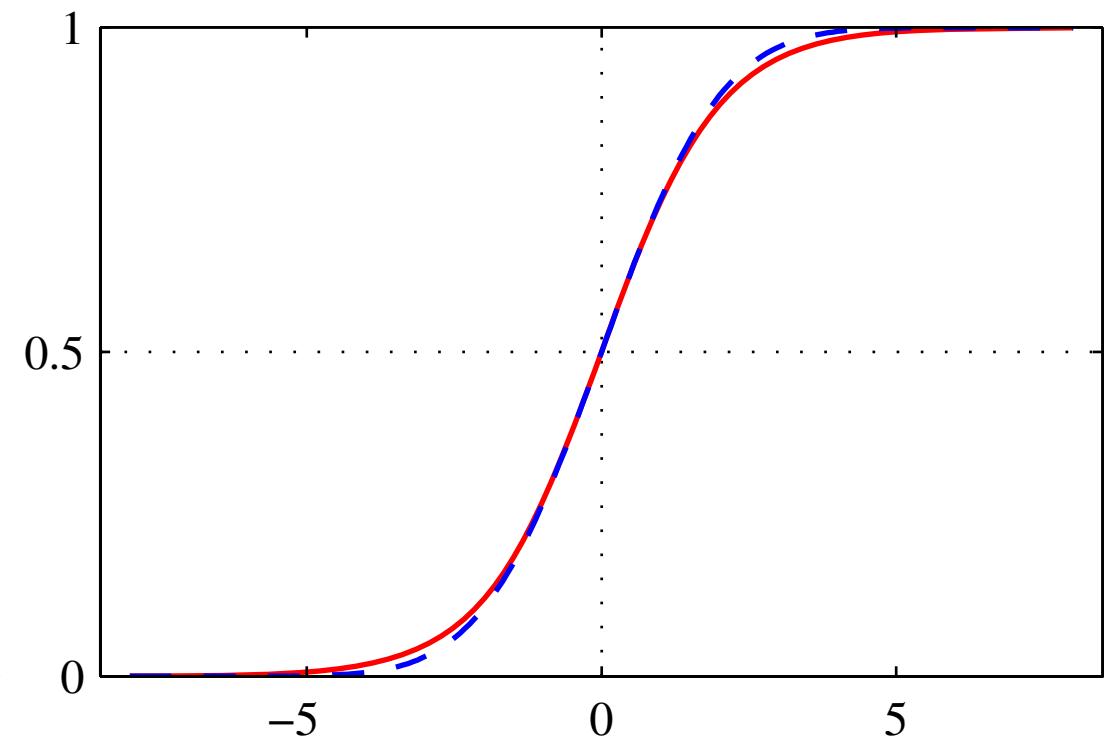
$$p(\mathcal{C}_1 | \mathbf{x}) = \frac{1}{1 + \exp(-a)} = \sigma(a)$$

où

$$a = \ln \frac{p(\mathbf{x} | \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} | \mathcal{C}_2)p(\mathcal{C}_2)}$$

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

**fonction sigmoïde**



# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** forme classification linéaire

- Dans le cas où  $p(\mathbf{x}_n|\mathcal{C}_1)$  et  $p(\mathbf{x}_n|\mathcal{C}_2)$  sont gaussiennes

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

où

$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T \Sigma^{-1} \boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T \Sigma^{-1} \boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** forme classification linéaire

- Dans le cas où  $p(\mathbf{x}_n|\mathcal{C}_1)$  et  $p(\mathbf{x}_n|\mathcal{C}_2)$  sont gaussiennes

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

où

$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$w_0 = -\frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

$$1 - \pi$$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** forme classification linéaire

- Puisque  $\sigma(0) = 0.5$ , alors la règle  $p(\mathcal{C}_1 | \mathbf{x}) \geq 0.5$  est équivalente à  $\mathbf{w}^T \mathbf{x} + w_0 \geq 0$
- On retrouve donc la forme d'un classifieur linéaire

# APPROCHE PROBABILISTE GÉNÉRATIVE

## Sujets: extensions

- On peut généraliser au cas à multiples classes
  - voir fin des sections 4.2 et 4.2.I
- On peut généraliser à des lois  $p(\mathbf{x}_n|\mathcal{C}_1)$  et  $p(\mathbf{x}_n|\mathcal{C}_2)$  autre que gaussiennes
  - observations binaires, voir section 4.2.3
  - cas général (famille exponentielle), voir section 4.2.4

# Apprentissage automatique

Classification linéaire - approche probabiliste discriminante

# APPROCHE PROBABILISTE

**Sujets:** approche probabiliste

- Prenons plutôt une approche probabiliste
  - on suppose que nos données ont été générées d'un modèle probabiliste donné
  - on cherche les paramètres de ce modèle qui maximisent la vraisemblance des données d'entraînement
- Deux options :
  - **approche générative** : on choisit un modèle pour  $p(\mathbf{x}, t)$
  - **approche discriminante** : on choisit un modèle pour  $p(t|\mathbf{x})$

# APPROCHE PROBABILISTE

**Sujets:** approche probabiliste

- Prenons plutôt une approche probabiliste
  - on suppose que nos données ont été générées d'un modèle probabiliste donné
  - on cherche les paramètres de ce modèle qui maximisent la vraisemblance des données d'entraînement
- Deux options :
  - **approche générative** : on choisit un modèle pour  $p(\mathbf{x}, t)$
  - **approche discriminante** : on choisit un modèle pour  $p(t|\mathbf{x})$

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** approche probabiliste discriminante, régression logistique

- Dans le cas génératif, on a vu que la probabilité de classifier dans la classe  $\mathcal{C}_1$  prend la forme

$$p(\mathcal{C}_1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

- Dans le cas discriminant, l'idée est d'utiliser directement cette forme comme modèle de  $p(t|\mathbf{x})$ 
  - plutôt que maximiser la probabilité jointe  $p(\mathbf{x}, t)$ , on maximise la probabilité conditionnelle  $p(t|\mathbf{x})$
  - on appelle ce modèle la **régression logistique**

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** maximum de vraisemblance

- Cas génératif : on cherche  $(\pi, \mu_1, \mu_2, \Sigma)$  qui maximise

$$p(\mathbf{t}|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \mu_1, \Sigma)]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \mu_2, \Sigma)]^{1-t_n}$$

- Cas discriminant : on cherche directement  $\mathbf{w}$  qui maximise

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

où  $y_n = p(\mathcal{C}_1 | \mathbf{x}_n)$

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** fonctions de bases

- On peut facilement remplacer la représentation de l'entrée à l'aide de fonctions de bases (comme pour la régression)

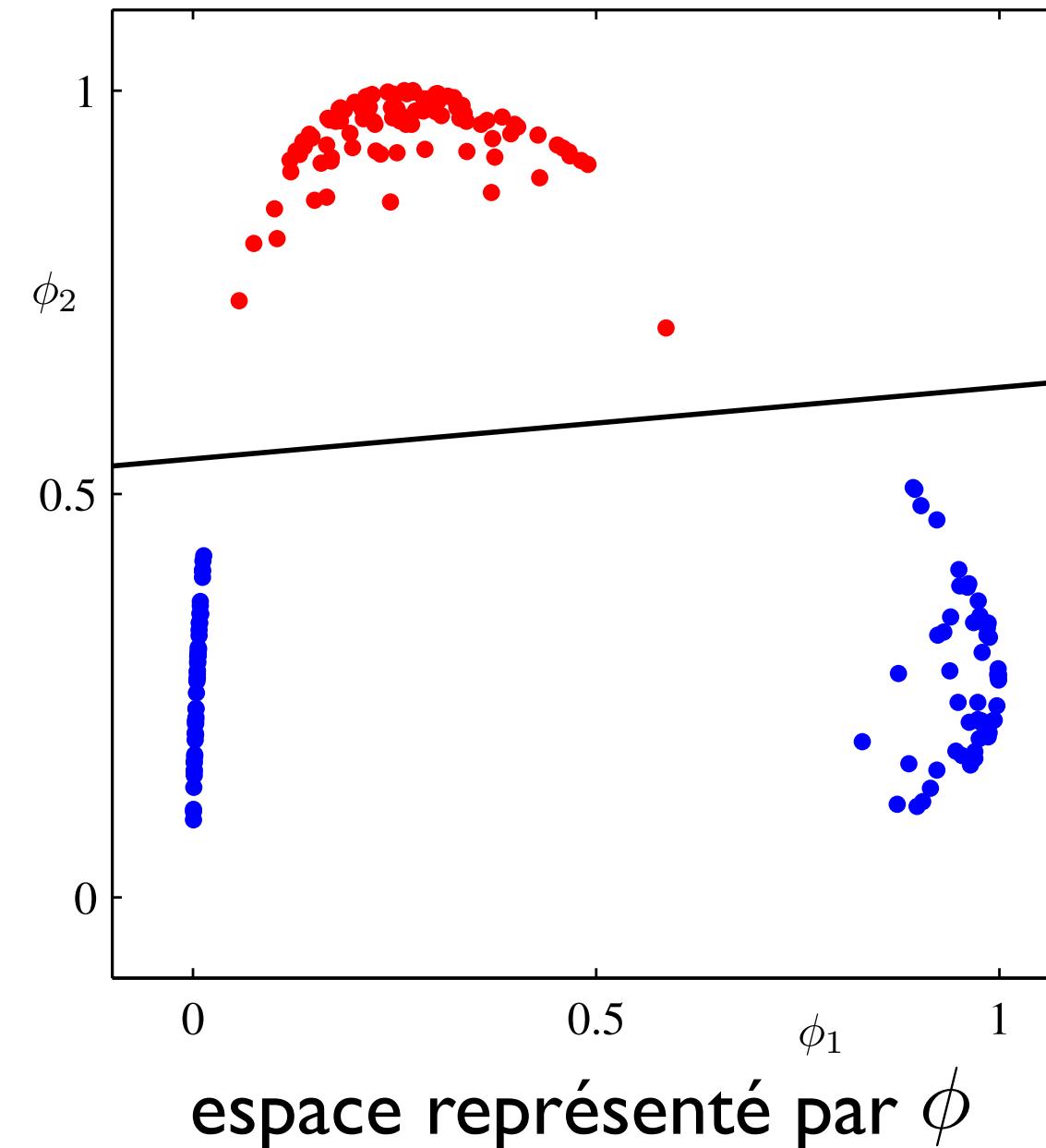
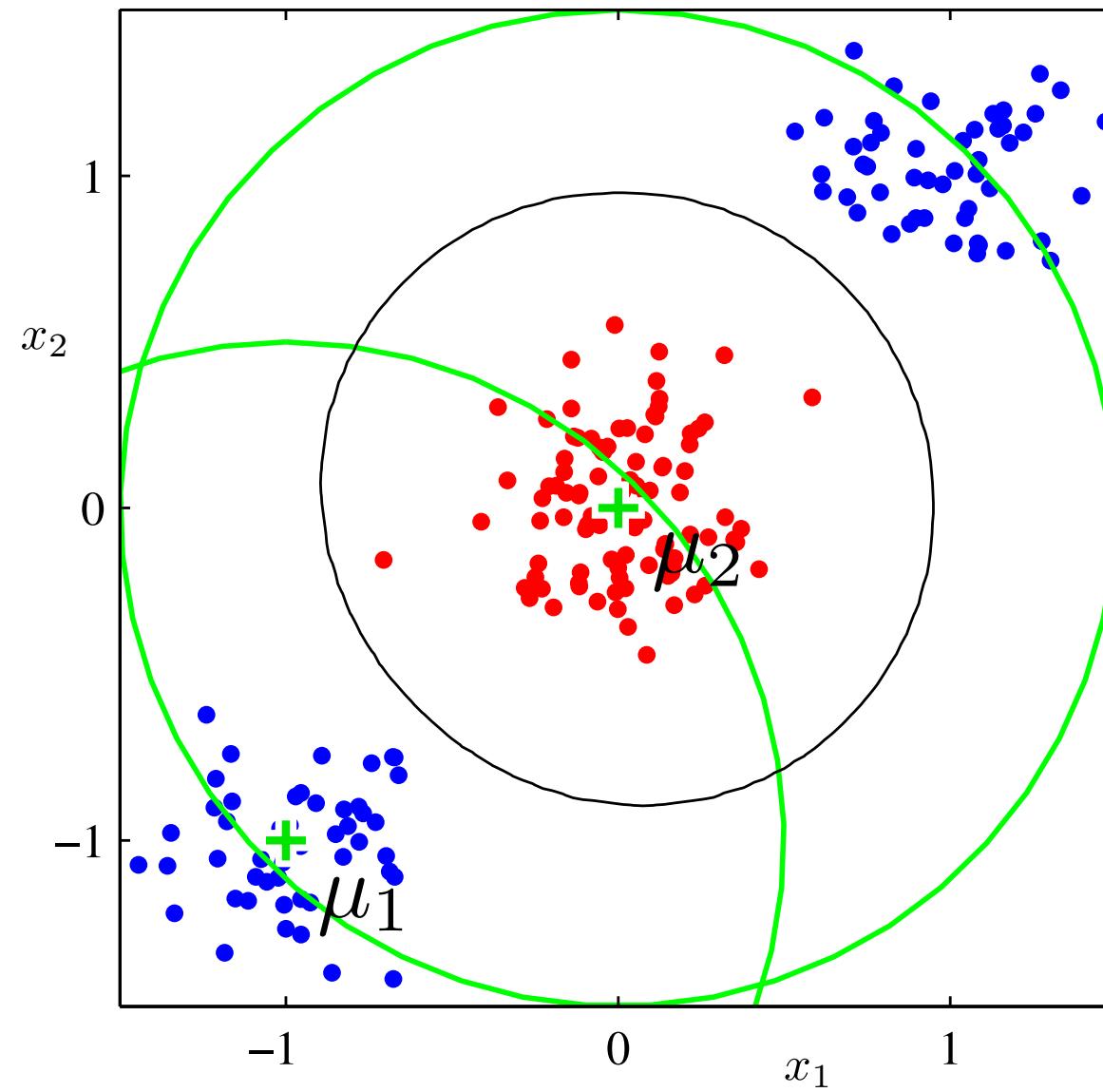
$$p(\mathcal{C}_1 | \phi) = y(\phi) = \sigma(\mathbf{w}^T \phi)$$

$\phi$  est équivalent à  $\phi(\mathbf{x})$

- Si les fonctions de bases sont non-linéaires, peut rendre les classes linéairement séparables

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** fonctions de bases



2 fonctions de bases gaussiennes  $\phi_j$

$$\exp \left\{ -\frac{(x - \mu_j)^2}{2s^2} \right\}$$

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** cross-entropie

- Maximiser la vraisemblance est équivalent à minimiser la log-vraisemblance négative

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = - \underbrace{\sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}}$$

**cross-entropie (binaire)**

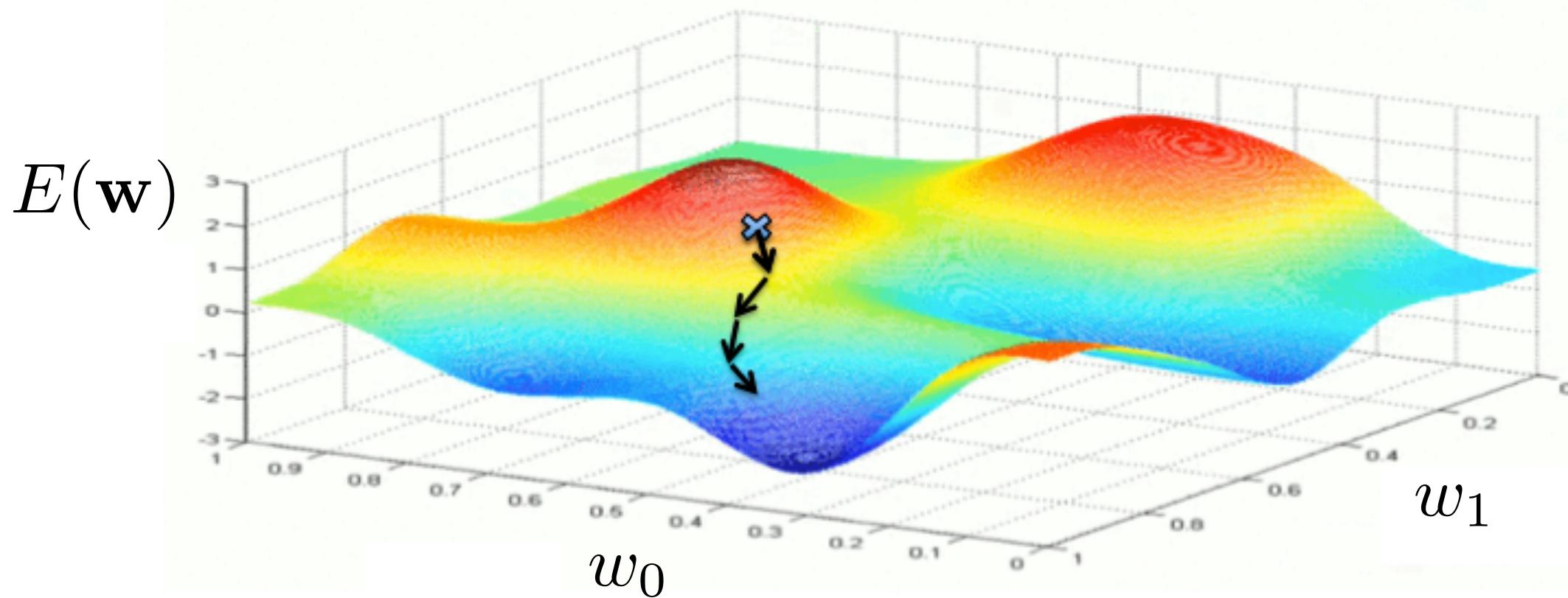
- Malheureusement, minimiser cette fonction ne se fait pas analytiquement
  - on va devoir trouver le minimum de façon numérique

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** descente de gradient

- **Descente de gradient**

- initialise la valeur de  $w$  aléatoirement
- durant  $I$  itérations
  - déplace  $w$  dans la direction opposée du gradient,  $w \leftarrow w - \eta \nabla E(w)$



# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** descente de gradient

- On peut montrer que le gradient est simplement

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n$$

où  $y_n = p(\mathcal{C}_1 | \phi_n)$

- Le nombre d'itérations ( $I$ ) de descente de gradient est un hyper-paramètre
  - on utilise un ensemble de validation pour déterminer quand arrêter

# APPROCHE PROBABILISTE DISCRIMINANTE

## Sujets: extensions

- La descente de gradient stochastique est souvent préférée en pratique, parce que plus rapide à converger
  - on met à jour  $w$  individuellement pour chaque exemple
  - voir section 3.1.3 (description pour la régression)
- On peut aussi généraliser au cas à multiples classes
  - voir section 4.3.4

# **Apprentissage automatique**

Classification linéaire - classification à multiples classes

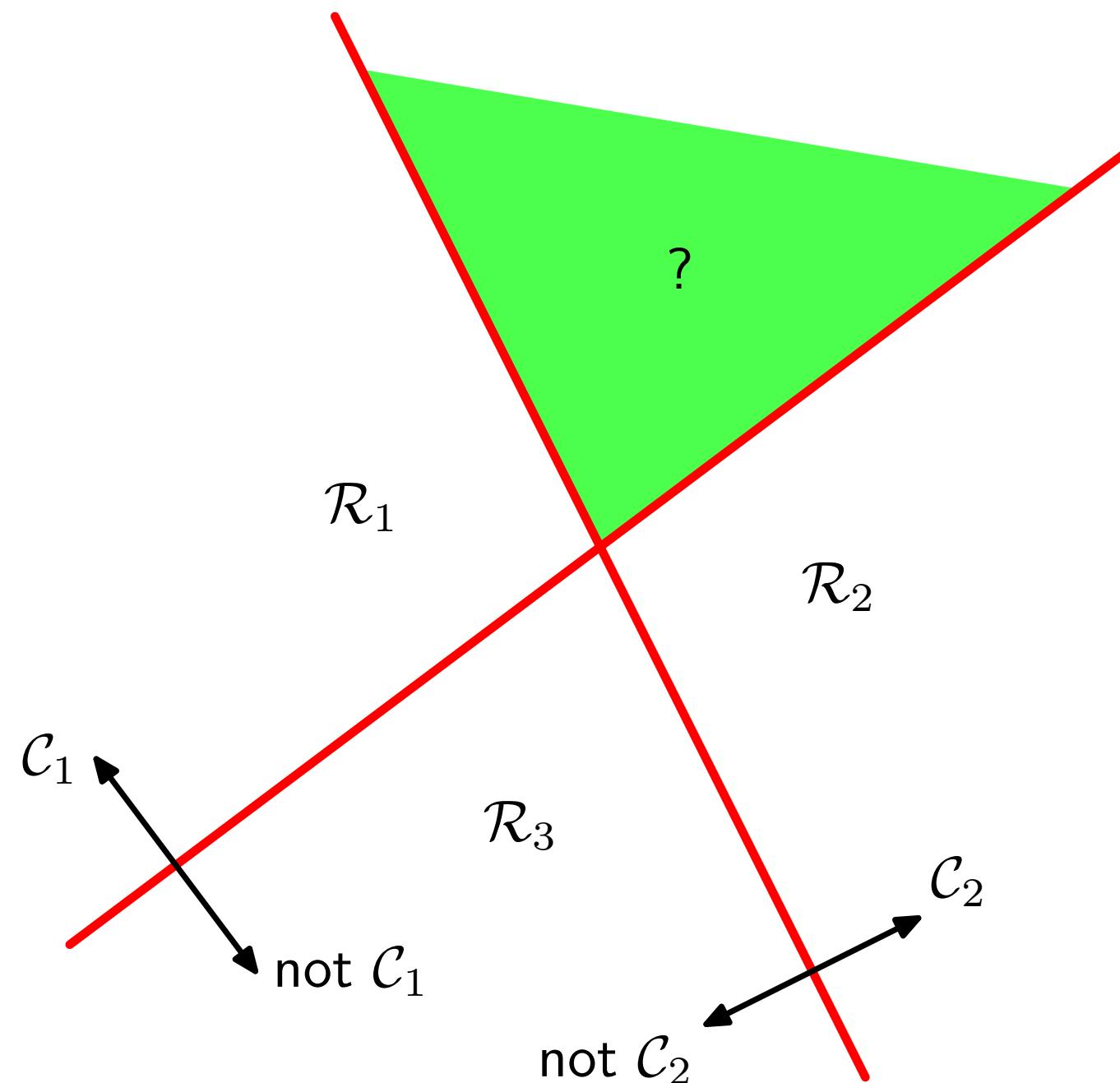
# CLASSIFICATION À MULTIPLES CLASSES

**Sujets:** classification à multiples classes

- Il est possible d'utiliser plusieurs classifieurs binaires pour résoudre un problème de classification à plus de 2 classes
- Approche *one-versus-rest* :
  - entraîne  $K-1$  classifieurs, chacun distinguant les entrées d'une classe vs. les entrées de toutes les autres classes
- Approche *one-versus-one* :
  - entraîne  $K(K-1)/2$  classifieurs, chacun distinguant les entrées d'une classe vs. les entrées d'une seule autre classe

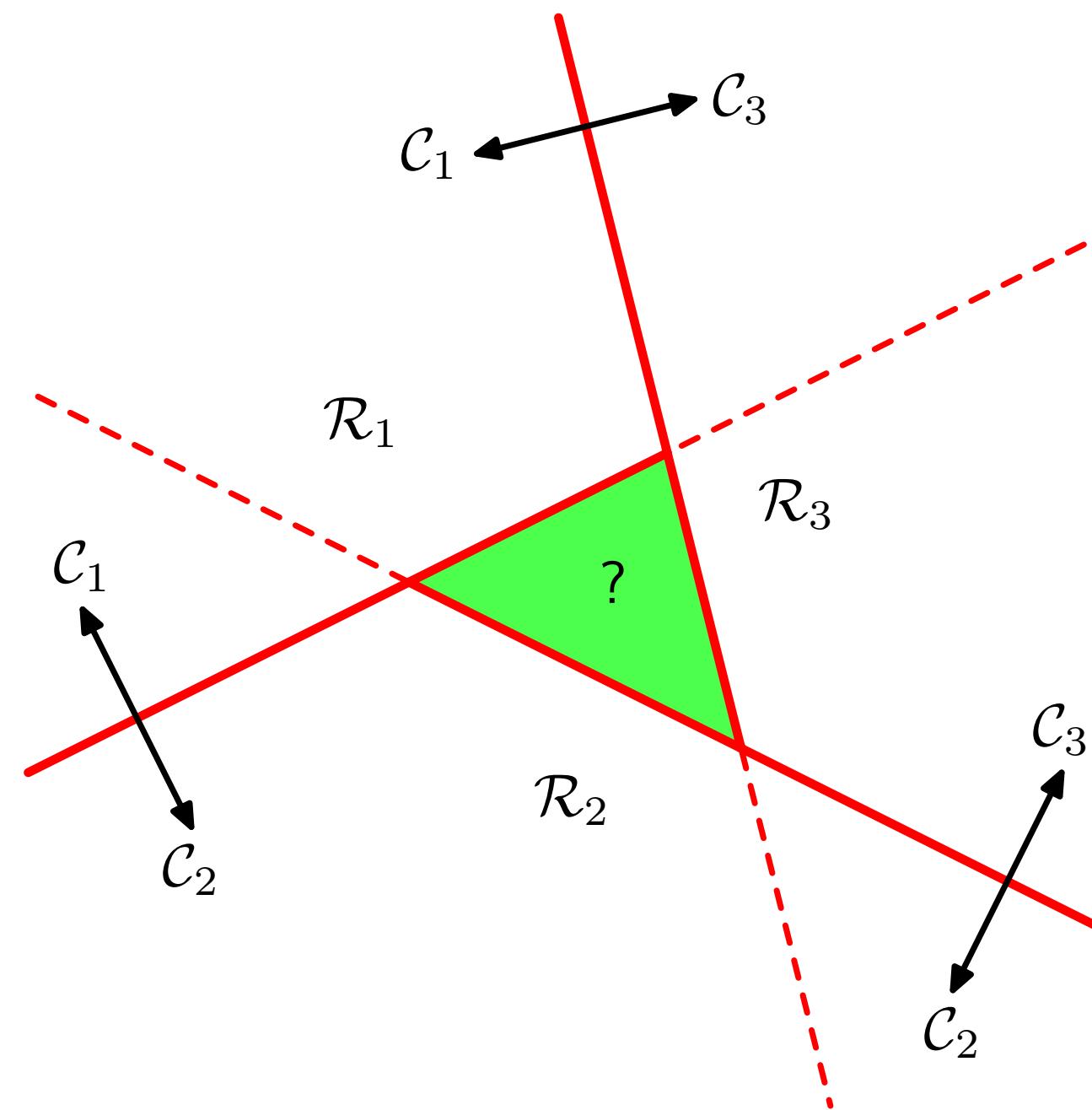
# CLASSIFICATION À MULTIPLES CLASSES

**Sujets:** *one-versus-rest*



# CLASSIFICATION À MULTIPLES CLASSES

**Sujets:** one-versus-one



# CLASSIFICATION À MULTIPLES CLASSES

**Sujets:** classification à multiples classes

- Il est possible de résoudre les ambiguïtés en pondérant les votes des classifieurs binaires
  - cas probabiliste : pondérer par la probabilité  $p(\mathcal{C}_1 | \mathbf{x})$
- L'idéal serait d'utiliser la version de l'algorithme adaptée à la classification à multiples classes directement

# Apprentissage automatique

Classification linéaire - résumé

# MÉTHODE DES MOINDRES CARRÉS

**Sujets:** résumé de la méthode des moindres carrés

- Modèle :  $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$$

- Entraînement :  $\mathbf{w} = (\lambda \mathbf{I} + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$  ( $t = 1$  vs.  $t = -1$ )  
(maximum de vraisemblance si  $\lambda=0$  ou maximum a posteriori si  $\lambda>0$ )
- Hyper-paramètre :  $\lambda$
- Prédiction :  $\mathcal{C}_1$  si  $y(\mathbf{x}, \mathbf{w}) \geq 0$ , sinon  $\mathcal{C}_2$

# ANALYSE DISCRIMINANTE LINÉAIRE

**Sujets:** résumé de l'analyse discriminante

- Modèle :  $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$$

- Entraînement :

$$\mathbf{w} \leftarrow \mathbf{S}_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

$$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

$$w_0 = (\mathbf{w}^T \mathbf{m}_1 + \mathbf{w}^T \mathbf{m}_2) / 2$$

- Prédiction :  $\mathcal{C}_1$  si  $y(\mathbf{x}, \mathbf{w}) \geq 0$ , sinon  $\mathcal{C}_2$

# APPROCHE PROBABILISTE GÉNÉRATIVE

**Sujets:** résumé de l'approche probabiliste générative

- Modèle :  $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$   
 $p(\mathbf{x}_n, \mathcal{C}_1) = \pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$   
 $p(\mathbf{x}_n, \mathcal{C}_2) = (1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$

$$\begin{aligned}\boldsymbol{\mu}_1 &= \frac{1}{N_1} \sum_{n=1}^N t_n \mathbf{x}_n & \boldsymbol{\mu}_2 &= \frac{1}{N_2} \sum_{n=1}^N (1 - t_n) \mathbf{x}_n \\ \boldsymbol{\Sigma} &= \frac{N_1}{N} \mathbf{S}_1 + \frac{N_2}{N} \mathbf{S}_2 & p(\mathcal{C}_1) &= \frac{N_1}{N} = 1 - p(\mathcal{C}_2)\end{aligned}$$

- Entraînement : ( $t = 1$  vs.  $t = 0$ )

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2} \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

- Prédiction :  $\mathcal{C}_1$  si  $y(\mathbf{x}, \mathbf{w}) \geq 0$ , sinon  $\mathcal{C}_2$

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** résumé de l'approche probabiliste discriminante (régression logistique)

- Modèle :  $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$

$$p(\mathcal{C}_1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

- Entraînement : descente de gradient  $(t = 1 \text{ vs. } t = 0)$

- initialise la valeur de  $\mathbf{w}$  aléatoirement

- durant  $I$  itérations

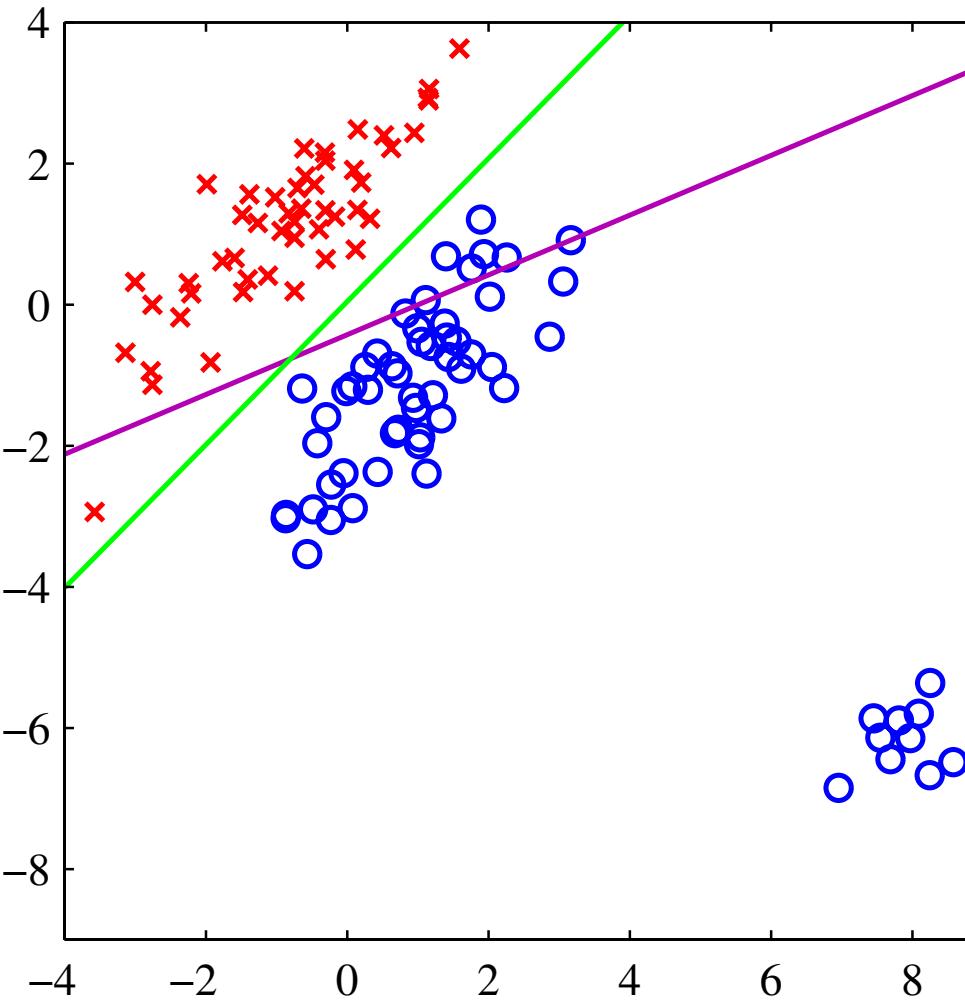
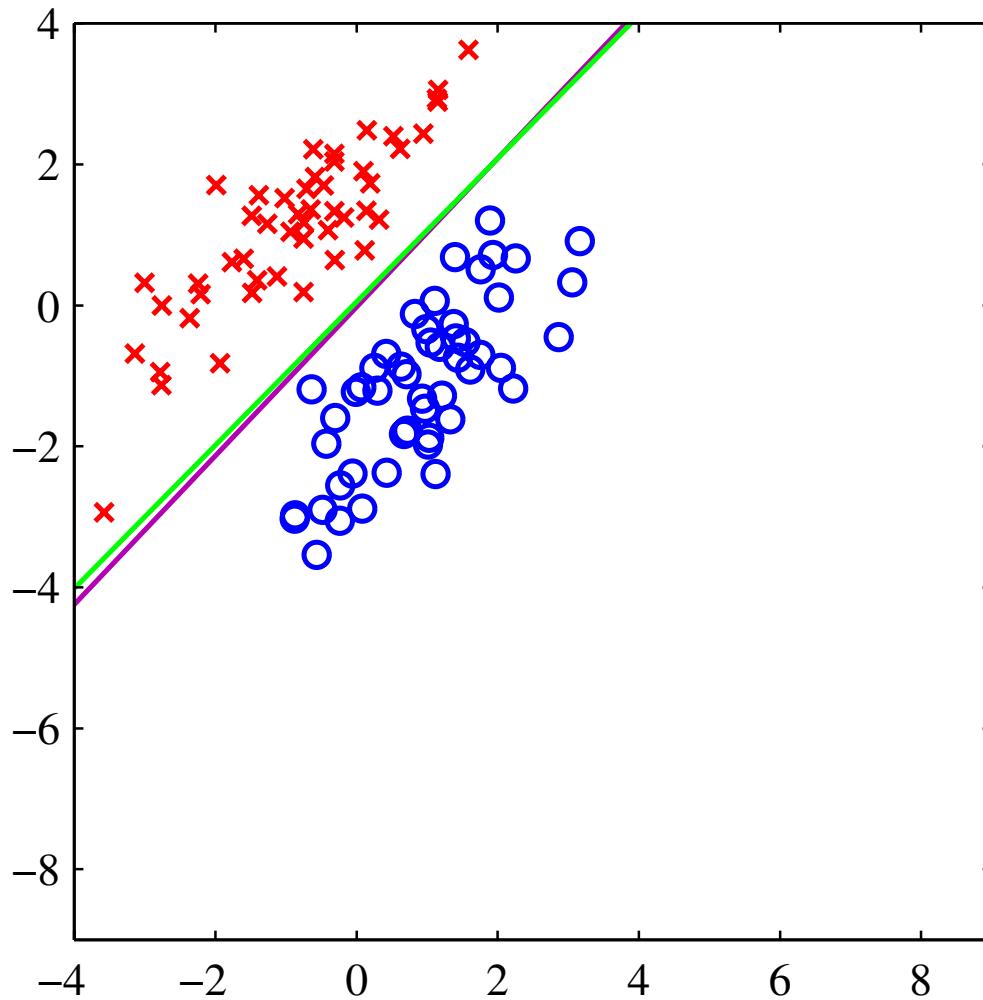
- $\mathbf{w} \leftarrow \mathbf{w} - \eta \sum_n (y_n - t_n) \phi_n$

- Prédiction :  $\mathcal{C}_1$  si  $y(\mathbf{x}, \mathbf{w}) \geq 0$ , sinon  $\mathcal{C}_2$

# MOINDRES CARRÉS VS. RÉGRESSION LOGISTIQUE

**Sujets:** moindres carrés vs. régression logistique

- Les résultats pourront être différents entre les algorithmes



— moindres carrés  
— rég. logistique