

Fondements de l'apprentissage machine

Automne 2014

Roland Memisevic

Leçon 3

Roland Memisevic Fondements de l'apprentissage machine

Classification linéaire

$$\mathbf{x} \rightarrow t$$

- ▶ Les sorties, t , sont discrètes et peuvent prendre K valeurs possibles $\mathcal{C}_1, \dots, \mathcal{C}_K$.
- ▶ Tâche : Apprendre à prédire t pour des nouveaux \mathbf{x} .
- ▶ Un problème d'apprentissage supervisé.
- ▶ Comme pour la régression linéaire, la linéarité rend la tâche simple.
- ▶ Mais «linéaire» veut dire autre chose que dans le cas de la régression linéaire.

Roland Memisevic Fondements de l'apprentissage machine

Plan

- ▶ Classification linéaire
- ▶ Modèles discriminatifs, régression logistique
- ▶ Modèles génératifs, Bayes naïf

Roland Memisevic Fondements de l'apprentissage machine

Discriminatif vs génératif

$$\mathbf{x} \rightarrow t$$

- ▶ Il existe principalement deux types de solutions.
- ▶ Les **méthodes discriminatives** essaient d'apprendre directement une fonction $y(\mathbf{x})$.
- ▶ Les **méthodes génératives** apprennent un modèle pour représenter chaque classe \mathcal{C}_k (par exemple, des distributions $p(\mathbf{x}|t)$). Puis, la classification revient à “inverser” ce modèle (par exemple, en calculant $p(t|\mathbf{x})$)

Roland Memisevic Fondements de l'apprentissage machine

Fonctions discriminantes

- ▶ Dans le cas $K = 2$, on peut définir :

Classification linéaire (deux classes)

- ▶ En utilisant la **fonction discriminante**

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

avec les paramètres \mathbf{w} , w_0 , attribuez \mathbf{x} à la classe \mathcal{C}_1 si $y(\mathbf{x}) \geq 0$, et à la classe \mathcal{C}_2 autrement.

- ▶ L'apprentissage revient à ajuster les paramètres \mathbf{w} et w_0 .

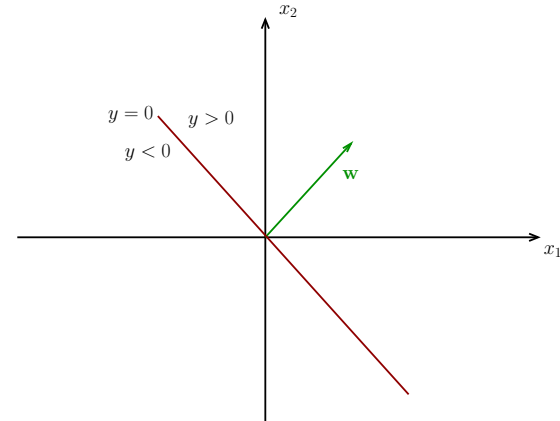
Le biais

- ▶ Le biais, w_0 , nous permet de déplacer la frontière de sorte qu'elle ne passe pas par l'origine.
- ▶ Comme dans le cas de la régression linéaire, nous pouvons l'éliminer formellement :

$$\begin{pmatrix} x_1 \\ \vdots \\ x_D \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{pmatrix}$$

- ▶ Ici (pour la classification), w_0 agit comme un **seuil** négatif.

Une interprétation géométrique



- ▶ Cette règle correspond à une frontière linéaire.
- ▶ En $D > 2$ dimensions, il s'agit d'un hyperplan de dimension $(D - 1)$.
- ▶ (Ici $w_0 = 0$)

Plusieurs classes

- ▶ Pour le cas $K \geq 2$, on peut définir :

Classification linéaire multi-classe

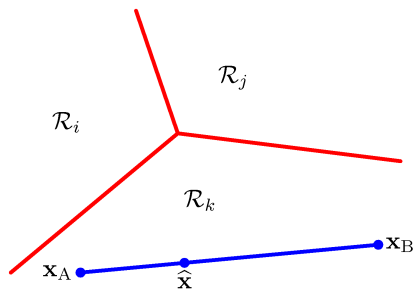
- ▶ En utilisant K **fonctions discriminantes**

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$

avec les paramètres \mathbf{w}_k, w_{k0} , $k = 1 \dots K$, attribuez \mathbf{x} à la classe \mathcal{C}_k si $y_k(\mathbf{x}) > y_j(\mathbf{x}) \forall j \neq k$

- ▶ Nous pouvons aussi définir un modèle 2-classe de cette façon.

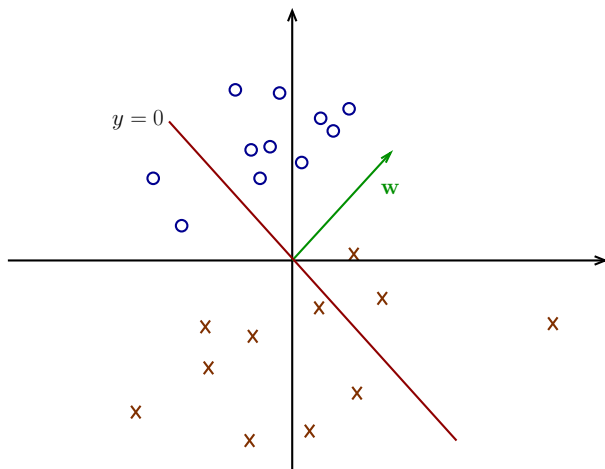
Plusieurs classes : une interprétation géométrique



- ▶ Le modèle divise l'espace en K régions convexes :
- ▶ Toute combinaison convexe de deux points dans la même classe sera également dans cette catégorie (Bishop, page 183)

Exemple

- ▶ S'il existe une frontière de classification qui ne fait aucune erreur sur l'ensemble d'entraînement, cet ensemble est appelé **linéairement séparable**.

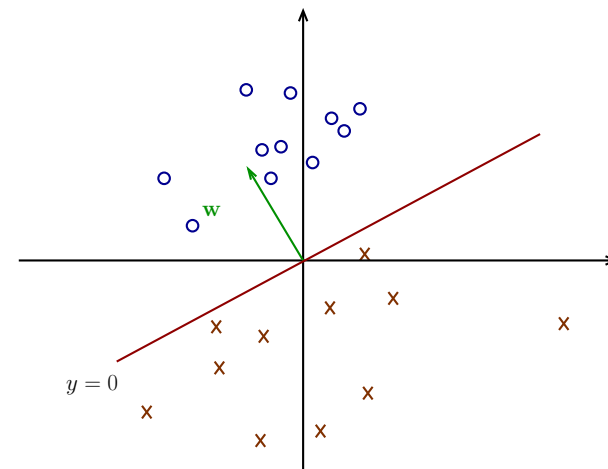


L'apprentissage

- ▶ L'objectif de l'apprentissage : en utilisant des données d'entraînement $\mathcal{D} = \{(\mathbf{x}_n, t_n)\}$, ajuster les paramètres de sorte que sur des données de test, le modèle fasse le moins d'erreurs de classification possible.
- ▶ Comme pour la régression linéaire, cela revient habituellement à l'optimisation d'un critère d'entraînement en gardant l'espace d'hypothèses petit (= en contraignant la flexibilité du modèle) pour prévenir le sur-apprentissage.

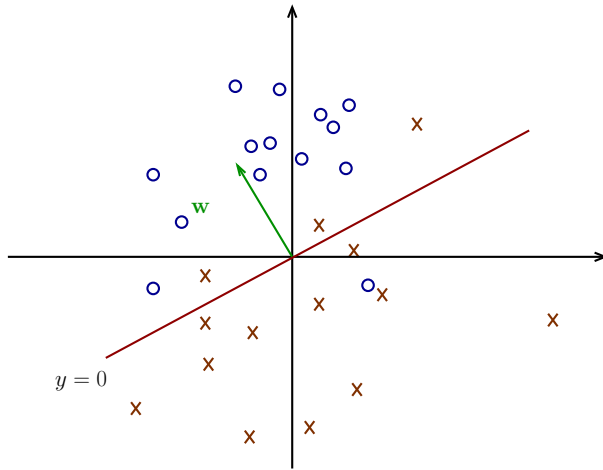
Exemple

- ▶ S'il existe une frontière de classification qui ne fait aucune erreur sur l'ensemble d'entraînement, cet ensemble est appelé **linéairement séparable**.



Exemple

- ▶ Un ensemble d'entraînement qui n'est pas linéairement séparable :



Représenter les valeurs discrètes

- ▶ Comme les sorties représentent des valeurs discrètes, il est utile d'utiliser le **codage orthogonal**.
- ▶ Nous utilisons la matrice \mathbf{T} pour représenter toutes les sorties de l'ensemble d'entraînement (une par rangée).
- ▶ Par conséquent, une valeur $t_{nk} = 1$ signifie que le n-ième exemple d'entraînement appartient à la classe K.

L'apprentissage de classifieurs linéaires

- ▶ Dans ce qui suit, nous empilons les vecteurs de paramètres \mathbf{w}_k côte à côte dans la matrice \mathbf{W} .
- ▶ Pour un cas d'entraînement \mathbf{x} ,

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

représente un vecteur de "scores" $\mathbf{w}_k^T \mathbf{x}$ (un pour chaque classe).

- ▶ Le critère d'apprentissage doit ajuster \mathbf{W} , de sorte que pour les exemples d'entraînement appartenant à la classe \mathcal{C}_k , le modèle produise de grandes valeurs $\mathbf{w}_k^T \mathbf{x}_n$ et de petites valeurs $\mathbf{w}_j^T \mathbf{x}_n \forall j \neq k$.

Régression logistique multinomiale

- ▶ La régression logistique est probablement le classifieur le plus ancien, mais est quand même l'un des plus utilisés.
- ▶ Il définit un modèle probabiliste sur t sachant \mathbf{x} :

Régression logistique multinomiale ("softmax regression")

$$p(\mathcal{C}_k | \mathbf{x}) = \frac{\exp(\mathbf{w}_k^T \mathbf{x}_n)}{\sum_{j=1}^K \exp(\mathbf{w}_j^T \mathbf{x}_n)}$$

- ▶ Explication : Le \exp assure la positivité de sorties, et la normalisation que leur somme soit égale à 1.

Maximum de vraisemblance pour la régression logistique multinomiale

- ▶ Comme dans le cas de la régression linéaire, nous pouvons minimiser la log-vraisemblance négative lors de l'entraînement.
- ▶ On obtient la fonction de coût :

$$\begin{aligned} E(\mathbf{W}; \mathcal{D}) &= -\log \prod_n p(\mathbf{t}_n | \mathbf{x}_n) \\ &= -\log \prod_n \prod_k p(C_k | \mathbf{x}_n)^{t_{nk}} \\ &= -\sum_n \sum_k t_{nk} \log p(C_k | \mathbf{x}_n) \\ &= -\sum_{nk} t_{nk} (\mathbf{w}_k^T \mathbf{x}_n - \log \sum_{j=1}^K \exp(\mathbf{w}_j^T \mathbf{x}_n)) \end{aligned}$$

Roland Memisevic Fondements de l'apprentissage machine

L'astuce "logsumexp"

- ▶ Les expressions comme

$$\frac{\exp(\mathbf{w}_k^T \mathbf{x}_n)}{\sum_{j=1}^K \exp(\mathbf{w}_j^T \mathbf{x}_n)}$$

sont très instables en pratique parce que les "exp" du dénominateur peuvent provoquer un débordement.

- ▶ Il ne faut jamais utiliser des expressions comme " $\sum_i \exp(a_i)$ " naïvement dans le code.
- ▶ Solution : ajouter une constante A à l'argument de chaque "exp" pour que même les plus grands arguments deviennent petits; annuler l'opération après avoir calculé la somme.

Roland Memisevic Fondements de l'apprentissage machine

Maximum de vraisemblance pour la régression logistique multinomiale

- ▶ Contrairement à la la régression linéaire, il n'y a pas de solution de forme fermée pour \mathbf{W} .
- ▶ Mais on peut utiliser l'optimisation basée sur le gradient (par exemple SGD) pour minimiser $E(\mathbf{W}; \mathcal{D})$.
- ▶ Le gradient par rapport au chaque vecteur \mathbf{w}_k est

$$\begin{aligned} \frac{\partial E(\mathbf{W}; \mathcal{D})}{\partial \mathbf{w}_k} &= -\sum_n t_{nk} \mathbf{x}_n - \frac{\exp(\mathbf{w}_k^T \mathbf{x}_n)}{\sum_j \exp(\mathbf{w}_j^T \mathbf{x}_n)} \mathbf{x}_n \\ &= \sum_n (p(C_k | \mathbf{x}_n) - t_{nk}) \mathbf{x}_n \end{aligned}$$

- ▶ On peut montrer que $E(\mathbf{W}; \mathcal{D})$ est convexe. Ainsi, il n'y a pas de minima non globaux.

Roland Memisevic Fondements de l'apprentissage machine

L'astuce "logsumexp"

- ▶ De nombreux logiciels fournissent une fonction "logsumexp" à cette fin

logsumexp

$$\begin{aligned} \text{logsumexp}(a_1, \dots, a_K) &= \log(\sum_i \exp(a_i + A)) - A \\ &\text{with } A = -(\max_i a_i) \end{aligned}$$

- ▶ D'autres instabilités numériques peuvent très souvent être résolues par la transformation de l'expression instable en une expression contenant $\sum_i \exp(a_i)$, puis en utilisant logsumexp.

Roland Memisevic Fondements de l'apprentissage machine

Régression logistique binaire

- ▶ Traditionnellement, le terme “régression logistique” a été utilisé pour se référer à des modèles à deux classes.
- ▶ Ces modèles ont été définis à l'aide d'un seul vecteur de paramètres \mathbf{w} :

Régression logistique binaire

$$p(\mathcal{C}_1|\mathbf{x}) = 1 - p(\mathcal{C}_2|\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

- ▶ Cela revient à retirer le surparamétrage dans les définitions précédentes, qui est dû à la contrainte $\sum_i p(\mathcal{C}_i|\mathbf{x}) = 1$.

La fonction sigmoïde

- ▶ La fonction

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

est une fonction en forme de "S", appelé “sigmoïde” ou “fonction logistique”. (Raison pour laquelle on dit “régression logistique”).

- ▶ Sa dérivée peut s'écrire :

$$\frac{d\sigma(a)}{da} = \sigma(1 - \sigma)$$

- ▶ Son inverse est $a = \log\left(\frac{\sigma}{1-\sigma}\right)$ (“fonction logit”)

Régression logistique binaire

- ▶ Pour la classe 1, nous pouvons écrire :

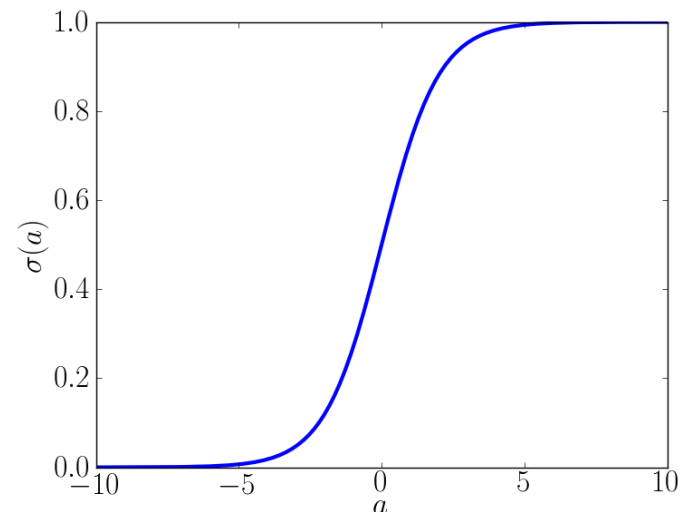
$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{\exp(\mathbf{w}_1^T\mathbf{x})}{\exp(\mathbf{w}_1^T\mathbf{x}) + \exp(\mathbf{w}_2^T\mathbf{x})} \\ &= \frac{1}{1 + \exp(-(\mathbf{w}_1 - \mathbf{w}_2)^T\mathbf{x})} \end{aligned}$$

- ▶ De même, pour la classe 2 :

$$p(\mathcal{C}_2|\mathbf{x}) = \frac{1}{1 + \exp((\mathbf{w}_1 - \mathbf{w}_2)^T\mathbf{x})}$$

- ▶ Donc, la définition sur la diapo précédente est équivalente au modèle multi-classe en utilisant : $\mathbf{w} := (\mathbf{w}_1 - \mathbf{w}_2)$
- ▶ On pourrait retirer le surparamétrage également pour les modèles multi-classe, mais cela n'est pas courant.

La fonction sigmoïde



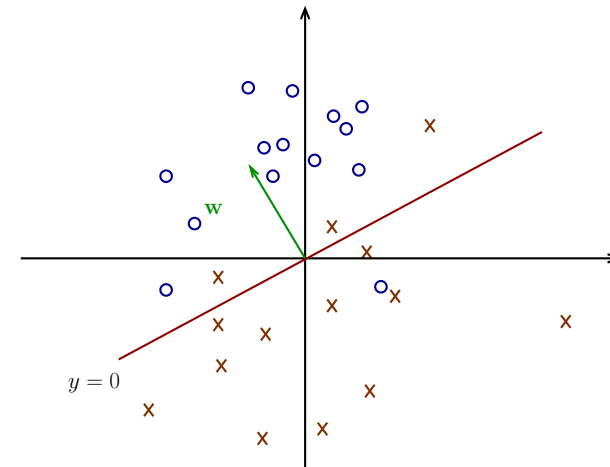
L'expansion de base

- ▶ Comme pour la régression linéaire, nous pouvons utiliser une expansion de base non linéaire $\mathbf{x} \rightarrow \Phi(\mathbf{x})$.
- ▶ Cela conduit à des frontières non linéaires entre les classes.
- ▶ Mais peut causer du sur-apprentissage.

Roland Memisevic

Fondements de l'apprentissage machine

L'expansion de base

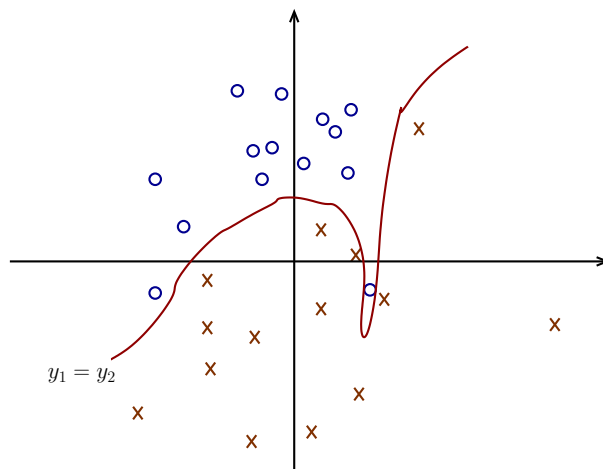


- ▶ Un modèle linéaire.

Roland Memisevic

Fondements de l'apprentissage machine

L'expansion de base



- ▶ Un modèle non linéaire.

Roland Memisevic

Fondements de l'apprentissage machine

Régularisation

- ▶ Comme pour la régression linéaire, on peut régulariser le modèle en ajoutant à l'objectif une pénalité telle que

$$E_{\mathbf{W}}(\mathbf{W}, \lambda) = \frac{\lambda}{2} \|\mathbf{W}\|^2$$

pour prévenir le sur-apprentissage.

Roland Memisevic

Fondements de l'apprentissage machine

Méthodes génératives

- ▶ Les **méthodes génératives** entraînent un modèle probabiliste $p(\mathbf{x}|\mathcal{C}_k)$ pour chaque classe.
- ▶ Pour faire la classification, il faut inverser cette relation, ce qui peut être fait en utilisant la règle de Bayes :

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

- ▶ Comme les classifieurs génératifs impliquent la règle de Bayes, ils sont parfois appelés “classifieur de Bayes” (mais ils ne sont pas vraiment des modèles bayésiens).

Méthodes génératives

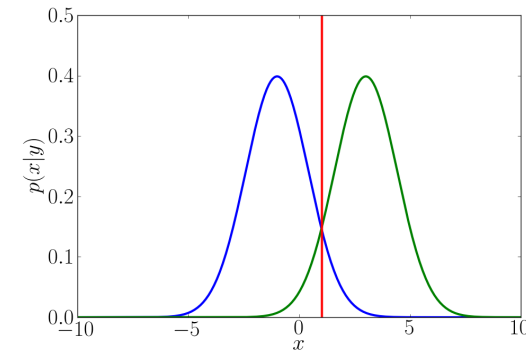
- ▶ Pour définir un classifieur génératif en pratique, il faut préciser deux quantités :

1. Les probabilités a priori $p(\mathcal{C}_k)$. En pratique, on utilise souvent

$$\frac{\sum_n t_{nk}}{N}$$

2. Les distributions $p(\mathbf{x}|\mathcal{C}_k)$. Il y a beaucoup de choix possibles.
- ▶ Notez que la régression logistique et les modèles génératifs définissent un classifieur en modélisant la distribution conditionnelle $p(\mathcal{C}_k|\mathbf{x})$. Ils ne diffèrent que *par la façon* dont ils définissent cette probabilité.

Exemple d'une méthode générative de dimension 1



- ▶ Les frontières entre les classes \mathcal{C}_1 et \mathcal{C}_2 peuvent être définies par :

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

Modèles génératifs avec des entrées réelles

- ▶ Pour des entrées réelles (de dimension D), un choix très commun pour la distribution $p(\mathbf{x}|\mathcal{C}_k)$ est la gaussienne :

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

- ▶ Normalement, chaque classe a sa propre moyenne. La matrice de covariance est parfois partagée entre les classes :

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

Estimation de paramètres

- ▶ Pour les modèles sans partage de la matrice de covariance, les estimations du maximum de vraisemblance sont

$$\boldsymbol{\mu}_k = \frac{1}{\sum_n t_{nk}} \sum_{n=1}^N t_{nk} \mathbf{x}_n$$

et

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n t_{nk}} \sum_{n=1}^N t_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

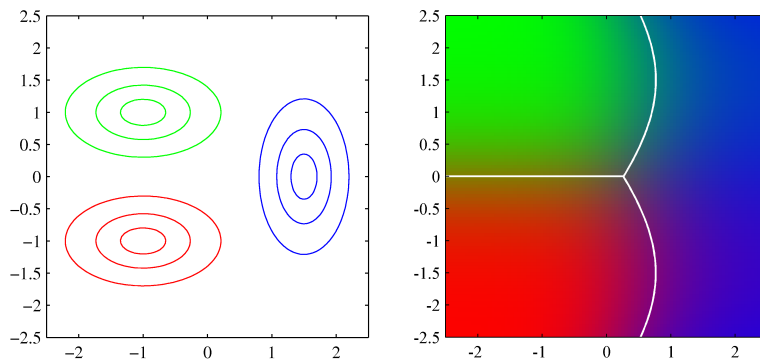
- ▶ Pour les modèles qui partagent la matrice de covariance, l'estimation est

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T$$

où $\boldsymbol{\mu}$ est la moyenne sur tous les exemples d'entraînement.

Frontières entre les classes : exemple

- ▶ Exemple d'un classifieur génératif gaussien à trois classes, où deux classes partagent une matrice de covariance :



Frontières entre les classes

- ▶ Pour les modèles qui partagent la matrice de covariance, les frontières satisfont la condition :

$$\begin{aligned} \log p(\mathbf{x}|\mathcal{C}_1) + \log p(\mathcal{C}_1) &= \log p(\mathbf{x}|\mathcal{C}_2) + \log p(\mathcal{C}_2) \\ \Leftrightarrow (\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_1) + \log p(\mathcal{C}_1) &= (\mathbf{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_2) + \log p(\mathcal{C}_2) \\ \Leftrightarrow \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - 2\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \log p(\mathcal{C}_1) &= \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 - 2\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \log p(\mathcal{C}_2) \\ \Leftrightarrow (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} \mathbf{x} &= \text{const} \end{aligned}$$

- ▶ Ceci est une condition **linéaire** pour les entrées \mathbf{x} .
- ▶ (On peut montrer que pour les modèles sans partage, les frontières sont des quadratiques.)

Classifieur génératif gaussien et régression logistique

- ▶ Pour un classifieur génératif gaussien à deux classes :

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} \end{aligned}$$

- ▶ Un petit calcul montre que

$$a = \log \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} \mathbf{x} := \mathbf{w}^T \mathbf{x}$$

- ▶ Le classifieur génératif gaussien avec une matrice de covariance commune a la même forme fonctionnelle que la régression logistique.

Modèles génératifs avec des entrées discrètes

- ▶ Des variables d'entrées discrètes sont faciles à traiter si nous supposons que les dimensions d'entrée sont *conditionnellement indépendantes, sachant la classe*.
- ▶ Des classifieurs génératifs basés sur cette hypothèse sont connus comme des **classifieurs de Bayes naïf**.
- ▶ Bien que l'indépendance conditionnelle semble être une restriction majeure en pratique, cela fonctionne souvent étonnamment bien.

Bayes naïf : Bernoulli

- ▶ Si nous utilisons des variables Bernoulli indépendantes pour chaque dimension d'entrée, nous pouvons écrire :

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^D \mu_{ki}^{x_i} (1 - \mu_{ki})^{1-x_i}$$

où $\mu_{ki} = p(x_i = 1|\mathcal{C}_k)$

- ▶ Les estimations du maximum de vraisemblance pour μ_{ki} sont :

$$\mu_{ki} = \frac{\sum_n t_{nk} x_{ni}}{\sum_n t_{nk}}$$

Bayes naïf discret (multinoulli)

- ▶ Si nous utilisons des variables discrètes pour chaque dimension d'entrée, nous pouvons écrire :

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^D \prod_{j=1}^M \mu_{kij}^{x_{ij}}$$

où $\mu_{kij} = p(x_i = j|\mathcal{C}_k)$.

- ▶ Les estimations du maximum de vraisemblance pour μ_{kij} sont :

$$\mu_{kij} = \frac{\sum_n t_{nk} x_{nij}}{\sum_n t_{nk}}$$

- ▶ Comment le savons-nous ? Maximiser la log-vraisemblance, imposer des contraintes $\sum_j \mu_{kij} = 1$ à l'aide de multiplicateurs de Lagrange.