Advanced Probabilistic Machine Learning
SSY316

Linear Models for Classification

Alexandre Graell i Amat
alexandre.graell@chalmers.se
https://sites.google.com/site/agraellamat

November 9 and 10, 2023

CHALMERS

Classification

Goal: Given a training set $\mathcal{D} = \{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$, with $t \in \{C_1, \ldots, C_K\}$, assign (classify) a new example $\boldsymbol{x}$ to one of the classes C_i.

- For convenience: $t \in \{1, \ldots, K\}$

Classification divides input space into decision regions $\mathcal{R}_i$, $i \in [K]$ (points in $\mathcal{R}_i$ assigned to class C_i).

Linear models: Decision boundaries (surfaces) are linear functions of $\boldsymbol{x}$ ($(d-1)$-dimensional hyperplanes).

Classification: Examples

Data set: annual income, credit card balance, whether or not the person has defaulted

Aim: Predict whether a person will default or not

Data set: Blood tests, X-rays, MRI scans

Aim: Predict whether a person has cancer or not

Classification as a supervised learning problem

Binary classification:

$$\begin{aligned} \boldsymbol{x} &\in C_1 \quad \text{if } t = 1 \\ \boldsymbol{x} &\in C_2 \quad \text{if } t = 0 \text{ (or } t = -1) \end{aligned}$$

e.g., $C_1 \equiv$ cancer, $C_2 \equiv$ no cancer

Multi-class classification ($K > 2$):

Label a point $\boldsymbol{x} \in C_k$ with a $K \times 1$ one-hot vector $\boldsymbol{t}$ with a single one at position k and zeroes elsewhere

$K = 5$, class C_2:

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

Three distinct modeling approaches

Discriminative deterministic models: Model directly the deterministic mapping between $\boldsymbol{x}$ and t via a parametrized function $\mu(\boldsymbol{x}, \boldsymbol{w})$ (discriminant function).

More powerful approach: model $p(\mathcal{C}_k|\boldsymbol{x})$ in an inference stage, then use this distribution to make optimal decisions.

Discriminative probabilistic models: Model $p(C_k|\boldsymbol{x})$ via a parametrized model.

Generative probabilistic models: Model $p(\boldsymbol{x}, t)$ specifying $p(t)$ ($p(C_k)$) and $p(\boldsymbol{x}|t)$ ($p(\boldsymbol{x}|C_k)$). Then,

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{p(\boldsymbol{x})}$$

Why a probabilistic approach?

Compensating for class priors (Cancer):

- Assume prevalence $1/1000$
- Naïve classifier that assigns every new case result to "no cancer" class: 99.9% accuracy!
- An unbalanced data set will lead to learning model that likely won't generalize well
- A balanced data set will lead to a more accurate model ... but we need to compensate for modification to training data!

Combining models (Cancer):

- 3 models based on blood tests, X-rays, MRIs
- If models provide posterior probabilities, they can be properly combined,

$$\begin{aligned} p(C_k|\boldsymbol{x}_\mathsf{b}, \boldsymbol{x}_\mathsf{x}, \boldsymbol{x}_\mathsf{m}) &\propto p(\boldsymbol{x}_\mathsf{b}, \boldsymbol{x}_\mathsf{x}, \boldsymbol{x}_\mathsf{m}|C_k) \\ &= p(\boldsymbol{x}_\mathsf{b}|C_k)p(\boldsymbol{x}_\mathsf{x}|C_k)p(\boldsymbol{x}_\mathsf{m}|C_k)p(C_k) \end{aligned}$$

Activation function

Linear regression: $\mu(\boldsymbol{x}, \boldsymbol{w})$ a linear function of $\boldsymbol{w}$

- Simplest case: $\mu(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + w_0$

Classification problem: want to predict discrete class labels, or posterior probabilities.

Generalized linear models:

- Generalization of the linear regression model: $\mu(\boldsymbol{x}, \boldsymbol{w}) = a(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + w_0)$
- Decision boundaries: constant $\mu(\boldsymbol{x}) \longrightarrow$ linear functions of $\boldsymbol{x}$!

Discriminative deterministic models

Discriminative deterministic models: Model directly the deterministic mapping between $\boldsymbol{x}$ and t via a parametrized function $\mu(\boldsymbol{x}, \boldsymbol{w})$ (discriminant function).

Binary classification:

- $\hat{t}(\boldsymbol{x})$ obtained by applying a threshold rule on $\mu(\boldsymbol{x}, \boldsymbol{w})$

Simplest form:

$$\begin{aligned}\mu(\boldsymbol{x}, \boldsymbol{w}) &= a(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0) \\ &= \boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0\end{aligned}$$

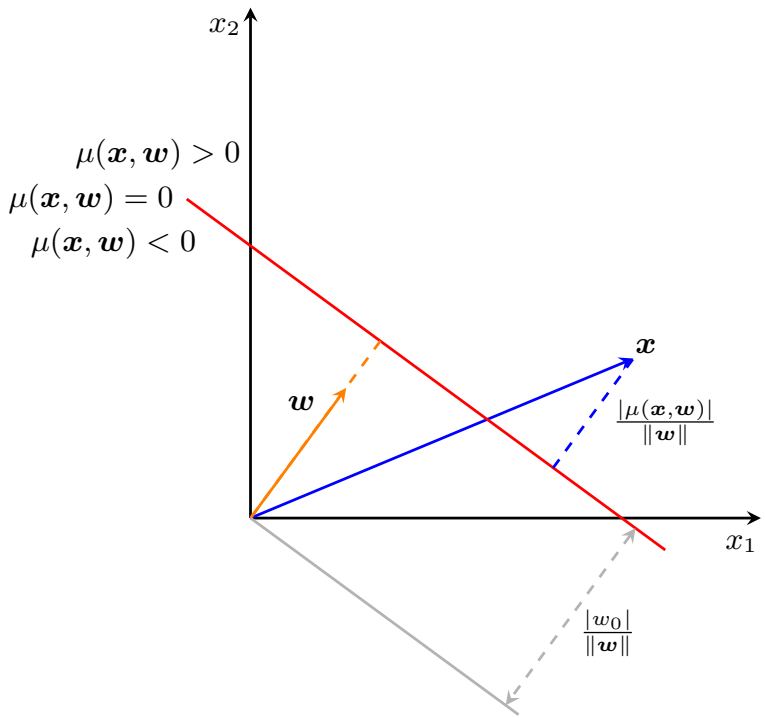
and

$$\begin{aligned}\hat{t}(\boldsymbol{x}, \boldsymbol{w}) &= \text{sign}(\mu(\boldsymbol{x}, \boldsymbol{w})) \\ &= \text{sign}(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + w_0)\end{aligned}$$

$$\boldsymbol{x} \in \mathcal{C}_1 \text{ if } \mu(\boldsymbol{x}, \boldsymbol{w}) > 0$$
$$\boldsymbol{x} \in \mathcal{C}_2 \text{ if } \mu(\boldsymbol{x}, \boldsymbol{w}) < 0$$

Geometric interpretation

Decision rule: Defines a hyperplane that separates the domain points classified as belonging to either of the two classes.



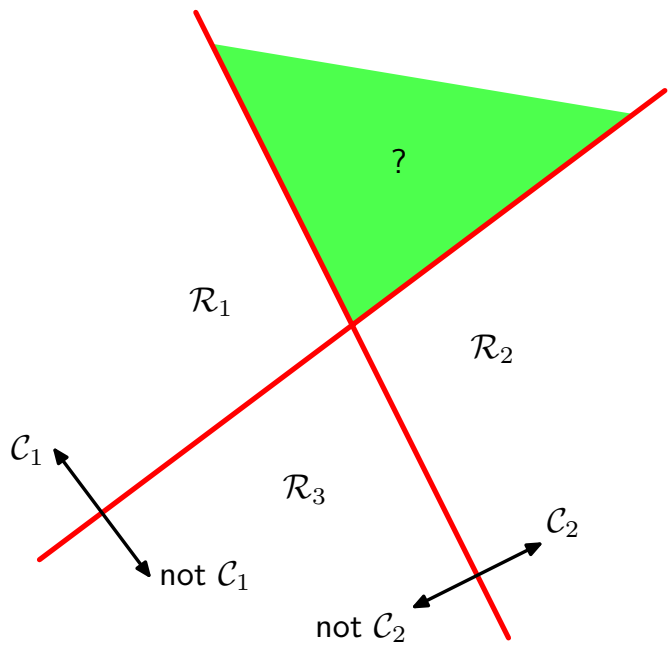
- $\boldsymbol{w}$ determines orientation of decision hyperplane
- w_0 determines location of decision surface
- Classification margin: $|\mu(\boldsymbol{x}, \boldsymbol{w})| / \|\boldsymbol{w}\|$

Multi-class classification

Combining 2-class discriminant functions:

One-versus the rest classifier: Separating points in C_k from points not in C_k.

- Leads to regions of input space that are ambiguously classified

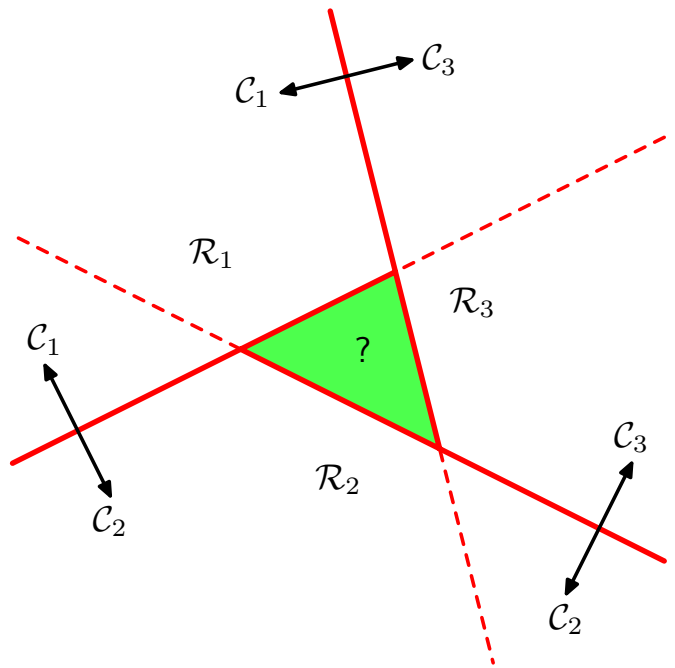


Multi-class classification

Combining 2-class discriminant functions:

One-versus-one classifier: $\binom{K}{2}$ binary discriminant functions. Each point classified according to a majority vote.

- Leads to regions of input space that are ambiguously classified



Multi-class classification

A single K-class discriminant with K linear functions

$$\mu_k(\boldsymbol{x}) = w_{k,0} + \boldsymbol{w}_k^\mathsf{T}\boldsymbol{x}$$

and

$$\hat{t}(\boldsymbol{x}) = \arg\max_k \mu_k(\boldsymbol{x})$$

$\mu_k(\boldsymbol{x})$ can be interpreted as $p(t = k|\boldsymbol{x})$.

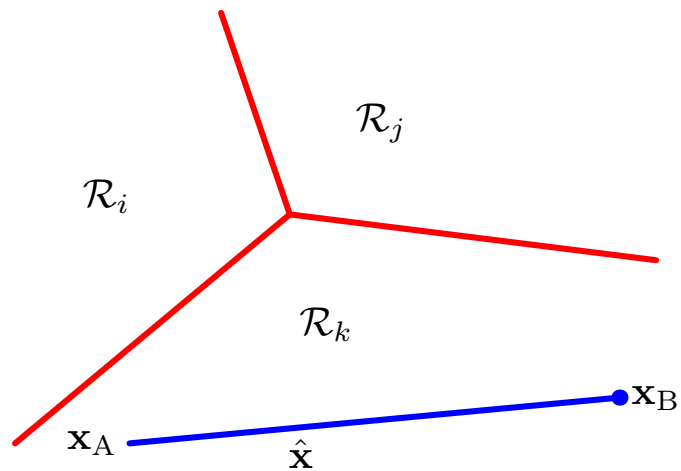
Decision boundary between C_k and C_j:

$$\{\boldsymbol{x} \,:\, (\boldsymbol{w}_k - \boldsymbol{w}_j)^\mathsf{T}\boldsymbol{x} + (w_{k,0} - w_{j,0}) = 0\}$$

Multi-class classification

A single K-class discriminant with K linear functions

$$\mu_k(\boldsymbol{x}) = w_{k,0} + \boldsymbol{w}_k^\mathsf{T}\boldsymbol{x}$$



- Decision regions singly connected and convex
- Parameters can be optimized via, e.g., least squares ($\boldsymbol{W}^* = (\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{T}$)

Discriminative probabilistic models

Discriminative probabilistic models: Model $p(C_k|\boldsymbol{x})$ via a parametrized model.

- Potentially more powerful than deterministic models since allow to model sources of uncertainty in the label assignment to input variables

We will consider:

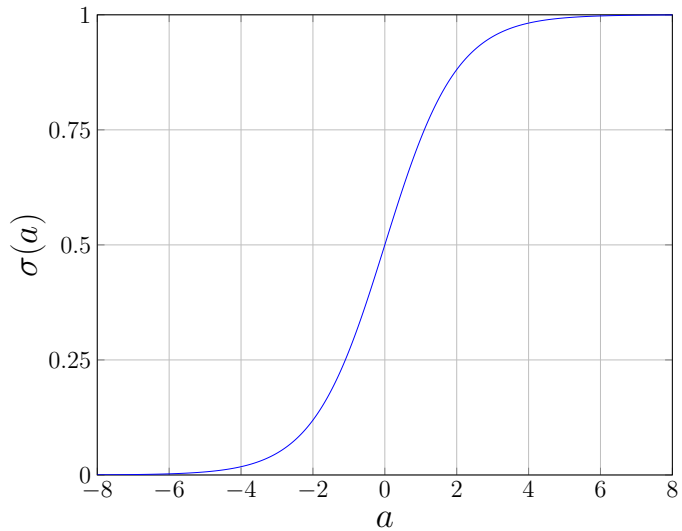
- Models linear in the parameters
- Applying a nonlinear transformation $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_M)$ to $\boldsymbol{x}$, then work with $\boldsymbol{\phi}(\boldsymbol{x})$ as our features

Decision boundaries linear in the feature space $\boldsymbol{\phi}$, nonlinear in original space $\boldsymbol{x}$.

Logistic regression

Binary classification:

$$
\begin{aligned}
p(\mathsf{t} = \mathcal{C}_1 | \boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{w}) &= p(\mathsf{t} = \mathcal{C}_1 | \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{w}) \\
&= \frac{\exp(\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}))}{1 + \exp(\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}))} \\
&= \frac{1}{1 + e^{-(\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}))}} \triangleq \sigma(\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}))
\end{aligned}
$$



For an M-dimensional feature space $\boldsymbol{\phi}$, M adjustable parameters.

Logistic regression

$$p(\mathrm{t} = \mathcal{C}_1 | \boldsymbol{\phi}, \boldsymbol{w}) = \frac{1}{1 + e^{-(\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}))}} \triangleq \sigma(\boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}))$$

Inference:

Misclassification error minimized by

$$\boldsymbol{x} \in \mathcal{C}_1 \text{ if } p(\mathcal{C}_1 | \boldsymbol{\phi}, \boldsymbol{w})) > 1/2$$
$$\boldsymbol{x} \in \mathcal{C}_2 \text{ if } p(\mathcal{C}_1 | \boldsymbol{\phi}, \boldsymbol{w})) < 1/2$$

Equivalently,

$$\boldsymbol{x} \in \mathcal{C}_1 \text{ if } \boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}) > 0$$
$$\boldsymbol{x} \in \mathcal{C}_2 \text{ if } \boldsymbol{w}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}) < 0$$

Logistic regression: Learning

ML learning:

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} \; p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w})$$

Likelihood function:

$$p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w}) = \prod_{i=1}^{N} p(t_i|\boldsymbol{x}_i, \boldsymbol{w}) = \prod_{i=1}^{N} p(t_i|\boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{w})$$
$$= \prod_{i=1}^{N} y_i^{t_i} (1 - y_i)^{1-t_i}$$

with $y_i = p(\mathsf{t}_i = \mathcal{C}_1|\boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{w})$

Negative log-likelihood:

$$-\ln p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w}) = -\sum_{i=1}^{N} \Big(t_i \ln(y_i) + (1 - t_i) \ln(1 - y_i) \Big)$$

Logistic regression: Learning

ML learning:

$$\begin{aligned}\boldsymbol{w}^* &= \arg\min_{\boldsymbol{w}} \; -\ln p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w}) \\ &= \arg\min_{\boldsymbol{w}} \; -\sum_{i=1}^{N} \Big(t_i \ln(y_i) + (1-t_i)\ln(1-y_i) \Big)\end{aligned}$$

(cross-entropy loss criterion)

No simple formula to find optimal $\boldsymbol{w}^*$.

However:

1. NLL has unique minimum unless classes perfectly separated by hyperplane
2. The negative log-likelihood surface is concave
3. The derivatives and the second derivatives can be easily computed

Efficient methods (SGD, Newton-Raphson) exist.

Discriminative probabilistic models

Other discriminative probabilistic models:

- Support vector machine
- Decision tree
- Random forest

Bayesian logistic regression

We want to obtain

$$p(t|\mathcal{D}, \boldsymbol{x}) = \int \underbrace{p(\boldsymbol{w}|\mathcal{D})}_{\text{posterior dist. of } \boldsymbol{w}} p(t|\boldsymbol{x}, \boldsymbol{w}) d\boldsymbol{w}$$

Bayesian approach: $p(t|\boldsymbol{x}, \boldsymbol{w})$ associated with each value of $\boldsymbol{w}$ weighted by the posterior belief

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\boldsymbol{w})p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w})}{p(t_{\mathcal{D}}|x_{\mathcal{D}})}$$

Bayesian logistic regression

We want to compute the full posterior $p(\boldsymbol{w}|\mathcal{D})$.

Problem: Exact Bayesian inference for logistic regression not possible as exact evaluation of $p(\boldsymbol{w}|\mathcal{D})$ intractable!

$$p(\boldsymbol{w}|\mathcal{D}) \propto p(\boldsymbol{w})p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w})$$
$$= p(\boldsymbol{w}) \prod_{i=1}^{N} p(t_i|\boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{w})$$

Bayesian logistic regression: More complex than for linear regression models (cannot integrate exactly over $\boldsymbol{w}$ since posterior not Gaussian) $\longrightarrow$ need to introduce some approximations.

Idea: Use Laplace approximation

The Laplace approximation

Laplace approximation: Aims to find a Gaussian approximation to a posterior distribution.

Case of a single continuous random variable z:

$$p(z) = \frac{1}{Z} f(z)$$

Z: (unknown) normalization constant ($Z = \int f(z) \mathrm{d}z$)

Goal: Find a Gaussian approximation $q(z)$ centered on a mode of $p(z)$.

The Laplace approximation

Goal: Find a Gaussian approximation $q(z)$ centered on a mode of $p(z)$.

1. Find a mode of $p(z)$ ($p'(z_0) = 0$)

Observation: The logarithm of a Gaussian is a quadratic function of the variables.

2. Taylor series expansion of $\ln f(z)$ centered on mode z_0:

$$\ln f(z) \approx \ln f(z_0) + (z - z_0) \left.\frac{\mathrm{d}}{\mathrm{d}z} \ln f(z)\right|_{z=z_0} + (z - z_0)^2 \frac{1}{2} \cdot \left.\frac{\mathrm{d}^2}{\mathrm{d}z^2} \ln f(z)\right|_{z=z_0}$$

$$= \ln f(z_0) + (z - z_0)^2 \frac{1}{2} \cdot \left.\frac{\mathrm{d}^2}{\mathrm{d}z^2} \ln f(z)\right|_{z=z_0}$$

$$= \ln f(z_0) - \frac{1}{2} A (z - z_0)^2$$

with

$$A = - \left.\frac{\mathrm{d}^2}{\mathrm{d}z^2} \ln f(z)\right|_{z=z_0}$$

The Laplace approximation

$$\ln f(z) \approx \ln f(z_0) - \frac{1}{2} A (z - z_0)^2$$

with

$$A = - \left. \frac{\mathrm{d}^2}{\mathrm{d}z^2} \ln f(z) \right|_{z=z_0}$$

We obtain:

$$f(z) \approx f(z_0) e^{-\frac{A}{2}(z-z_0)^2}$$

and normalized distribution

$$\begin{aligned} q(z) &= \left(\frac{A}{2\pi} \right)^{1/2} e^{-\frac{A}{2}(z-z_0)^2} \\ &= \mathcal{N}(z | z_0, A^{-1}) \end{aligned}$$

The Laplace approximation for higher dimensions

$$p(\boldsymbol{z}) = \frac{1}{Z} f(\boldsymbol{z})$$

1. Find a mode of $p(z)$ ($\nabla f(\boldsymbol{z}) = 0$)
2. Taylor series expansion of $\ln f(z)$ centered on mode z_0:

$$\ln f(\boldsymbol{z}) \approx \ln f(\boldsymbol{z}_0) - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^{\mathsf{T}} \boldsymbol{A} (\boldsymbol{z} - \boldsymbol{z}_0)$$

with $\boldsymbol{A}$ the $d \times d$ Hessian matrix (evaluated at $\boldsymbol{z} = \boldsymbol{z}_0$)

$$\boldsymbol{A} = -\nabla\nabla \ln f(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{z}_0} = \begin{pmatrix} \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_1^2} & \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_1 \partial z_2} & \cdots & \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_1 \partial z_d} \\ \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_2 \partial z_1} & \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_2^2} & \cdots & \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_2 \partial z_d} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_d \partial z_1} & \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_d \partial z_2} & \cdots & \frac{\partial^2 \ln f(\boldsymbol{z})}{\partial z_d^2} \end{pmatrix}$$

The Laplace approximation for higher dimensions

We obtain:

$$f(\boldsymbol{z}) \approx f(\boldsymbol{z}_0) e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{z}_0)^{\mathsf{T}} \boldsymbol{A} (\boldsymbol{z}-\boldsymbol{z}_0)}$$

and normalized distribution

$$\begin{aligned} q(\boldsymbol{z}) &= \frac{|\boldsymbol{A}|^{1/2}}{(2\pi)^{d/2}} e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{z}_0)^{\mathsf{T}} \boldsymbol{A} (\boldsymbol{z}-\boldsymbol{z}_0)} \\ &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{z}_0, \boldsymbol{A}^{-1}) \end{aligned}$$

Bayesian logistic regression

We want to compute the full posterior $p(\boldsymbol{w}|\mathcal{D})$.

Problem: Exact Bayesian inference for logistic regression not possible as exact evaluation of $p(\boldsymbol{w}|\mathcal{D})$ intractable!

$$p(\boldsymbol{w}|\mathcal{D}) \propto p(\boldsymbol{w})p(t_{\mathcal{D}}|x_{\mathcal{D}}, \boldsymbol{w})$$
$$= p(\boldsymbol{w}) \prod_{i=1}^{N} p(t_i|\boldsymbol{x}_i, \boldsymbol{w})$$

Idea: Use Laplace approximation

Also need to select a prior for $\boldsymbol{w} \longrightarrow$ a Gaussian prior,

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{S}_0)$$

Bayesian logistic regression

Log-likelihood function:

$$\ln p(\boldsymbol{w}|\mathcal{D}) = -\frac{1}{2}(\boldsymbol{w}-\boldsymbol{m}_0)^\mathsf{T}\boldsymbol{S}_0^{-1}(\boldsymbol{w}-\boldsymbol{m}_0) + \sum_{i=1}^{N}\Big(t_i \ln(y_i) + (1-t_i)\ln(1-y_i)\Big)$$

$$+ \text{const.}$$

We proceed as follows:

1. We maximize $p(\boldsymbol{w}|\mathcal{D})$ (MAP solution $\boldsymbol{w}_{\text{MAP}}$) $\longrightarrow$ defines the mean of the Gaussian
2. Covariance given by

$$\boldsymbol{S}_N^{-1} = -\nabla\nabla \ln p(\boldsymbol{w}|\mathcal{D}) = \boldsymbol{S}_0^{-1} + \sum_{i=1}^{N} y_i(1-y_i)\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\mathsf{T}$$

Laplace approximation of $p(\boldsymbol{w}|\mathcal{D})$:

$$q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{\text{MAP}}, \boldsymbol{S}_N)$$

Bayesian logistic regression

Why Laplace approximation?

- Posteriors more "Gaussian like" as sample-size increases $\longrightarrow$ Good approximation for sufficiently large N

Predictive distribution

Want to predict t! $\longrightarrow$ Want to learn posterior predictive distribution $p(t|\mathcal{D}, \boldsymbol{x}) = p(t|\mathcal{D}, \boldsymbol{\phi})$

Predictive distribution for class $\mathcal{C}_1$, $p(\mathrm{t} = \mathcal{C}_1|\mathcal{D}, \boldsymbol{\phi})$:

$$p(\mathcal{C}_1|\mathcal{D}, \boldsymbol{\phi}) = \int p(\mathcal{C}_1|\boldsymbol{\phi}, \boldsymbol{w}) p(\boldsymbol{w}|\mathcal{D}) \mathrm{d}\boldsymbol{w}$$

Given $\boldsymbol{\phi}$ (or $\boldsymbol{x}$), we can forecast t via the predictive distribution.

Predictive distribution

We use

$$p(\mathsf{t} = \mathcal{C}_1|\boldsymbol{\phi}, \boldsymbol{w}) = \frac{1}{1 + e^{-(\boldsymbol{w}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x}))}} \triangleq \sigma(\boldsymbol{w}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x}))$$

and

$$q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{\mathsf{MAP}}, \boldsymbol{S}_N)$$

We obtain:

$$\begin{aligned} p(\mathcal{C}_1|\mathcal{D}, \boldsymbol{\phi}) &\approx \int \sigma(\boldsymbol{w}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x}))q(\boldsymbol{w})\mathsf{d}\boldsymbol{w} \\ &= \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)\mathsf{d}a \end{aligned}$$

with

$$\begin{aligned} a &= \boldsymbol{w}^\mathsf{T}\boldsymbol{\phi} \\ \mu_a &= \mathbb{E}[a] = \boldsymbol{w}_{\mathsf{MAP}}^\mathsf{T}\boldsymbol{\phi} \\ \sigma_a^2 &= \mathsf{Var}[a] = \boldsymbol{\phi}^\mathsf{T}\boldsymbol{S}_N\boldsymbol{\phi} \end{aligned}$$

Predictive distribution

$$p(\mathcal{C}_1|\mathcal{D}, \phi) \approx \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)\mathrm{d}a$$

Cannot be evaluated analytically

Idea: exploit that $\sigma(a)$ is similar to the probit function $\Phi(a)$ to obtain an approximation

$$\Phi(a) \triangleq \int_{-\infty}^{a} \mathcal{N}(x|0, 1)\mathrm{d}x$$

$\sigma(a)$ well approximated by $\Phi(\lambda a)$ with $\lambda = \sqrt{\pi/8}$.

$$\int \Phi(\lambda a)\mathcal{N}(a|\mu, \sigma^2)\mathrm{d}a = \Phi\left(\frac{\mu}{(\lambda^{-2} + \sigma^2)^{1/2}}\right)$$

Predictive distribution

$$\int \Phi(\lambda a)\mathcal{N}(a|\mu, \sigma^2)\mathrm{d}a = \Phi\left(\frac{\mu}{(\lambda^{-2} + \sigma^2)^{1/2}}\right)$$

Then:

$$\begin{aligned} p(\mathcal{C}_1|\mathcal{D}, \boldsymbol{\phi}) &\approx \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)\mathrm{d}a \\ &\approx \Phi\left(\frac{\mu_a}{(\lambda^{-2} + \sigma_a^2)^{1/2}}\right) \\ &\approx \sigma\left(\frac{\mu_a}{(1 + \pi\sigma_a^2/8)^{1/2}}\right) \end{aligned}$$

Generative probabilistic models

Generative probabilistic models: Model $p(\boldsymbol{x}, t)$ specifying $p(t)$ ($p(C_k)$) and $p(\boldsymbol{x}|t)$ ($p(\boldsymbol{x}|C_k)$). Then,

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{p(\boldsymbol{x})}$$

- Make more assumptions about the data by considering distribution of covariates $\boldsymbol{x}$ $\longrightarrow$ may suffer from bias when model incorrectly selected
- Capability to capture properties of $\boldsymbol{x}$ can improve learning if $p(\boldsymbol{x}|t)$ has significant structure

Generative probabilistic models

Binary classification:

$$p(\mathcal{C}_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$
$$= \frac{1}{1+e^{-a}} \triangleq \sigma(a)$$

with

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

Multi-class classification ($K > 2$):

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\boldsymbol{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}} \qquad \text{(softmax)}$$

with

$$a_k = \ln\, p(\boldsymbol{x}|C_k)p(C_k)$$

Generative probabilistic models

Model: Different models for $p(\boldsymbol{x}|C_i)$ lead to different generative models. Generative models typically defined by assuming

$$\mathbf{x}|\mathsf{t} = t \sim \text{exponential}(\eta_t)$$

Exponential family: $p(\boldsymbol{x}|\boldsymbol{\eta}) = \frac{1}{Z(\boldsymbol{\eta})} h(\boldsymbol{x}) \exp(\boldsymbol{\eta}^\mathsf{T} \boldsymbol{u}(\boldsymbol{x}))$

Gaussian discriminant analysis: Class-conditional density modeled by a multivariate Gaussian

- Linear discriminant analysis
- Quadratic discriminant analysis

Linear discriminant analysis

Principle: $\boldsymbol{x} \in \mathbb{R}^d$ is Gaussian distributed for each class C_k, with covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$,

$$\mathbf{x}|\mathrm{t} = C_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$$

Equivalently,

$$p(\boldsymbol{x}|C_k) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

Fitting an LDA model to data $\equiv$ estimating $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$.

LDA gives rise to linear decision boundaries.

Linear discriminant analysis

$$
\begin{aligned}
\log \frac{p(\mathsf{t}=C_i|\boldsymbol{x})}{p(\mathsf{t}=C_j|\boldsymbol{x})} &= \log \frac{p(\boldsymbol{x}|C_i)p(\mathsf{t}=C_i)}{p(\boldsymbol{x}|C_j)p(\mathsf{t}=C_j)} \\
&= \log \frac{p(\boldsymbol{x}|C_i)}{p(\boldsymbol{x}|C_j)} + \log \frac{p(\mathsf{t}=C_i)}{p(\mathsf{t}=C_j)} \\
&= -\frac{1}{2}\boldsymbol{\mu}_i^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \frac{1}{2}\boldsymbol{\mu}_j^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j + \boldsymbol{x}^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \log \frac{p(\mathsf{t}=C_i)}{p(\mathsf{t}=C_j)} \\
&= w_0 + \boldsymbol{w}^\mathsf{T}\boldsymbol{x}
\end{aligned}
$$

with

$$
\begin{aligned}
\boldsymbol{w} &= \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \\
w_0 &= -\frac{1}{2}\boldsymbol{\mu}_i^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \frac{1}{2}\boldsymbol{\mu}_j^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j + \log \frac{p(C_i)}{p(C_j)}
\end{aligned}
$$

Decision boundaries: points for which $p(\mathsf{t}=C_i|\boldsymbol{x}) = p(\mathsf{t}=C_j|\boldsymbol{x}) \longrightarrow \{\boldsymbol{x} : w_0 + \boldsymbol{w}^\mathsf{T}\boldsymbol{x} = 0\}$

Quadratic discriminant analysis

Principle: $\boldsymbol{x} \in \mathbb{R}^D$ is Gaussian distributed for each class C_k, with covariance matrix $\boldsymbol{\Sigma}_k$,

$$\mathbf{x}|\mathrm{t} = C_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Equivalently,

$$p(\boldsymbol{x}|C_k) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\mathsf{T}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

Discriminant functions:

$$\log \frac{p(\mathrm{t} = C_i|\boldsymbol{x})}{p(\mathrm{t} = C_j|\boldsymbol{x})} = -\frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\mathsf{T}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \log\frac{p(C_i)}{p(C_j)}$$

The decision boundaries are given by a quadratic equation.

Gaussian discriminant analysis: Learning

Need to fit model: Determine values of $\boldsymbol{w}$ and $p(\mathcal{C}_k)$ via maximum likelihood criterion.

Binary case:

- N training examples $\mathcal{D} = \{(\boldsymbol{x}_i, t_i)\}$
- $t = 1$ for class C_1 and $t = 0$ for class C_2
- $p(C_1) = \pi$, $p(C_2) = 1 - \pi$

We have:

$$p(\boldsymbol{x}_i, \mathcal{C}_1) = p(\mathcal{C}_1)p(\boldsymbol{x}_i|\mathcal{C}_1) = \pi\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_1, \Sigma)$$
$$p(\boldsymbol{x}_i, \mathcal{C}_2) = p(\mathcal{C}_2)p(\boldsymbol{x}_i|\mathcal{C}_2) = (1 - \pi)\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_2, \Sigma)$$

Likelihood function:

$$p(\mathcal{D}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \left(\pi\mathcal{N}\left((\boldsymbol{x}_i|\mu_1, \Sigma)\right)^{t_i} \left((1 - \pi)\mathcal{N}\left((\boldsymbol{x}_i|\mu_2, \Sigma)\right)^{1-t_i}\right.\right.$$

Gaussian discriminant analysis: Learning

Likelihood function

$$p(\mathcal{D}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \left(\pi \mathcal{N}\left((\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}\right)\right)^{t_i} \left((1-\pi)\mathcal{N}\left((\boldsymbol{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}\right)\right)^{1-t_i}$$

Log-likelihood function:

$$\begin{aligned}
&\ln p(\mathcal{D}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \\
&\quad = \sum_{i=1}^{N} \left(t_i \ln\left(\pi \mathcal{N}\left((\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}\right)\right) + (1-t_i) \ln\left((1-\pi)\mathcal{N}\left((\boldsymbol{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}\right)\right) \right) \\
&\quad = \sum_{i=1}^{N} \left(t_i \ln(\pi) + (1-t_i) \ln(1-\pi) \right) + \sum_{i=1}^{N} \left(t_i \ln\left(\mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}\right)\right) \right) \\
&\qquad + \sum_{i=1}^{N} \left((1-t_i) \ln\left(\mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}\right)\right) \right)
\end{aligned}$$

Can estimate π, $\boldsymbol{\mu}_i$, and $\boldsymbol{\Sigma}$ separately!

Gaussian discriminant analysis: Learning

Maximization with respect to π:

$$\sum_{i=1}^{N} \Big(t_i \ln(\pi) + (1 - t_i) \ln(1 - \pi) \Big)$$

Differentiating and equating to zero:

$$\frac{\mathrm{d}}{\mathrm{d}\pi} \sum_{i=1}^{N} \Big(t_i \ln(\pi) + (1 - t_i) \ln(1 - \pi) \Big) = \sum_{i=1}^{N} t_i \frac{1}{\pi} - (1 - t_i) \frac{1}{1 - \pi}$$

$$\Longrightarrow \quad \pi = \frac{1}{N} \sum_{i=1}^{N} t_i = \frac{N_1}{N}$$

N_1: number of data points in C_1

Gaussian discriminant analysis: Learning

Maximization with respect to μ_1:

$$\sum_{i=1}^{N} \left(t_i \ln \left(\mathcal{N} \left(\boldsymbol{x}_i | \mu_1, \Sigma \right) \right) \right) = -\frac{1}{2} \sum_{i=1}^{N} t_i (\boldsymbol{x}_i - \boldsymbol{\mu}_1)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_1) + \text{const.}$$

Differentiating and equating to zero:

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{i=1}^{N} t_i \boldsymbol{x}_i$$

Similarly,

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{i=1}^{N} (1 - t_i) \boldsymbol{x}_i$$

Gaussian discriminant analysis: Learning

Maximization with respect to $\boldsymbol{\Sigma}$:

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

with

$$\boldsymbol{\Sigma}_1 = \frac{1}{N_1} \sum_{\substack{i=1: \\ t_i = C_1}}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^\mathsf{T}$$

$$\boldsymbol{\Sigma}_2 = \frac{1}{N_2} \sum_{\substack{i=1: \\ t_i = C_2}}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_2)(\boldsymbol{x}_i - \boldsymbol{\mu}_2)^\mathsf{T}$$

Gaussian discriminant analysis: Learning

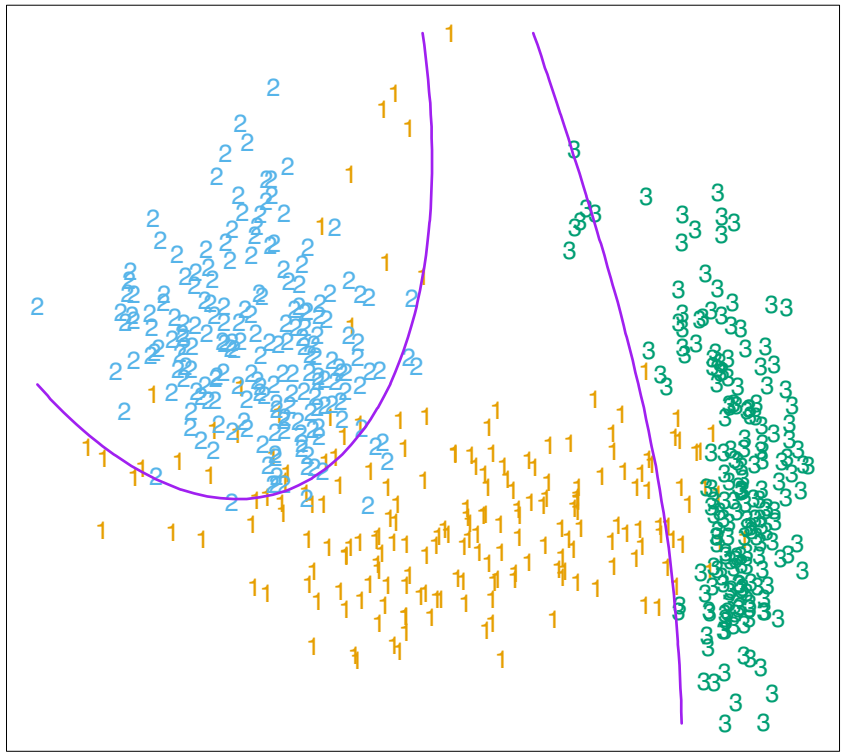
Multi-class classification ($K > 2$):

$$\pi_k = \frac{N_k}{N}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{\substack{i=1: \\ t_i = C_k}}^{N} t_i \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{\substack{i=1: \\ t_i = C_1}}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\mathsf{T}$$

Quadratic discriminant analysis



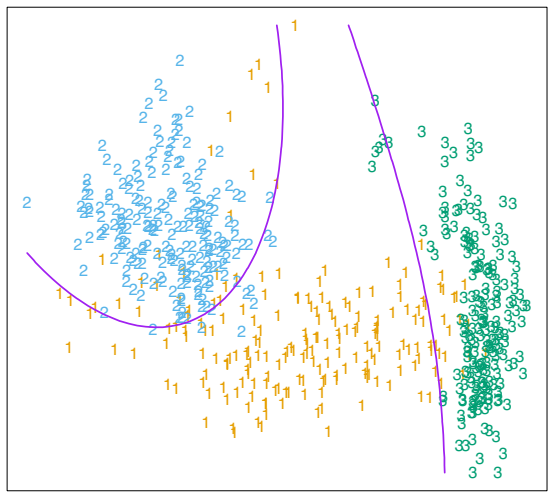
- Data $\{\boldsymbol{x}_i = (x_{i,1}, x_{i,2})\}$ generated from three different Gaussian mixtures

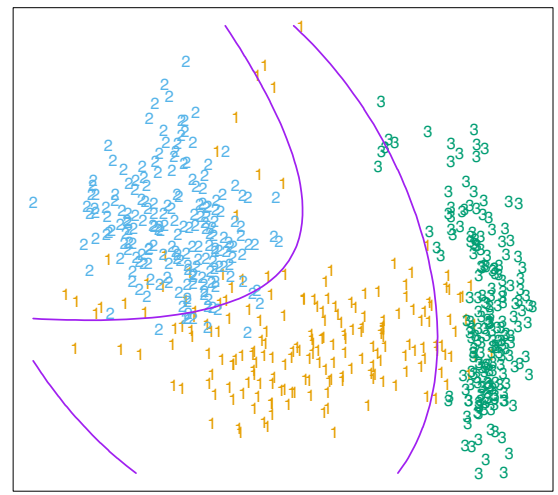
Gaussian discriminant analysis

Two alternative ways to realize quadratic decision boundaries:

1. Apply QDA in the original d-dimensional space $\boldsymbol{x} = (x_1, \ldots, x_d)$
2. Apply LDA in an augmented space of dimension $d^2 + 1$:
   $\boldsymbol{x}' = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2, x_1, x_2, \ldots, x_{d-1}x_d)$

Linear functions in the augmented space correspond to quadratic functions in the original d-dimensional space.





- Left: QDA applied to original 2-dimensional space $\boldsymbol{x} = (x_1, x_2)$
- Right: LDA applied to 5-dimensional space $\boldsymbol{x} = (x_1, x_2, x_1x_2, x_1^2, x_2^2)$

Gaussian discriminant analysis

Assumption in LDA and QDA: $p(\boldsymbol{x}|C_i)$ are Gaussian distributed

. . . and yet perform well on a large variety of classification tasks!

Gaussian discriminant analysis: Learning

Observations:

- From $p(\boldsymbol{x}|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$, we use Bayes' theorem to find $p(\mathcal{C}_k|\boldsymbol{x})$
- Generative probabilistic model: We can generate synthetic data by drawing values of $\boldsymbol{x}$ from $p(\boldsymbol{x})$
- ML learning: fast and simple. . .but it may overfit in high dimensions

Discriminative or generative models?

Easy to fit?

- Generative: Easy (counting and averaging)
- Discriminative (logistic regression): Solving a complex optimization problem

Adding more classes?

- Generative: No need to retrain (parameters of each class conditional density estimated independently)
- Discriminative: Need to retrain (all parameters interact)

Feature processing?

- Generative: Hard
- Discriminative: Easy, we can simply replace $\boldsymbol{x}$ by $\boldsymbol{\phi}(\boldsymbol{x})$

Reading

"Pattern recognition and machine learning,"

Chapter 4 (Intro, 4.1.1–4.1.3, 4.1.7, 4.2 (until 4.2.3), 4.3 (until 4.3.4), 4.4. (not 4.4.1), 4.5)