Ch4. Classification

ST4240, 2015/2016

Version 0.1

Alexandre Thiéry

Department of Statistics and Applied Probability

Outline

Classification

- Some possibly unstructured data
- Extract some feature $x \in \mathcal{X}$ from the data
- A finite set of possible classes $\mathcal{Y}$.
- Training examples: $\{(x_i, y_i)\}_{i=1}^{N}$
- **Possible goals**:
  - Design a function $f : \mathcal{X} \to \mathcal{Y}$ that predicts the class of a new piece of data. (E.g. Support Vector Machine)
  - Design a function $f : \mathcal{X} \to [0, 1]^{|\mathcal{Y}|}$ that predicts the probability that a new piece of data belongs to each class. (E.g. naive Bayes classifier, Logistic regression, Random forest)

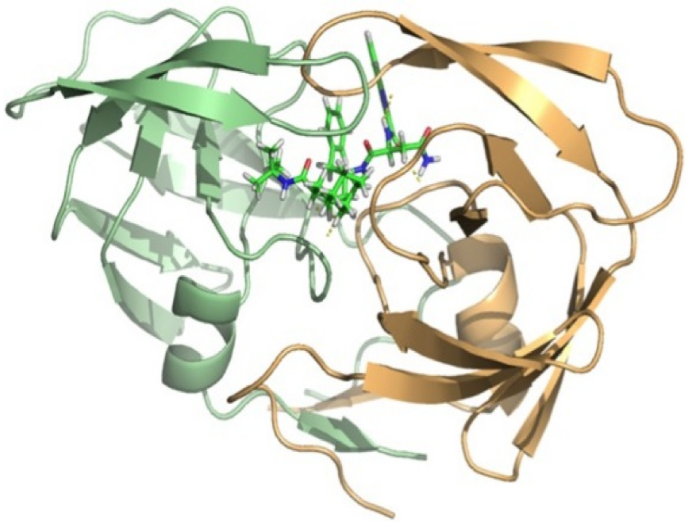
Titanic: who survived?

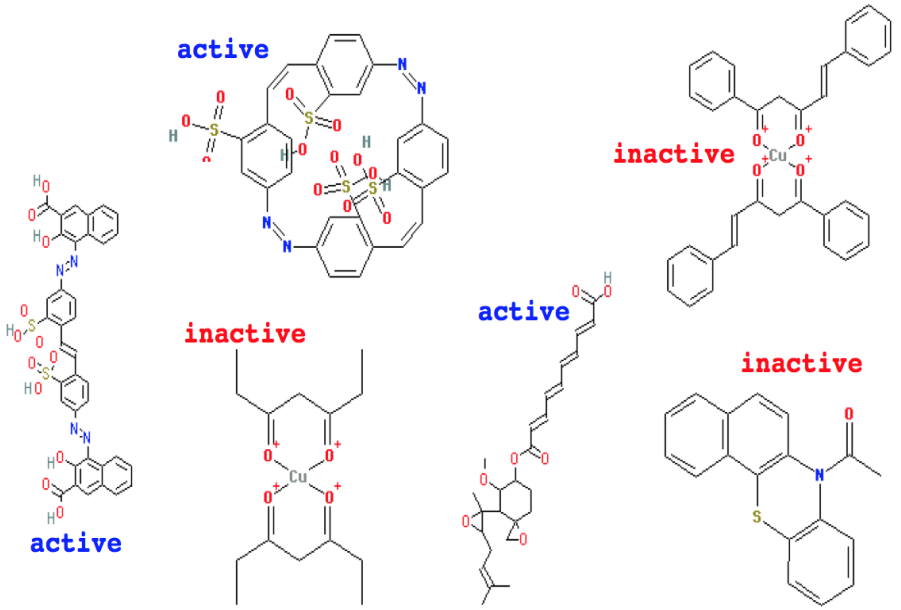
```
> head(titanic)
  Survived Pclass    Sex Age SibSp Parch    Fare Embarked
1        0      3   male  22     1     0  7.2500        S
2        1      1 female  38     1     0 71.2833        C
3        1      3 female  26     0     0  7.9250        S
4        1      1 female  35     1     0 53.1000        S
5        0      3   male  35     0     0  8.0500        S
7        0      1   male  54     0     0 51.8625        S
```

Credit scoring

Variable Name	Description	Type
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y/N
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
DebtRatio	Monthly debt payments, alimony,living costs divided by monthly gross income	percentage
MonthlyIncome	Monthly income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans and Lines of credit	integer
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

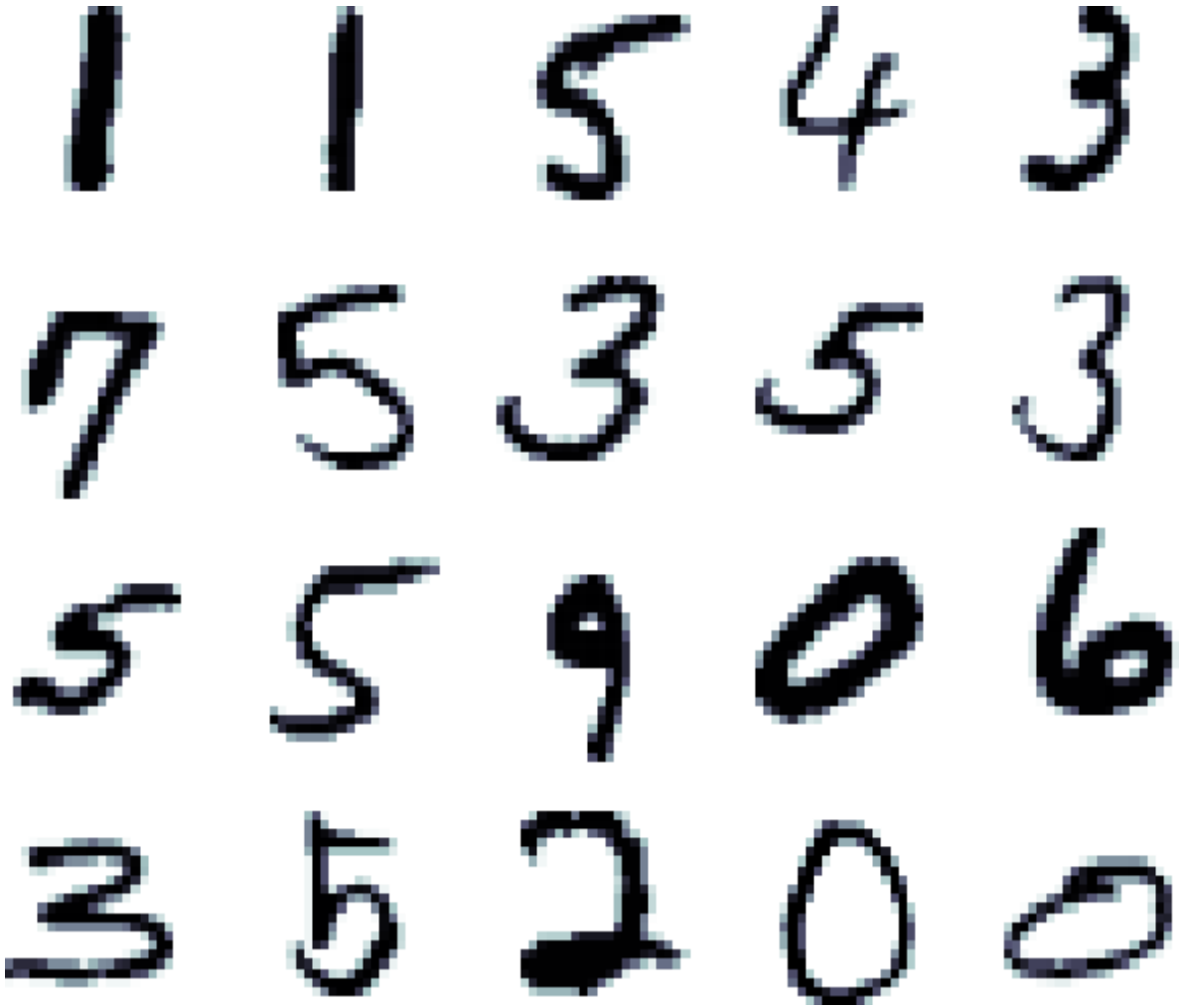
Drug discovery





Given a new candidate molecule, is it likely to be active?

Digit recognition



Advantage of probabilistic output

- Making an error of one type may be much more "expensive" than another (e.g. cancer v.s. no cancer)
- Can tell if we are uncertain about our prediction: and if so, we can refuse to classify it, and pass the case on to a human expert.
- Easier to combine several probabilistic output given by several models

Error rate and optimality

- Suppose that training examples $\{(x_i, y_i)\}$ are samples from a distribution (X, Y) and $Y \in \{-1, 1\}$
- Suppose that one is trying to minimise the error rate

$$(\text{Error rate}) \equiv \mathbb{P}\left(f(X) \neq Y\right)$$

- If one is trying to minimise the error rate, it is optimal to choose

$$f(x) = \begin{cases} 1 & \text{if} \quad \mathbb{P}(Y = 1 \mid X = x) \geq 1/2 \\ -1 & \text{if} \quad \mathbb{P}(Y = -1 \mid X = x) < 1/2. \end{cases}$$
$$= \textbf{Sign}(\mathbb{E}[Y \mid X = x]).$$

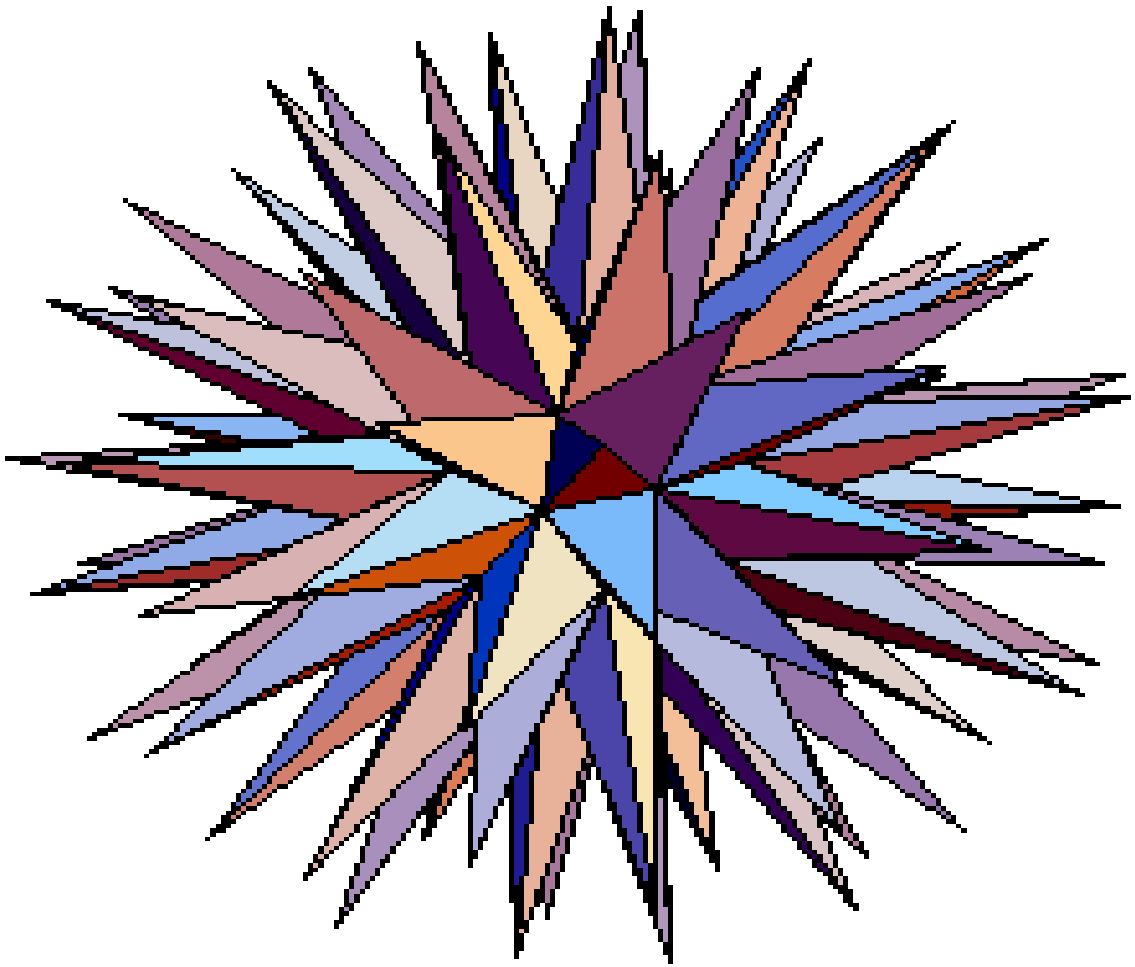
Is the problem solved?

- If one is trying to minimise the error rate, it thus suffices to estimate the conditional probability $\mathbb{P}(Y \mid X)$ from the training examples $\{x_i, y_i\}_{i=1}^N$. Is the problem solved, then?
- **Curse of dimensionality**: typically, $x \in \mathcal{X}$ is high dimensional (e.g. many features). For simplicity, let us suppose that

$$x_i = (x_{i,1}, \ldots, x_{i,d}) \in \{a, b\}^d$$

and that there is only two classes. To learn $\mathbb{P}(Y \mid X)$, one needs to learn roughly 2^d parameters!! Hopeless for d reasonably high.

High-Dimensional spaces are weird...



Measuring performances: error rate

- Error rate: note that this is not always a sensible thing to do, especially when the classes are highly unbalanced

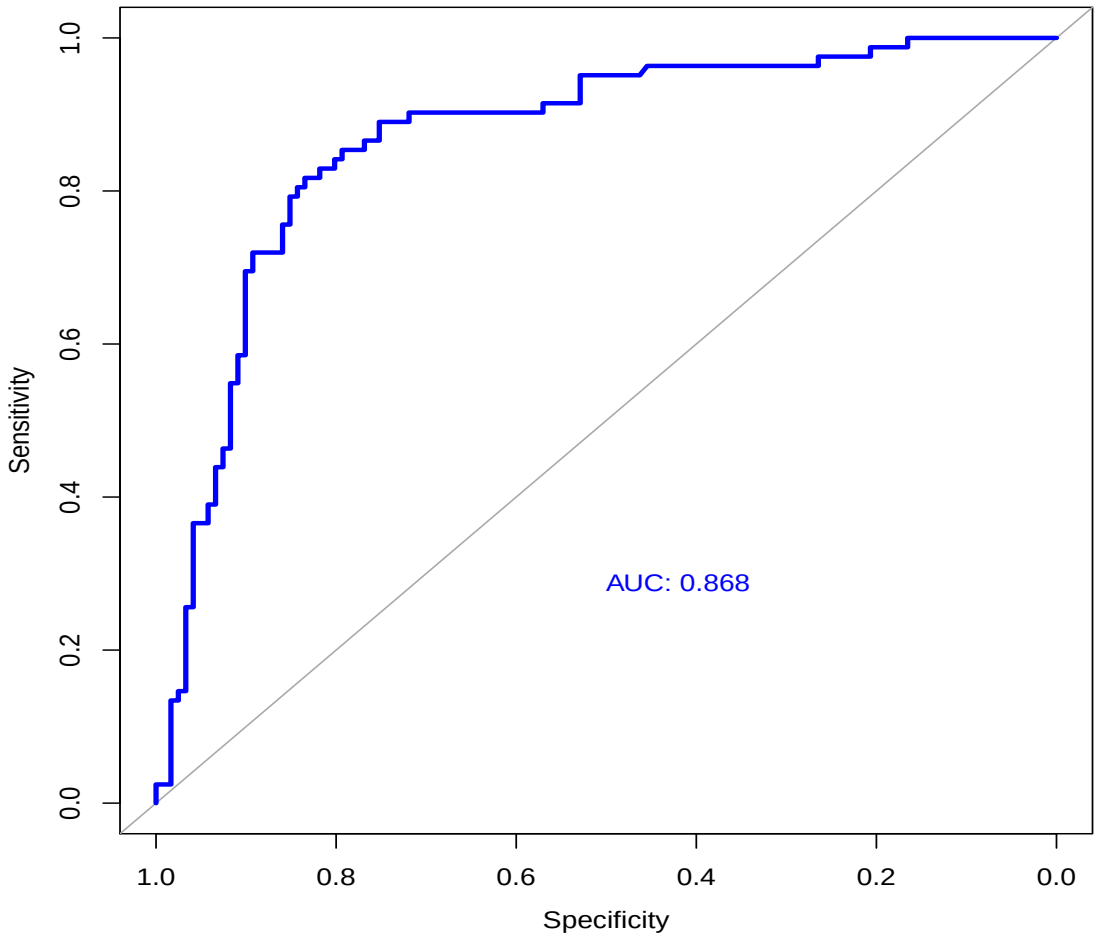
$$(\text{Error rate}) \equiv \mathbb{P}\left(f(X) \neq Y\right) \approx \frac{1}{N} \sum_{i=1}^{N} \text{Loss}\left(f(x_i), y_i\right)$$

  with $\text{Loss}\left(f(x), y\right) = \mathbb{I}\left(f(x) \neq y\right)$.

- Directly aiming at minimising the error rate leads most of the time to a **non-convex** optimisation problem.

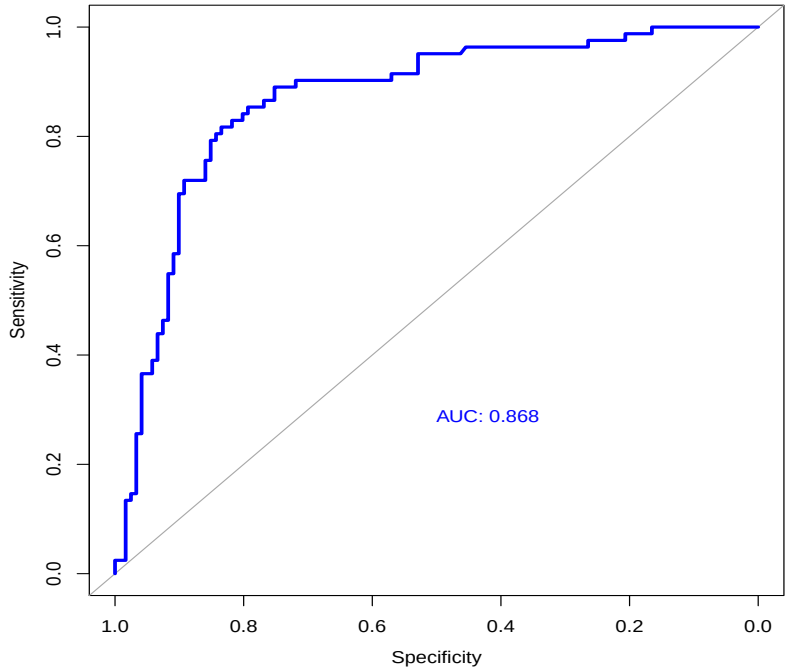
Measuring performances: confusion matrix

Measuring performances: ROC curve



Receiver Operating Characteristic (ROC) of a probabilistic output.

Measuring performances: AUC



- **A**rea **U**nder the **C**urve
- AUC=1 for a perfect classifier
- AUC=0.5 for a random/useless classifier
- Also equals the probability that a random true positive has a score higher that a random true negative.
- [**Exercise**] : would you be happy if you managed to design a classifier with an AUC extremely close to zero?

Outline

Bayesian Classification

- Given a new instance $x \in \mathcal{X}$ and a probabilistic model, the posterior probability that this new piece of data belongs to a class $y \in \mathcal{Y}$ is

$$\mathbb{P}(y \mid x) = \frac{\mathbb{P}(x \mid y)\ \mathbb{P}(y)}{\mathbb{P}(x)} \propto \mathbb{P}(x \mid y)\ \mathbb{P}(y)$$

- for a d-dimensional feature $x = (x^{(1)}, \ldots, x^{(d)})$, it is typically **very difficult to reliably evaluate** $\mathbb{P}(x \mid y)$ from a set of training examples that is not huge: **curse of dimensionality.**

Posterior independence assumption

- In full generality, we have

$$\mathbb{P}(x \mid y) = \mathbb{P}(x^{(1)} \mid y) \times \mathbb{P}(x^{(2)} \mid y, x^{(1)}) \times \ldots \times \mathbb{P}(x^{(d)} \mid y, x^{(1:[d-1])})$$

- Naives Bayes classifier **assumes** posterior independence,

$$\mathbb{P}(x \mid y) = \mathbb{P}(x^{(1)} \mid y) \times \mathbb{P}(x^{(2)} \mid y) \times \ldots \times \mathbb{P}(x^{(d)} \mid y)$$

Posterior independence assumption: examples

- Three dimensional features: (Height, Weight, Number of shoes).
- Two possible classes: $\mathcal{Y} = (\text{Female, Male})$
- In full generality,

$$\begin{aligned}\mathbb{P}(\text{weight=70kg, height=1.7m, shoes=few} \mid \text{male}) = \\ \mathbb{P}(\text{weight=70kg} \mid \text{male}) \times \\ \mathbb{P}(\text{height=1.7} \mid \text{male,weight=70kg}) \times \\ \mathbb{P}(\text{shoes=few,} \mid \text{male,weight=70kg, height=1.7}).\end{aligned}$$

- Posterior independence assumption,

$$\begin{aligned}\mathbb{P}(\text{weight=70kg, height=1.7m, shoes=few} \mid \text{male}) = \\ \mathbb{P}(\text{weight=70kg} \mid \text{male}) \times \\ \mathbb{P}(\text{height=1.7} \mid \text{male}) \times \\ \mathbb{P}(\text{shoes=few,} \mid \text{male}).\end{aligned}$$

Naive Bayes Classifier

- Training set (x_i, y_i) with d-dimensional features,

$$x_i = (x_i^{(1)}, \ldots, x_i^{(d)}),$$

  where $x_i^{(j)} \in \mathcal{X}_j$ and $\mathcal{Y} \equiv \{1, 2, \ldots, R\}$.
- Estimate $\mathbb{P}(y = r)$ and $\mathbb{P}(x \mid y = r)$ from training examples
- Given a new piece of data with feature $x = (x^{(1)}, \ldots, x^{(d)})$, the naive Bayes classifier estimate that the probability that it belongs to class r is

$$\frac{\mathbb{P}(x \mid y = r)\ \mathbb{P}(y = r)}{\sum_{k=1}^{R} \mathbb{P}(x \mid y = k)\ \mathbb{P}(y = k)}$$

  where $\mathbb{P}(x \mid y = r) = \mathbb{P}(x^{(1)} \mid y = r) \times \ldots \times \mathbb{P}(x^{(d)} \mid y = r)$

Naive Bayes Classifier: categorical class probabilities

- To estimate the class probabilities $\mathbb{P}(y = r)$ for $r = 1, \ldots, R$, a pragmatic solution is to use

$$\mathbb{P}(y = r) = \frac{\textbf{(Number of training examples where } y = r\textbf{)}}{\textbf{(Total number of training examples)}}$$

Naive Bayes Classifier: covariates probabilities

- Categorical features: $x^{(j)} \in \mathcal{X}_j \equiv \{1, 2, \ldots, R_j\}$. The MLE estimate of $\mathbb{P}(x^{(j)} = s \mid y = r)$ from a multinomial distribution is

$$\frac{\textbf{(Number of training examples where } x^{(j)} = s \textbf{ and } y = r\textbf{)}}{\textbf{(Total number of training examples with } y = r\textbf{)}}$$

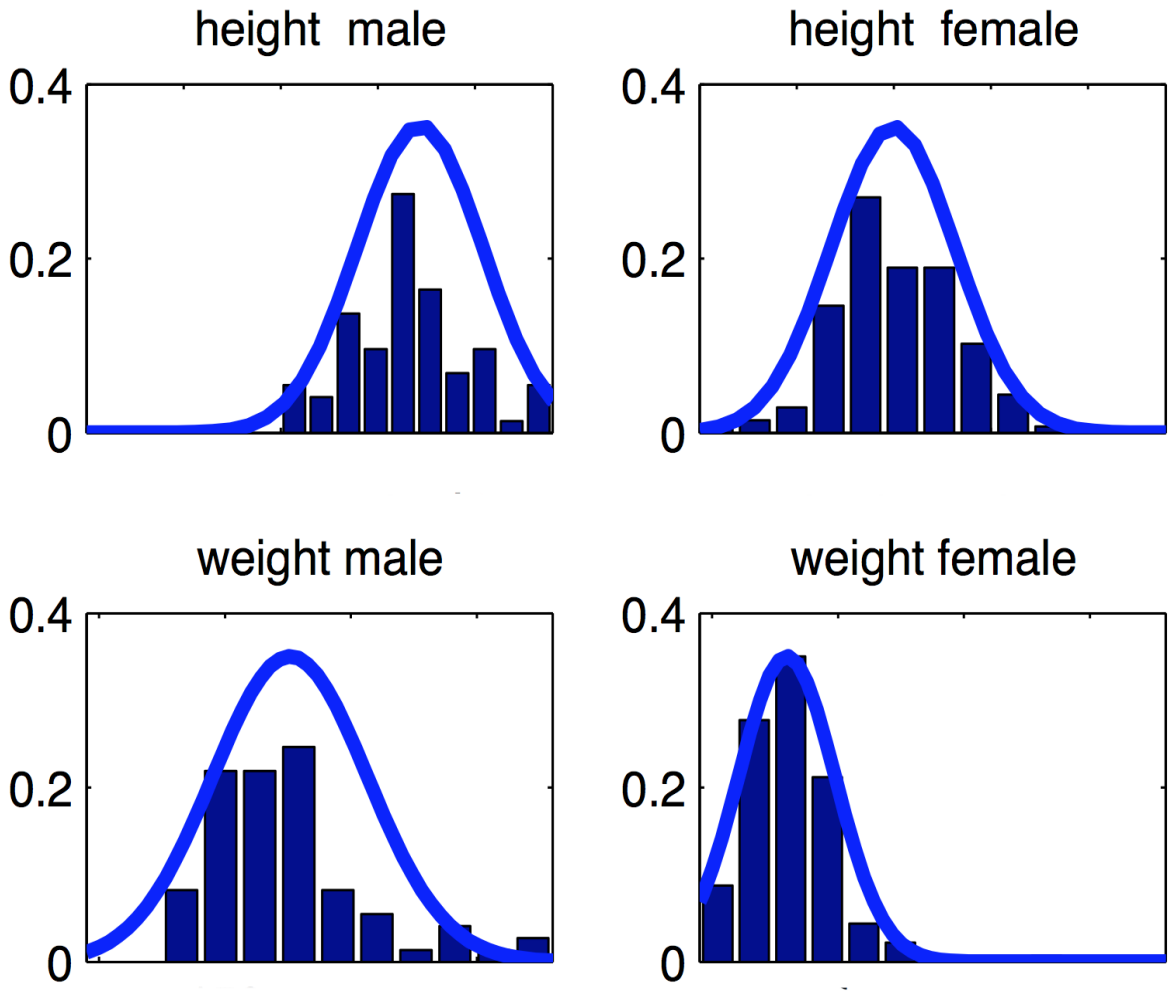
- Continuous features: $x^{(j)} \in \mathcal{X}_j \equiv \mathbb{R}$. To estimate $\mathbb{P}(x^{(j)} = s \mid y = r)$ one can fit a Gaussian distribution to the training example where $y = r$.
- Indeed, many other choices of family of distributions.
- We will see that sometimes the MLE is not adapted.

Male v.s. Female: (height, weight, shoes)

Sex	*M*	*M*	*F*	*F*	*F*	*M*	*M*	*M*	*M*	*F*	*F*	*F*
Height	1.70	1.67	1.60	1.62	1.54	1.82	1.75	1.7	1.69	1.6	1.70	1.70
Weight	70	71	60	50	55	80	68	62	69	55	66	70
Shoe	*Few*	*Avg*	*Lot*	*Lot*	*Avg*	*Avg*	*Few*	*Lot*	*Lot*	*Avg*	*Avg*	*Lot*

- Class Probabilities: $\mathbb{P}(\text{Male}) = 1/2 = \mathbb{P}(\text{Female})$
- Height:
  - $(\text{height}) \mid (\text{Male}) \sim \mathbf{N}\left(1.72, 0.05^2\right)$
  - $(\text{height}) \mid (\text{Female}) \sim \mathbf{N}\left(1.63, 0.06^2\right)$
- Weight:
  - $(\text{weight}) \mid (\text{Male}) \sim \mathbf{N}\left(70, 5.8^2\right)$
  - $(\text{weight}) \mid (\text{Female}) \sim \mathbf{N}\left(60, 7.5^2\right)$
- Number of Shoes (Few, Avg, Lot)
  - $(\text{shoes}) \mid (\text{Male}) \sim \text{Multinomial}(0.33, 0.33, 0.33)$
  - $(\text{shoes}) \mid (\text{Female}) \sim \text{Multinomial}(0, 0.5, 0.5)$

Gaussian fit



Male v.s. Female: (height, weight, shoes)

- An individual is $1.7m$ tall, weights $65Kg$ and has an average number of shoes. What is the probability that this individual is a male?

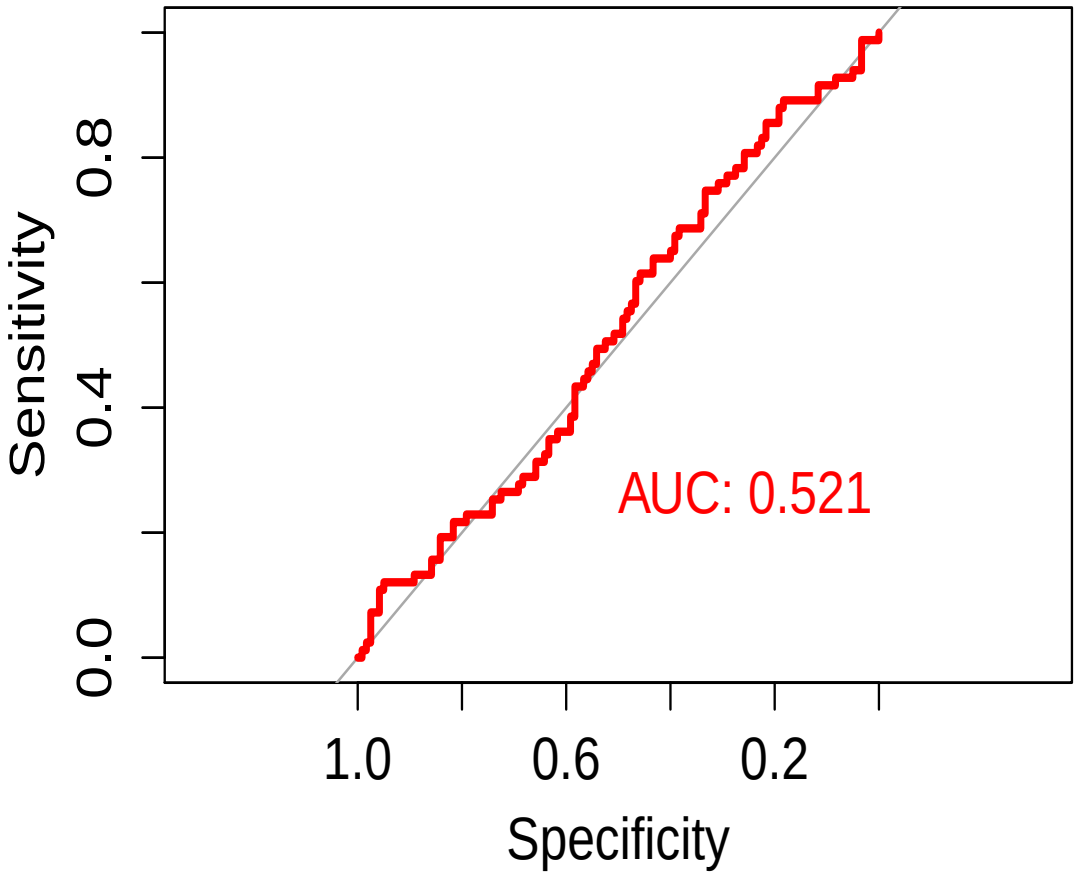
Regularisation

- In the above model, any individual with a **few** number of shoes will automatically be classified as a **Male**. This is clearly not satisfying.
- For a regularisation parameter $c > 0$, one can estimate $\mathbb{P}(x^{(j)} = s \mid y = r)$ by

$$\frac{(\text{Number of training examples where } x^{(j)} = s \text{ and } y = r) + c}{(\text{Total number of training examples with } y = r) + R_j \times c}$$

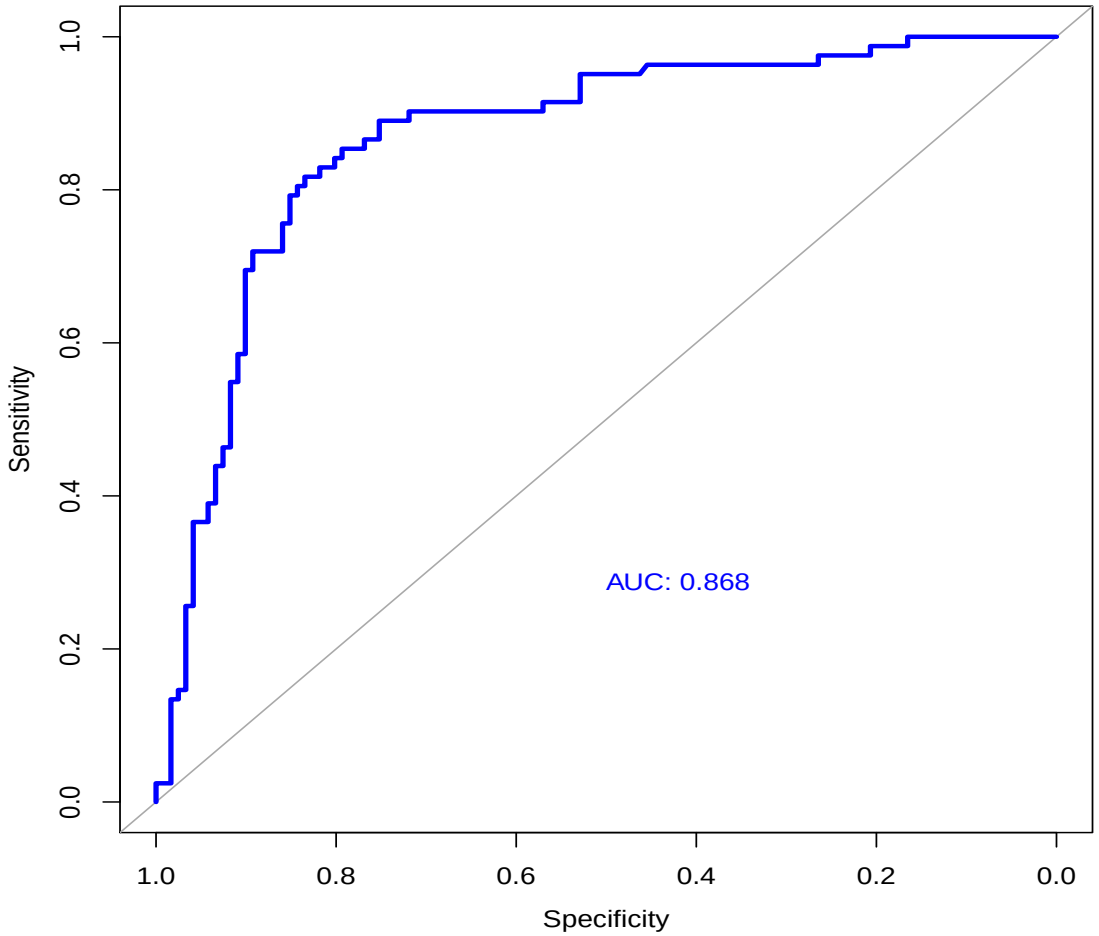
  where R_j is the number of possible values for $x^{(j)}$.
- Can be justifiable by introducing a Dirichlet prior.
- The amount of regularisation can be tuned by cross-validation

So, who is going to survive then?



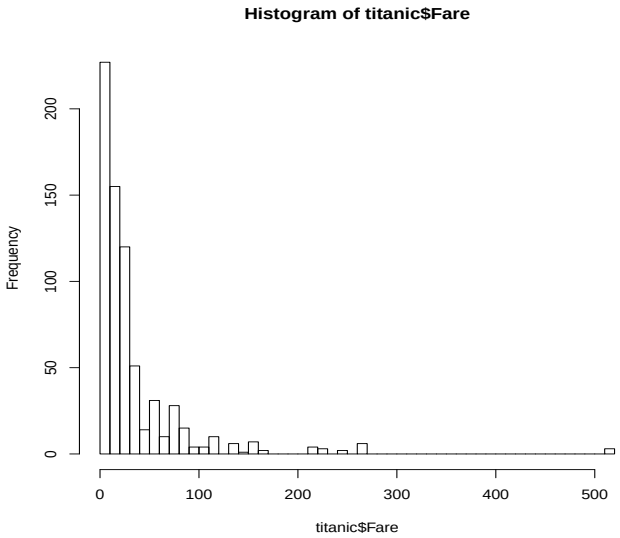
Random guess!

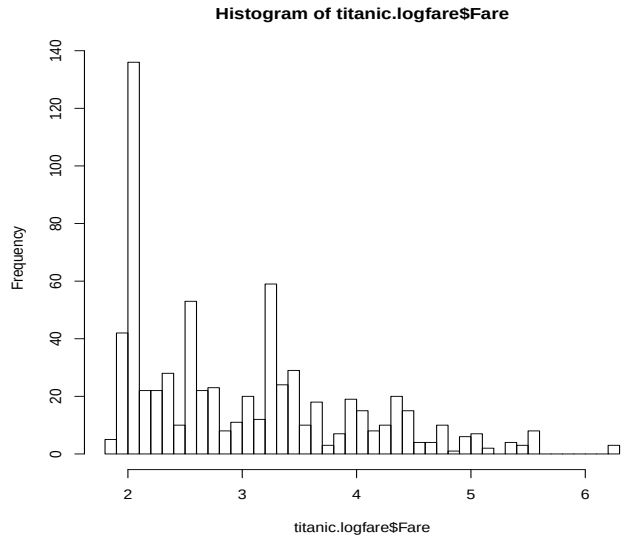
So, who is going to survive then?

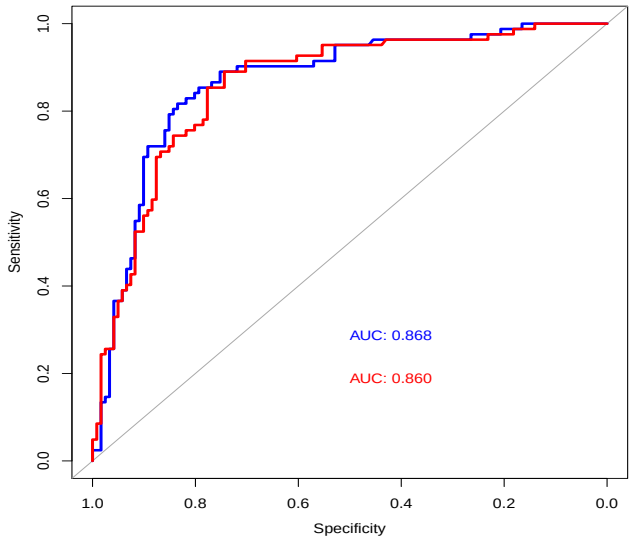


Naive Bayes Classifier!

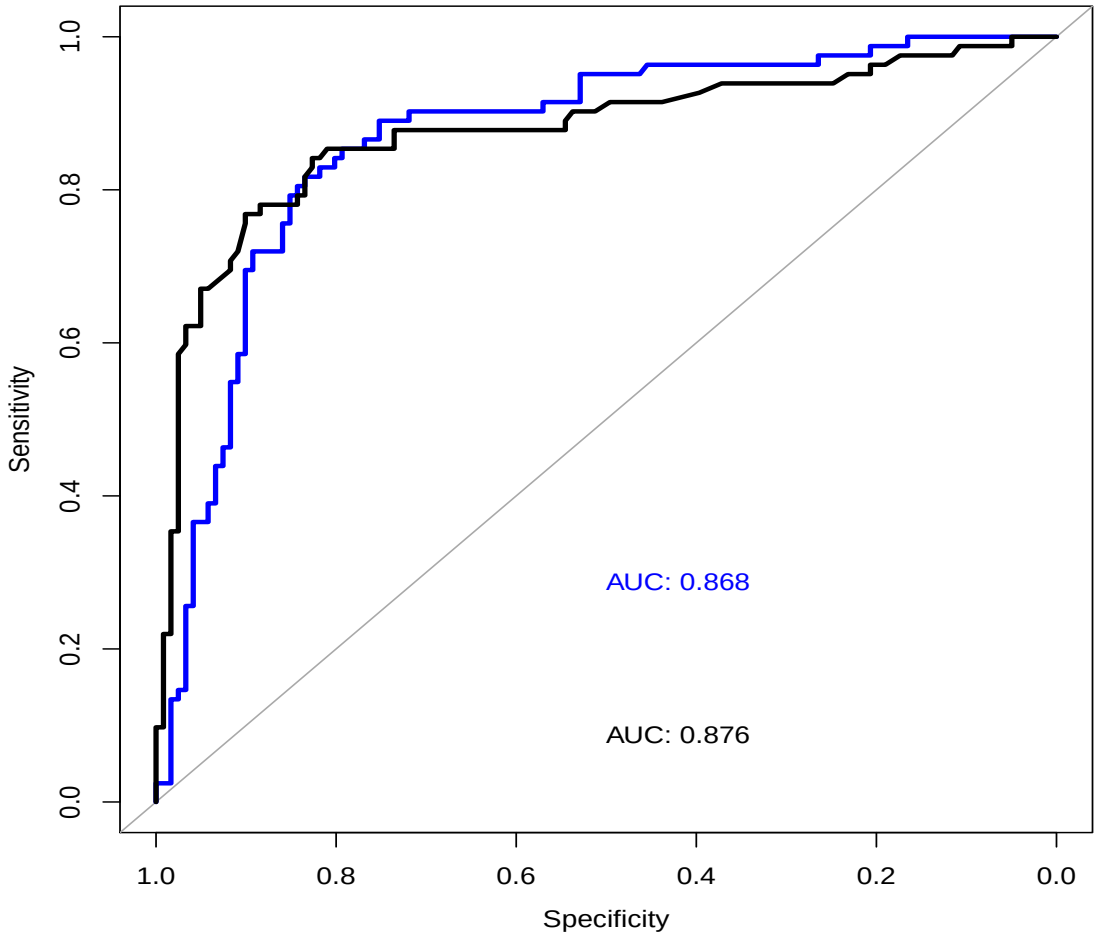
Transformation?







Should we drop some data?



Only three covariates kept: Age, Sex, Pclass

Outline

A linear model

- Feature $x \in \mathbb{R}^p$
- Class $y \in \mathcal{Y} = \{-1, +1\}$
- **Goals**:
  - Generative model that describes the probabilities

$$\mathbb{P}(y = +1 \,|\, x) \qquad \text{and} \qquad \mathbb{P}(y = -1 \,|\, x)$$

  - The model should be linear
  - The following **cannot work**:

$$\mathbb{P}(y = +1 \,|\, x) = \langle \beta, x \rangle$$

A linear model for the log-odds

- Since $\mathbb{P}(y = +1 \,|\, x) = \langle \beta, x \rangle$ does not work, one first needs to transform $\mathbb{P}(y = +1 \,|\, x)$ into something that can take value in $(-\infty, +\infty)$.
- The odds can take any positive value

$$(\text{odds}) \equiv \frac{\mathbb{P}(y = +1 \,|\, x)}{\mathbb{P}(y = -1 \,|\, x)} = \frac{\mathbb{P}(y = +1 \,|\, x)}{1 - \mathbb{P}(y = +1 \,|\, x)}$$

- the log-odds can take any value in $(-\infty, +\infty)$,

$$\log(\text{odds}) \equiv \log\left\{ \frac{\mathbb{P}(y = +1 \,|\, x)}{1 - \mathbb{P}(y = +1 \,|\, x)} \right\}$$

- This can also be written as

$$\log(\text{odds}) \equiv \text{Logit}\left(\mathbb{P}(y = +1 \,|\, x)\right)$$

with $\text{Logit}(x) = \log\left(\frac{x}{1-x}\right)$

A linear model for the log-odds

- In summary, logistic regression assumes that

$$\log(\text{odds}) = \log\left\{\frac{\mathbb{P}(y=+1\,|\,x)}{1-\mathbb{P}(y=+1\,|\,x)}\right\} = \langle\beta, x\rangle$$

  for some vector of parameter $\beta \in \mathbb{R}^p$ and feature $x \in \mathbb{R}^p$.

- **[Exercise]** this can equivalently be expressed as

$$\mathbb{P}(y=+1\,|\,x) = \frac{e^{\langle\beta,x\rangle}}{1+e^{\langle\beta,x\rangle}} \quad \text{and} \quad \mathbb{P}(y=-1\,|\,x) = \frac{1}{1+e^{\langle\beta,x\rangle}}$$

Fitting: maximum likelihood

- Training examples $\{(x_i, y_i)\}_{i=1}^N$
- **Goal:** find the maximum likelihood estimate $\beta_\star$
- The log-likelihood reads [**Exercise**]

$$-\sum_{i=1}^{N} \log\left\{1 + e^{-y_i \langle \beta, x_i \rangle}\right\}$$

- The MLE $\beta_\star$ is solution of the optimization problem

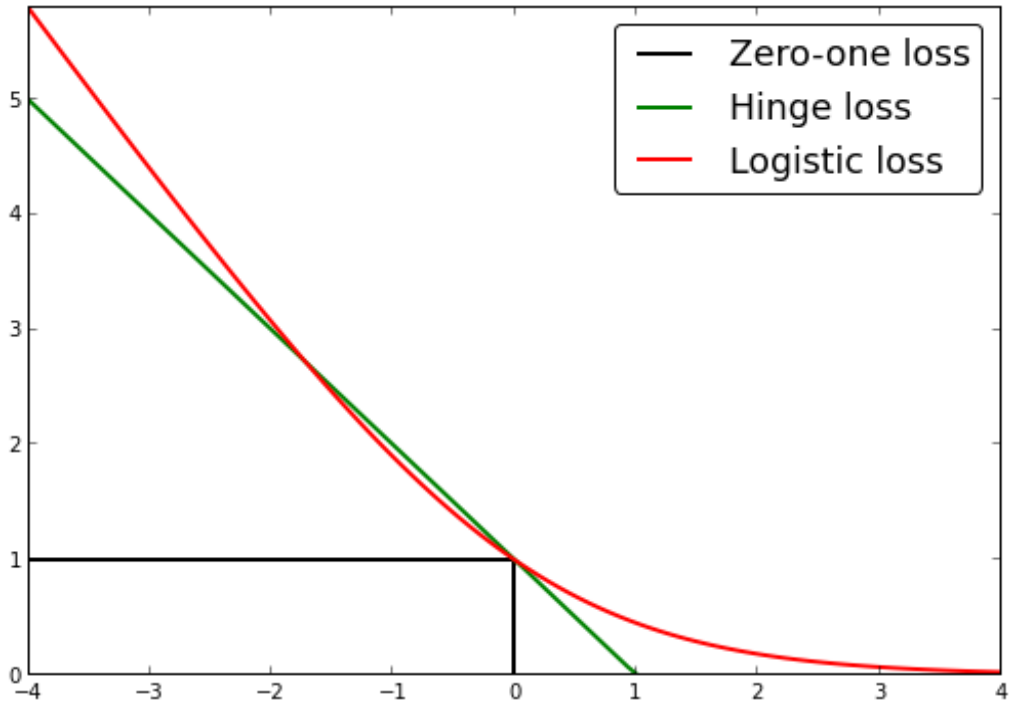
$$\beta_\star = \textbf{argmin}\left\{\beta \mapsto \sum_{i=1}^{N} \textbf{Loss}(\beta, x_i, y_i)\right\}$$

with loss function $\textbf{Loss}(\beta, x_i, y_i) = \log\left\{1 + e^{-y_i \langle \beta, x_i \rangle}\right\}$.

The loss function

$$\textbf{Loss}(\beta, x_i, y_i) = \log\left\{1 + e^{-y_i \langle \beta, x_i \rangle}\right\} = F(y_i \langle \beta, x_i \rangle)$$

- Logistic loss: $F(x) = \log\{1 + e^{-x}\}$
- Find β that makes $\langle \beta, x_i \rangle$ as positive as possible for $y_i = +1$ and $\langle \beta, x_i \rangle$ as negative as possible for $y_i = -1$. In other words, make $y_i \langle \beta, x_i \rangle$ as large as possible.

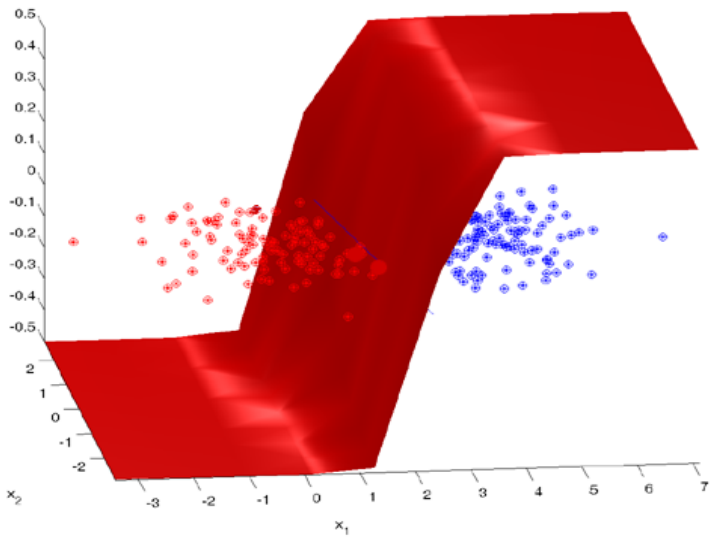


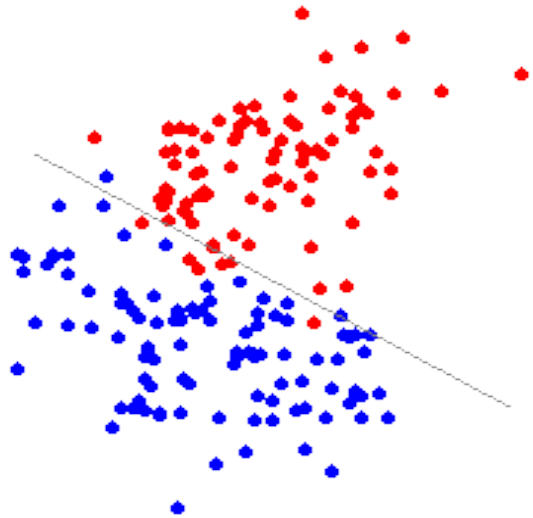
The geometry of the prediction

- Consider the plane

$$\mathcal{H} = \{x \in \mathbb{R}^p \ : \ \langle \beta, x \rangle = 0\}$$

- **Linear separation:** on one side of the plane, $\mathbb{P}(y = +1 \,|\, x) > 1/2$ and on the other side $\mathbb{P}(y = +1 \,|\, x) < 1/2$. **[Exercise]**
- **Overfitting**: if the training dataset are linearly separable, there is no solution.





Regularized logistic regression

- For better prediction performances and avoid pathologies of the MLE estimate, one can consider regularized version of the logistic regression procedure
- **Ridge logistic regression:** an L^2 penalization term is added

$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto \sum_{i=1}^{N} \textbf{Loss}(\beta, x_i, y_i) \; + \; \lambda \, \|\beta\|_2^2 \right\}$$

  Contrarily to ridge regression, there is no closed form solution!
- **Lasso logistic regression:** an L^1 penalization term is added

$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto \sum_{i=1}^{N} \textbf{Loss}(\beta, x_i, y_i) \; + \; \lambda \, \|\beta\|_1 \right\}$$

- [**Exercise**] prove that the above two optimization problems are convex. For ridge logistic regression, it suffices to follow the gradient!

Regularized logistic regression: gradient descent

- **Ridge logistic regression:**

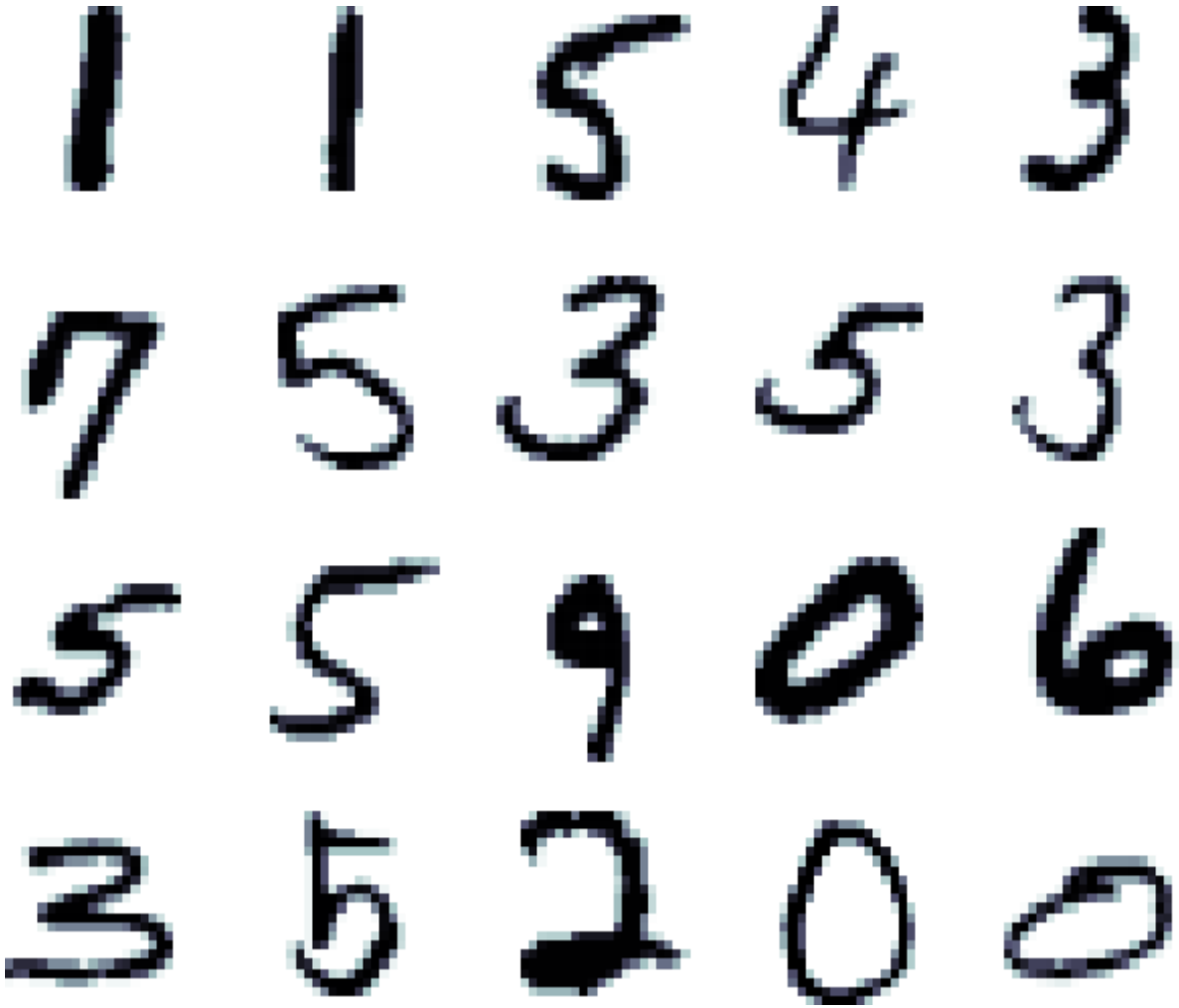
$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto \sum_{i=1}^{N} \textbf{Loss}(\beta, x_i, y_i) \; + \; \lambda \, \|\beta\|_2^2 \right\}$$

  Contrarily to ridge regression, there is no closed form solution!

- [**Exercise**] write the gradient descent update for finding $\beta_\star$.

Outline

Digit recognition



Softmax regression: the model

- N training examples $\{x_i, y_i\}_{i=1}^N$
- p-dimensional covariate $x_i = \left(x_i^{(1)}, \ldots, x_i^{(p)}\right) \in \mathbb{R}^p$
- $C \geq 2$ possible classes: $y_i \in \{r_1, r_2, \ldots, r_C\}$
- Parameters: $\beta_1, \beta_2, \ldots, \beta_C \in \mathbb{R}^p$
- Probabilistic model

$$\mathbb{P}(y = r_k \,|\, x) = \frac{e^{\langle \beta_k, x \rangle}}{\sum_{j=1}^{C} e^{\langle \beta_j, x \rangle}}$$

- **[Exercise]**: For any vector $\nu \in \mathbb{R}^p$, prove that replacing $(\beta_1, \ldots, \beta_C)$ by $(\beta_1 - \nu, \ldots, \beta_C - \nu)$ leads to the same likelihood function. One can thus assume that $\beta_1 = (0, 0, \ldots, 0)$.

Softmax regression: fitting

- Probabilistic model

$$\mathbb{P}(y = r_k \,|\, x) = \frac{e^{\langle \beta_k, x \rangle}}{\sum_{j=1}^{C} e^{\langle \beta_j, x \rangle}}$$

- **[Exercise]**: Prove that finding the MLE estimate $(\beta_2, \ldots, \beta_C)$ is equivalent to maximizing the function

$$\ell(\beta_2, \ldots, \beta_C) \equiv \sum_{i=1}^{N} \sum_{k=1}^{C} \mathbf{1}(y_i = r_k) \log \left\{ \frac{e^{\beta_k, x_i}}{\sum_{j=1}^{C} e^{\langle \beta_j, x_i \rangle}} \right\}$$

  with the convention $\beta_1 = (0, 0, \ldots, 0)$.

- **[Exercise]**: compute the gradient of $\ell(\cdot)$ with respect to β_k.

Outline

Linear model for prediction

- Feature $x \in \mathbb{R}^p$
- Class $y \in \mathcal{Y} = \{-1, +1\}$
- **Goals**:
  - Function $F : \mathbb{R}^p \to \{-1, +1\}$ that predict the class of a new piece of data
  - The model should be linear i.e. only depends on the scalar product $\langle \beta, x \rangle$ between a parameter $\beta \in \mathbb{R}^p$ and the vector of features $x \in \mathbb{R}^p$

$$F(x) = \text{sign}\left(\langle \beta, x \rangle\right) = \begin{cases} +1 & \text{if} \quad \langle \beta, x \rangle \geq 0 \\ -1 & \text{if} \quad \langle \beta, x \rangle < 0. \end{cases}$$

  - Note that we implicitly assumed that the vector of feature $x \in \mathbb{R}^p$ contains one intercept i.e. $x = (1, \cdot, \ldots, \cdot) \in \mathbb{R}^p$.

SVM Loss Function

- Recall **Ridge logistic regression:**

$$\beta_\star = \textbf{argmin}\left\{\beta \mapsto \sum_{i=1}^{N} \textbf{Loss}_{\text{logistic}}(\beta, x_i, y_i) \; + \; \lambda \|\beta\|_2^2\right\}$$

with $\textbf{Loss}_{\text{logistic}}(\beta, x_i, y_i) = F_{\text{logistic}}(y_i \; \langle \beta, x_i \rangle)$ and

$$F(x) = \log\left(1 + e^{-x}\right)$$

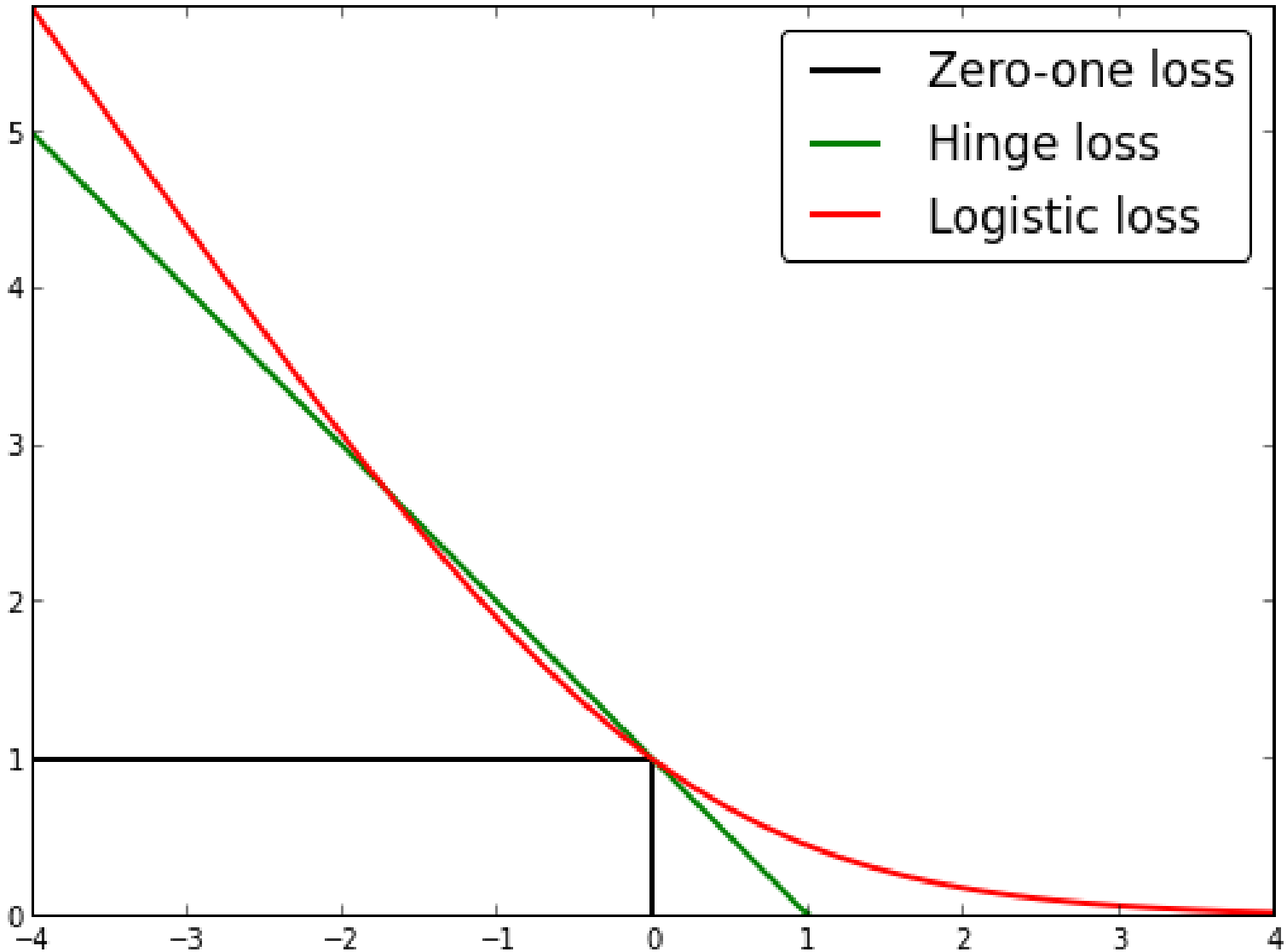
- **Support Vector Machine:**

$$\beta_\star = \textbf{argmin}\left\{\beta \mapsto \sum_{i=1}^{N} \textbf{Loss}_{\text{SVM}}(\beta, x_i, y_i) \; + \; \lambda \|\beta\|_2\right\}$$

with $\textbf{Loss}_{\text{SVM}}(\beta, x_i, y_i) = F_{\text{SVM}}(y_i \; \langle \beta, x_i \rangle)$ and

$$F_{\text{SVM}}(x) = (1 - x)^+$$

SVM v.s. Logistic Loss Functions



SVM Loss Function

- **Support Vector Machine:**

$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto \sum_{i=1}^{N} \textbf{Loss}_{\text{SVM}}(\beta, x_i, y_i) \; + \; \lambda \, \|\beta\|_2 \right\}$$

- Equivalently, and more commonly in practice, the SVM optimization problem is formulated as follows

$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto C \sum_{i=1}^{N} \textbf{Loss}_{\text{SVM}}(\beta, x_i, y_i) \; + \; \|\beta\|_2 \right\}$$

for a parameter $C = 1/\lambda$. A high value of C correspond to small amount of regularization.

Linearly separable case: large margin interpretation

- **Support Vector Machine:**

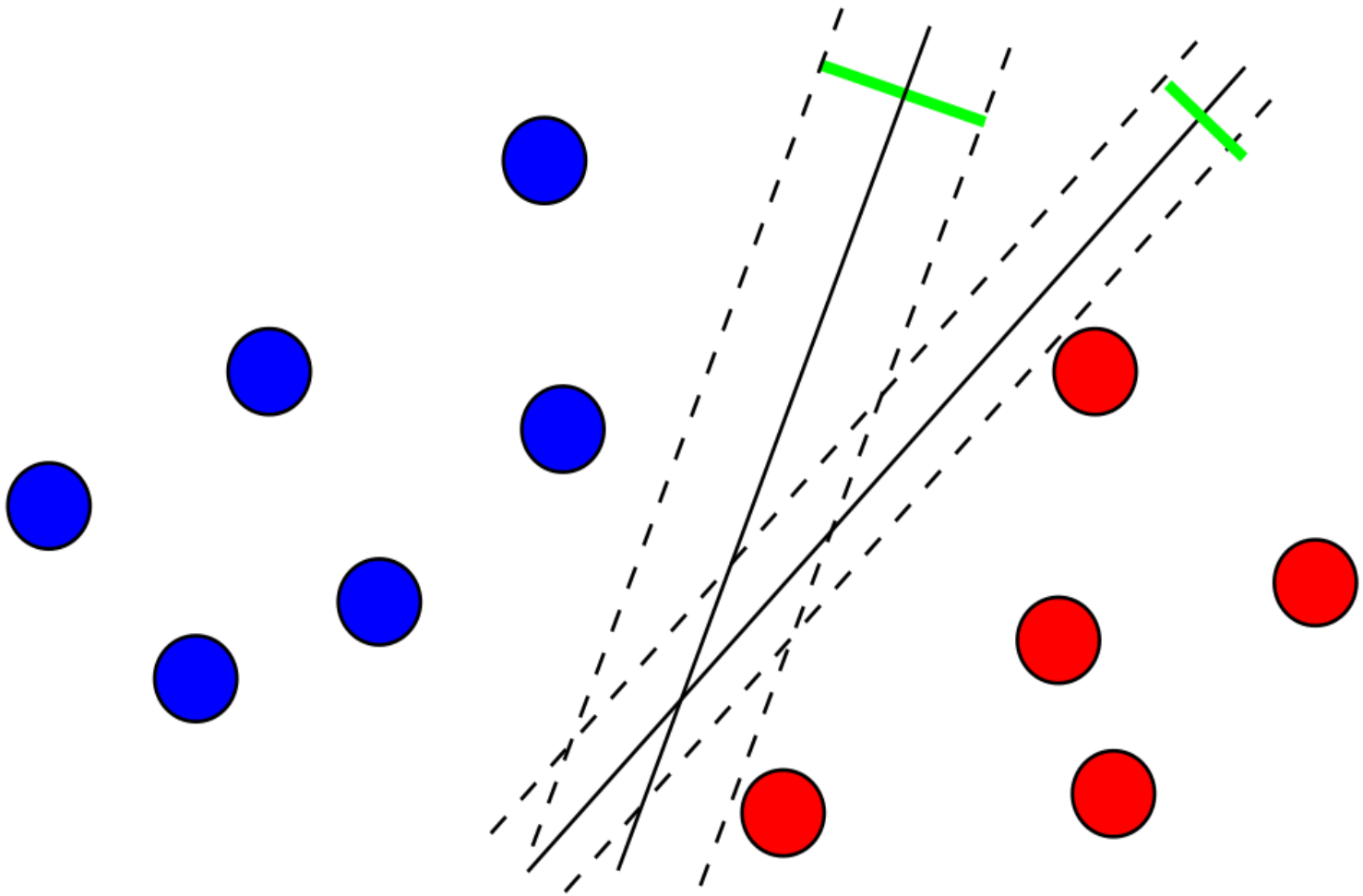
$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto \sum_{i=1}^{N} \textbf{Loss}_{\mathrm{SVM}}(\beta, x_i, y_i) \; + \; \lambda \, \|\beta\|_2 \right\}$$

- Equivalently, and more commonly in practice, the SVM optimization problem is formulated as follows

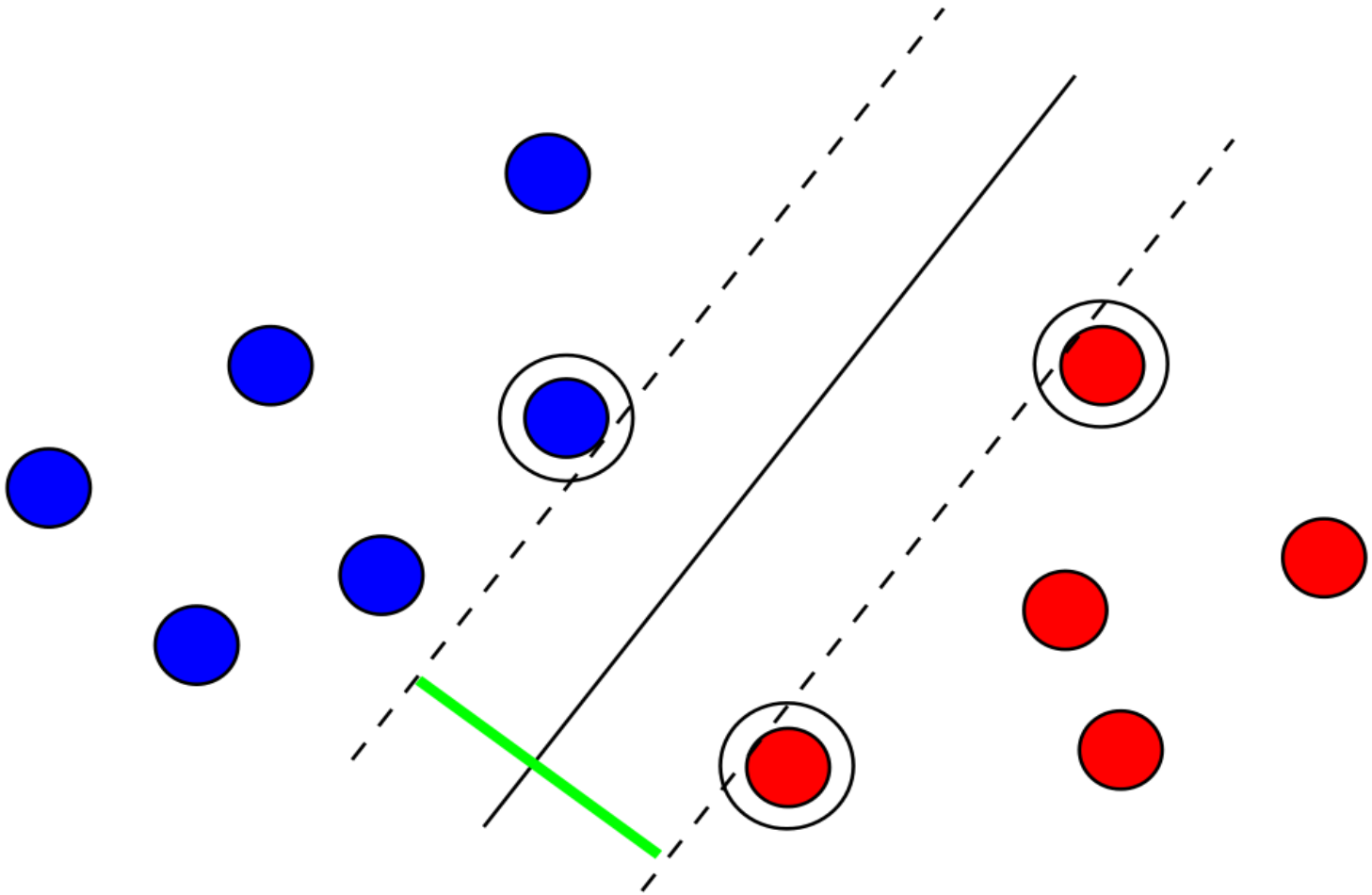
$$\beta_\star = \textbf{argmin} \left\{ \beta \mapsto C \sum_{i=1}^{N} \textbf{Loss}_{\mathrm{SVM}}(\beta, x_i, y_i) \; + \; \|\beta\|_2 \right\}$$

for a parameter $C = 1/\lambda$. A high value of C correspond to small amount of regularization.

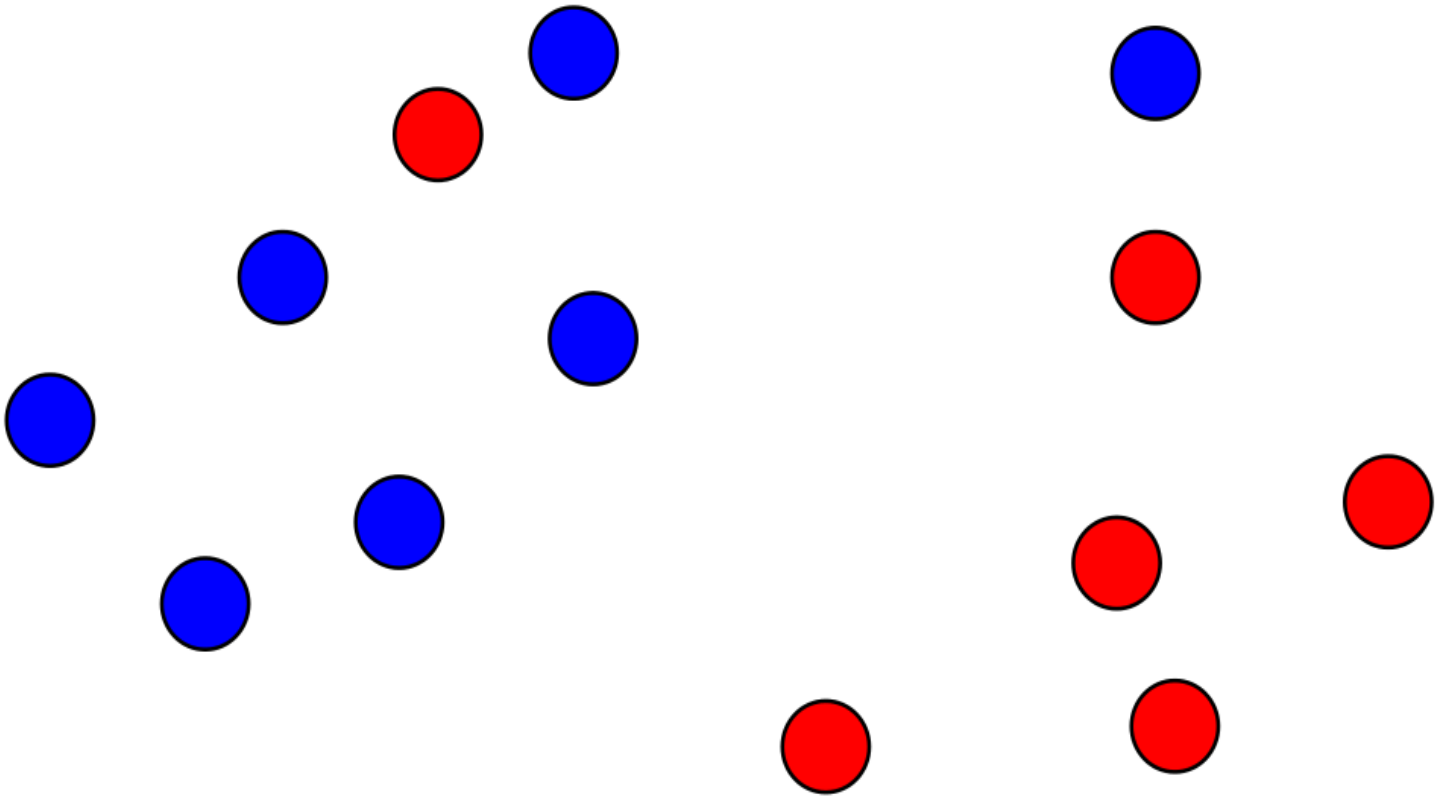
SVM, margins, errors, etc...



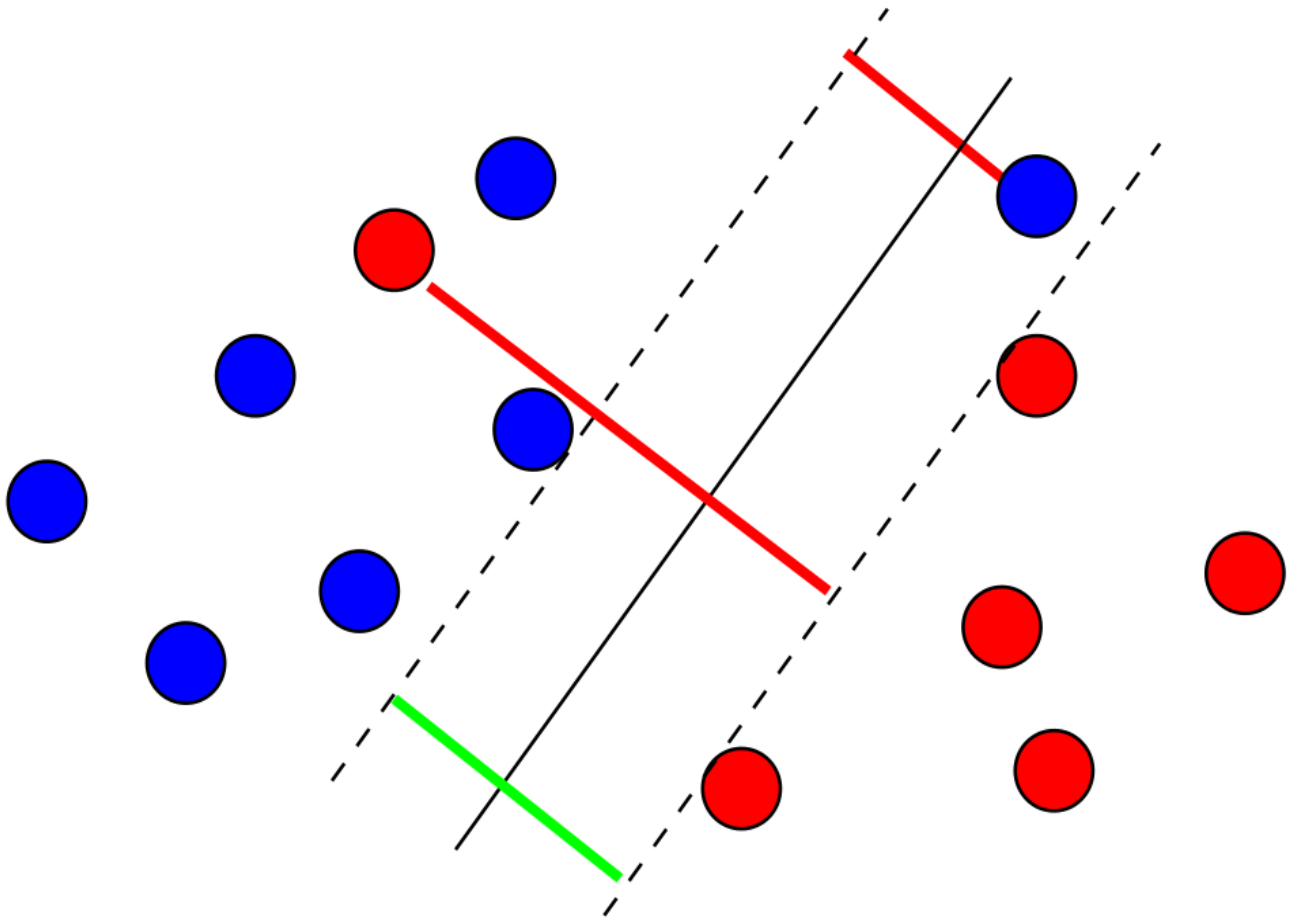
SVM, margins, errors, etc...



SVM, margins, errors, etc...



SVM, margins, errors, etc...



SVM, margins, errors, etc...

- SVM: trade-off between large margin and small error.
- The parameter C represents in some ways the cost of making mistakes. A high value of C indicates that it is very expensive to make mistakes i.e. one wants to avoid mistakes as much as possible.
- Avoiding as much as possible mistakes may lead to poor generalization performances.
- The parameter C is usually found by cross-validation!

Nonlinear Classification