Machine Learning

Ludovic Samper

Antidot

September 1st, 2015

Antidot

Software vendor since 1999

- Paris, Lyon, Aix-en-Provence
- 45 employees
- Founders : Fabrice Lacroix CEO, Stéphane Loesel CTO, Jérôme Mainka Chief Scientist Officer

Software products and solutions

- Antidot Finder Suite (AFS) search engine
- Antidot Information Factory (AIF) a pipe & filters framework
- SaaS, Hosted License, 0n-site License

50% of the revenue invested in R&D

Antidot

Machine Learning

- Automatic text document classification
- Named Entity Extraction
- Compound Splitter (for german words)
- Clustering algorithm (for news agregation)

Open Data, Semantic Web

- `http://www.rechercheisidore.fr/` Social Sciences and Humanities research platform. Enriched with open resources
- `https://github.com/antidot/db2triples/` open source library to export a db in RDF

Antidot is a Partner organization in WDAqua project

Tutorial

- Study a classical task in Machine Learning : text classification
- Show `scikit-learn.org` Python machine learning library
- Follow the "Working with text data" tutorial : `http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html`
- Additional material on `http://blog.antidot.net/`

Summary of the tutorial

1. Problem definition
   - Supervised classification
   - Evaluation metrics
2. Extracting features from text files
   - Bag of words model
   - Term frequency inverse document frequency (tfidf)
3. Algorithms for classification
   - Naïve Bayes
   - Support Vector Machine (SVM)
   - Tuning parameters
     - Cross validation
     - Grid search
4. Conclusion
   - Methodology

Sommaire

1. Problem definition
   - Supervised classification
   - Evaluation metrics
2. Extracting features from text files
3. Algorithms for classification
4. Conclusion

20 newsgroups dataset

`http://qwone.com/~jason/20Newsgroups/`

20 newsgroups

- 20 newsgroups documents collected in the 90's
- The label is the newsgroup the document belongs to
- A popular collection
- 18846 documents : 11314 in train, 7532 in test

wiss-ml.ipynb#The-20-newsgroups-dataset

Classification

Problem statement

- One label per document
- Automatically determine the label of an unseen document. Set of documents and their labels

A supervised classification problem

Training

- Set of documents and their labels

Build a model

Inference

- Given a new document, use the model to predict its label

Precision and Recall I

Binary classification

	C	$\overline{C}$
Labeled C	TP True Positive	FP False Positive
Not labeled C	FN False Negative	TN True Negative

Precision

$$\frac{TP}{TP + FP}$$

Proba($e \in C | e$ labeled C)

Recall

$$\frac{TP}{TP + FN}$$

Proba(e labeled $C | e \in C$)

Precision and Recall II

F_1

$$F_1 = 2\frac{P \times R}{P + R}$$

Harmonic mean of Precision and Recall

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Multiclass I

N_C = number of class

Macro Average

$$B_{macro} = \frac{1}{N_C} \sum_{k=1}^{N_C} (B_{binary}(TP_k, FP_k, TN_k, FN_k))$$

Average mesure by class. Large classes count has much as small ones.

Micro Average

$$B_{micro} = B_{binary}(\sum_{k=1}^{N_C} TP_i, \sum_{k=1}^{N_C} FP_i, \sum_{k=1}^{N_C} TN_k, \sum_{k=1}^{N_C} FN_k)$$

Average mesure by instance

Multiclass II

Micro average in single label multiclass

$$\sum_{k=1}^{N_C}(FN_k) = \sum_{k=1}^{N_C}(FP_k)$$

and

$$\sum_{k=1}^{N_C}(TN_k) = \sum_{k=1}^{N_C}(TP_k)$$

Then,

$$Precision_{micro} = Recall_{micro} = Accuracy = \frac{\sum_{k=1}^{N_C}(TP_k)}{Nbdoc}$$

Sommaire

1. Problem definition
2. Extracting features from text files
   - Bag of words model
   - Term frequency inverse document frequency (tfidf)
3. Algorithms for classification
4. Conclusion

Bag of words

From text to features

- Count the number of occurrences of words in text
- "bag" because position isn't taken into account

Extensions

- Remove stop words
- Remove too frequent words (`max_df`)
- lowercase
- Ngram (`ngram_range`) tokenize ngrams instead of words. Useful to take into account word positions

wiss-ml.ipynb#Bag-of-words

Term frequency inverse document frequency (tfidf) I

Intuition

Take into account relative importance of each word regarding the whole dataset

If a word occurs in every document, it doesn't hold any information

Term frequency inverse document frequency (tfidf) II

Definition

Term frequency $\times$ inverse document frequency

$$\begin{aligned} tfidf(w,d) &= tf(w,d) \times idf(w,d) \\ tf(w,d) &= \text{term frequency}(\text{word } w \text{ in doc } d) \\ idf(w) &= \log(\frac{N_{doc}}{\text{doc freq}(w)}) \end{aligned}$$

In scikit-learn :

$$tfidf(w,d) = tf(w,d) \times (idf(w) + 1)$$

Terms that occurs in all documents $idf = 0$ will not be ignored

Term frequency inverse document frequency (tfidf) III

Options

- Normalisation $||doc|| = 1$. Ex, for norm L_2, $\sum_{w \in d} \text{tfidf}(w, d)^2 = 1$
- Smoothing : add one to document frequencies as if an extra doc contained every term in the collection exactly once

$$idf(w) = \log(\frac{N_{doc} + 1}{\text{doc freq}(w) + 1})$$

Example

Show most significants words of a doc wiss-ml.ipynb#Tfidf

Sommaire

1. Problem definition
2. Extracting features from text files
3. Algorithms for classification
   - Naïve Bayes
   - Support Vector Machine (SVM)
   - Tuning parameters
     - Cross validation
     - Grid search
4. Conclusion

Supervised classification problem I

Notations

- $x = (x_1, \cdots, x_n) = (x_i)_{0 \leq i < n}$ feature vector
- $\{(x^d, y^d)\}_{0 \leq d < D}$ the training set
- $\forall i, x_i \in \mathbb{R}^n$
  - x_i feature vector for document i
  - n dimension of the feature space
- $\forall d, y^d \in \{1, \cdots, N_C\}$
  - N_C the number of classes
  - y^d the class of document d
- $\hat{y}$ class prediction
  For a new vector x, $\hat{y}$ is the predicted class of x.

Supervised classification problem II

Goal

Find a function F :

$$\begin{aligned} \mathbb{R}^n &\rightarrow \{1, \cdots, N_C\} \\ x &\mapsto \hat{y} \end{aligned}$$

In 20newsgroups I

Values in 20 newsgroups

- $n = 130107$ nb features (number of unique terms)
- $D = 11314$ training samples
- $N_C = 20$ different classes

Goal

Find a function F that given a new document predicts its class

Naïve Bayes Algorithm I

Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Naïve Bayes Algorithm II

Posterior probability of class C

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

$P(x)$ does not depend on C,

$$P(C|x) \propto P(x|C)P(C)$$

Naïve Bayes independent assumption : each feature i is conditionally independent of every other feature j

$$P(C|x) \propto P(C) \times \prod_{i=1}^{n} P(x_i|C)$$

Naïve Bayes Algorithm III

Classifier from the probability model

$$\hat{y} = \underset{k \in \{1, \cdots, N_C\}}{\arg\max} \; P(y = k) \times \prod_{i=0}^{n} P(x_i | y = k)$$

Parameter estimation in Naïve Bayes' classifier

Prior of a class

$$P(y = k) = \frac{\text{nb samples in class k}}{\text{total nb samples}}$$

Can also be uniform : $P(y = k) = \frac{1}{N_C}$

Multinomial Naïve Bayes I

Naïve Bayes

$P(x|y = k) = \prod_{i=1}^{n} P(x_i|y = k)$

Multinomial distribution

Event word is i follows a multinomial distribution with parameters $(p_1, \cdots, p_n)$ where $p_i = P(word = i)$

$$P(x_1, \cdots, x_n) = \prod_{i=1}^{n} p_i^{x_i}$$

Where $\sum_i p_i = 1$.
$p_i = P(w = i)$
One distribution for each class y.

Multinomial Naïve Bayes II

Multinomial Naïve Bayes

One multinomial distribution for each class

$$\begin{aligned} P(i|y=k) &= \frac{\text{sum of occurrences of word } x_i \text{ in class } k}{\text{total nb words in class } k} \\ &= \frac{\sum_{d \in k} x_i}{\sum_{0 \leq j < n} \sum_{d \in k} x_j} \end{aligned}$$

With smoothing,

$$P(i|y=k) = \frac{\sum_{d \in k} x_i + \alpha}{\sum_{0 \leq j < n} \sum_{d \in k} x_j + \alpha n}$$

Multinomial Naïve Bayes III

Inference in Multinomial Naïve Bayes

$$
\begin{aligned}
\hat{y} &= \arg\max_k P(y = k|x) \\
&= \arg\max_k P(y = k) \prod_{0 \leq i < n} P(i|y = k)^{x_i} \\
&= \arg\max_k \left(\log(P(y = k)) + \sum_{0 \leq i < n} x_i \log(P(i|y = k)) \right)
\end{aligned}
$$

Multinomial Naïve Bayes IV

A linear model

In the log space,

$$(\log P(y = k|x))_k \propto W_0 + W^T.x$$

W_0, is the vector of priors :

$$W_0 = \log(P(y = k))$$

W is the matrix of distributions :

$$W = (w_{ik}), i \in [1, n], k \in [1, N_C]$$

$$w_{ik} = \log P(i|y = k)$$

Multinomial Naïve Bayes V

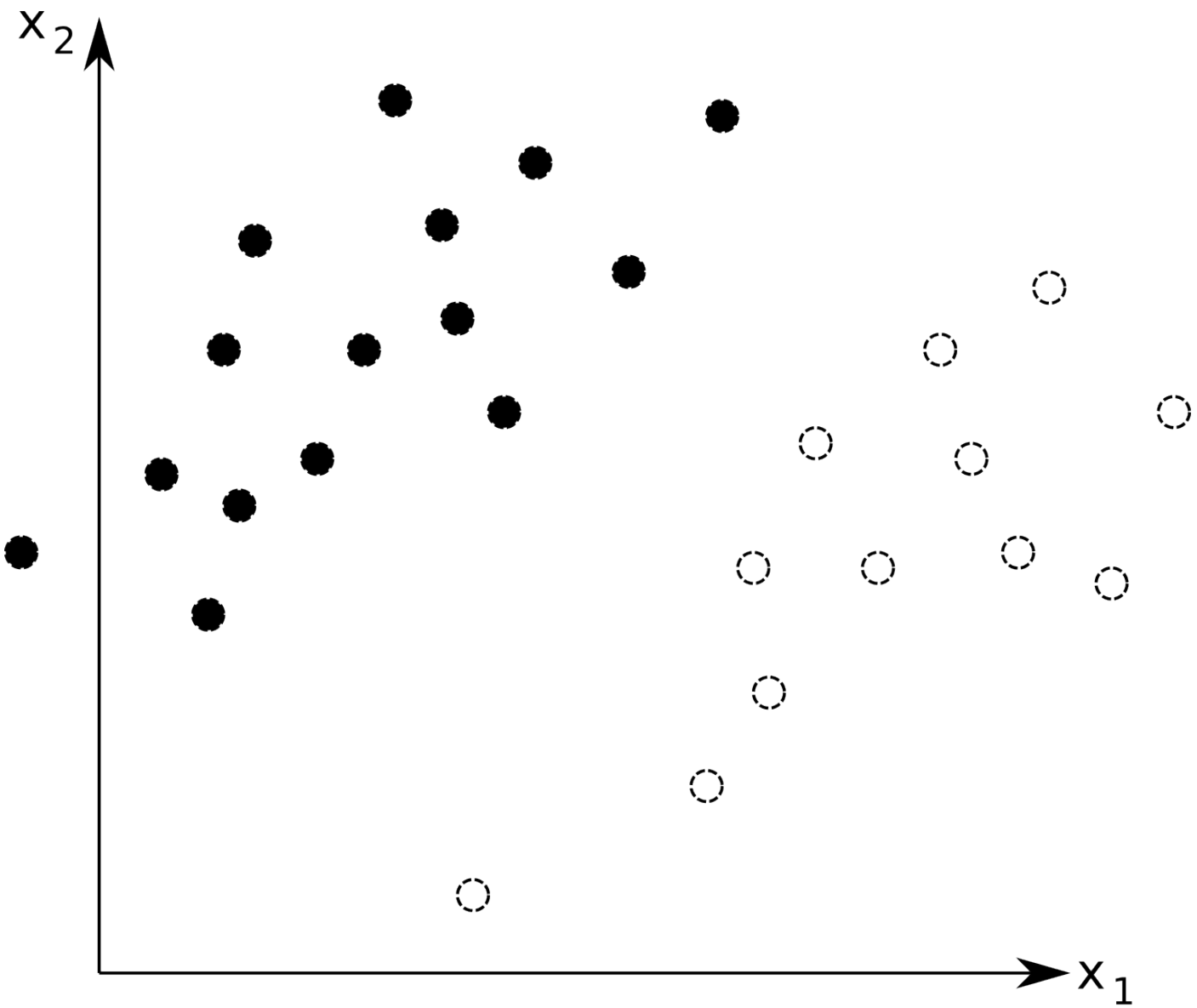
Example step-by-step

`http://www.antidot.net/wiss2015/wiss-ml.html#Naive-Bayes`

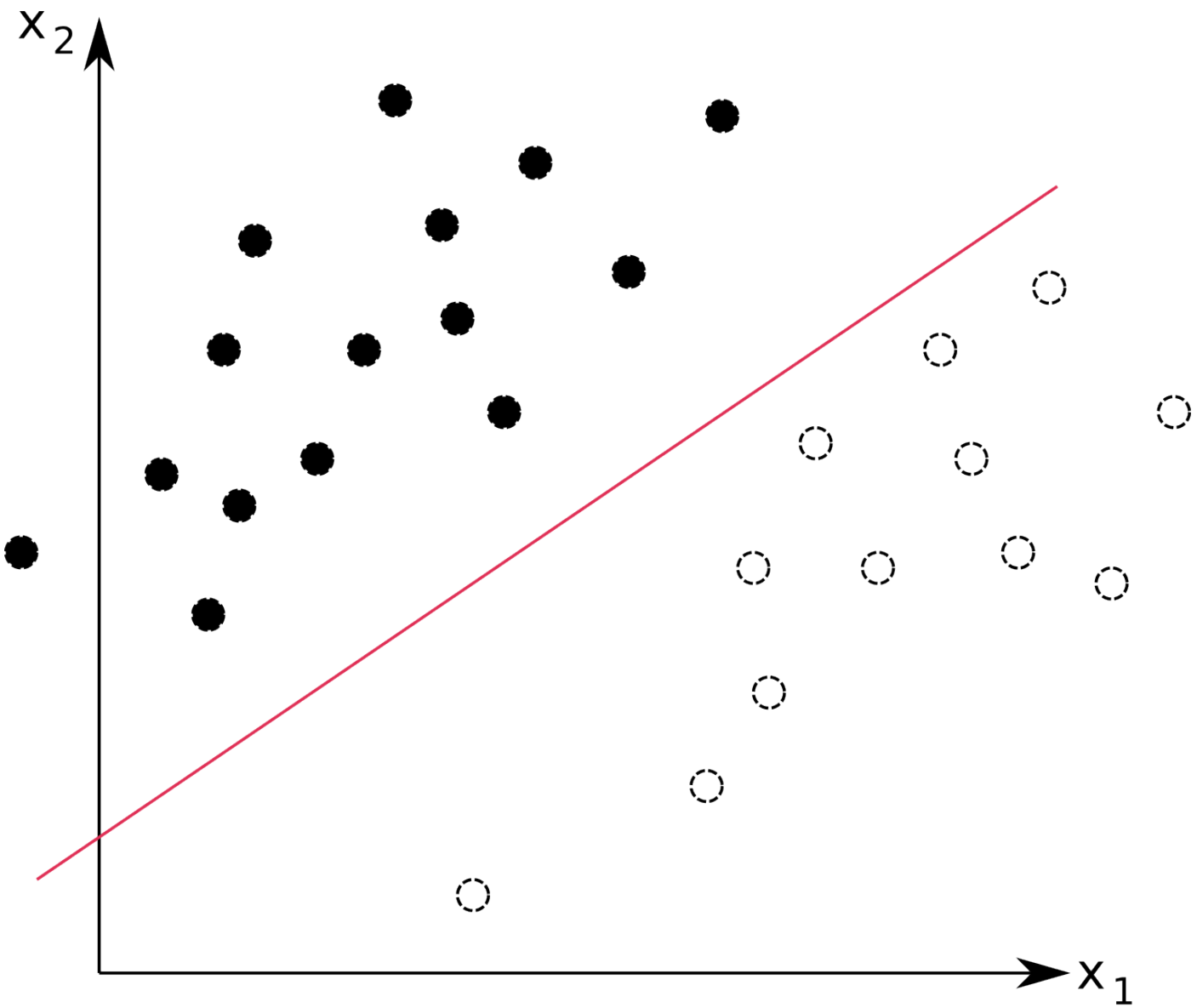
Sommaire

1. Problem definition
2. Extracting features from text files
3. Algorithms for classification
   - Naïve Bayes
   - Support Vector Machine (SVM)
   - Tuning parameters
     - Cross validation
     - Grid search
4. Conclusion

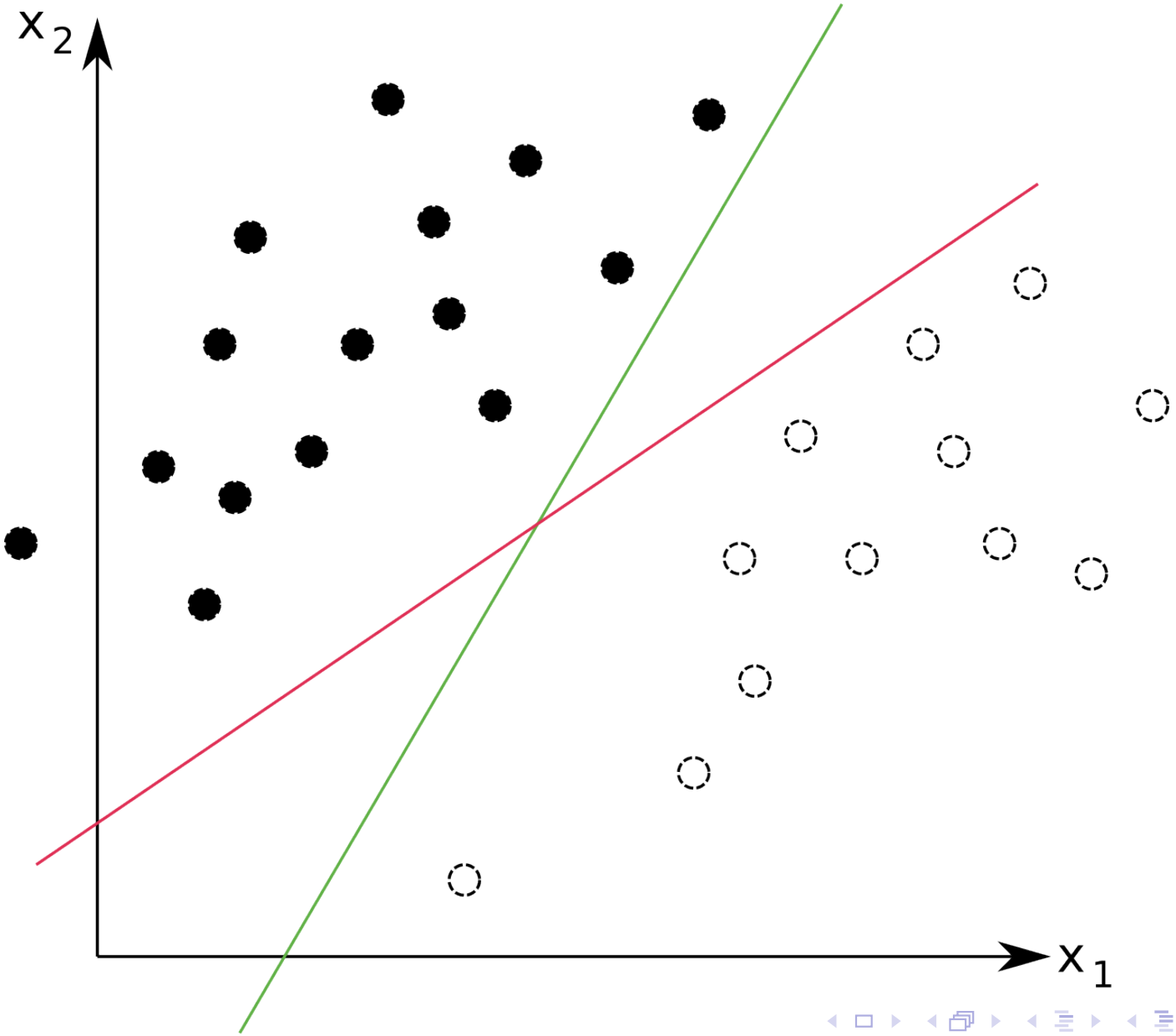
A linear classifier



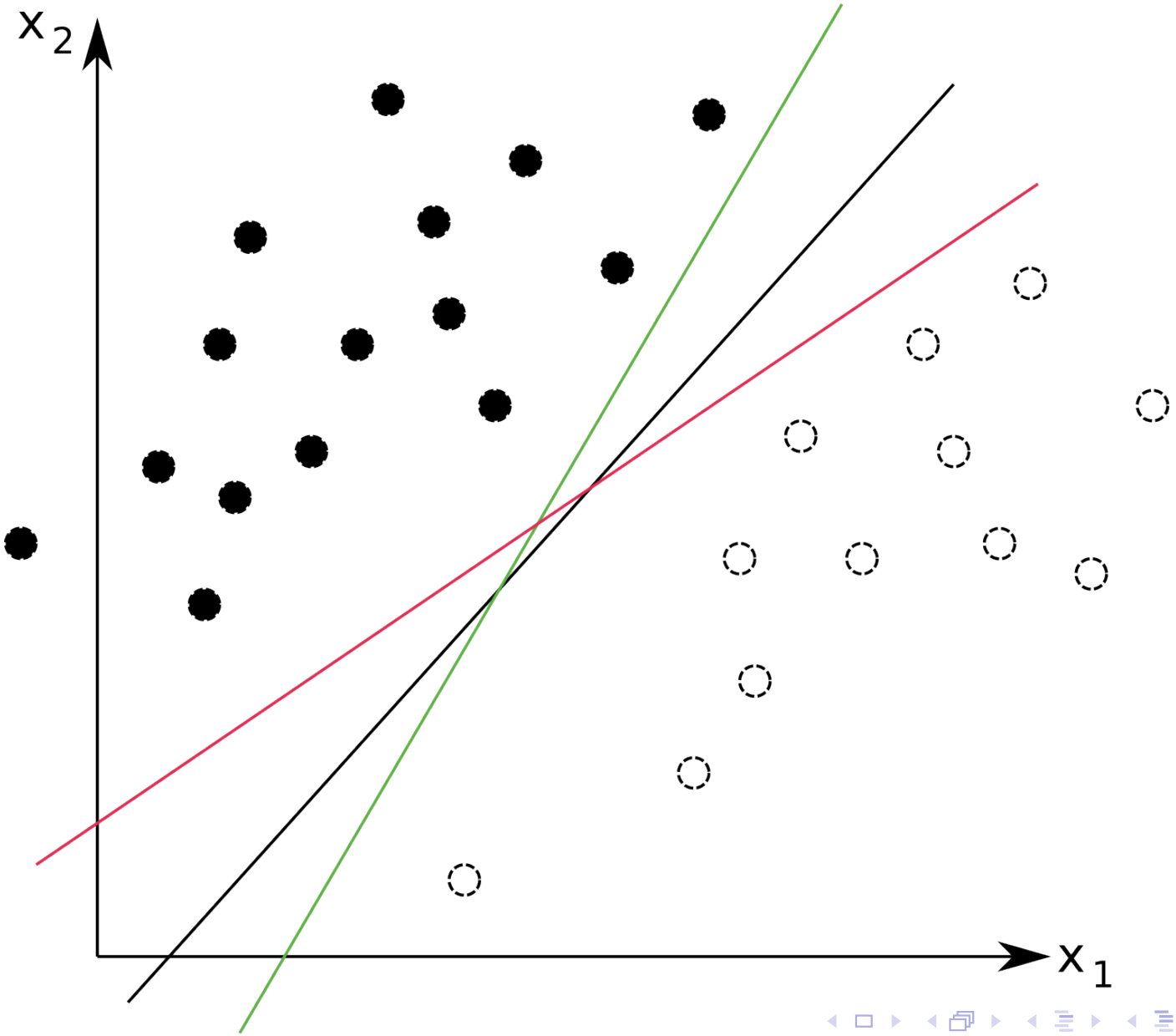
A linear classifier



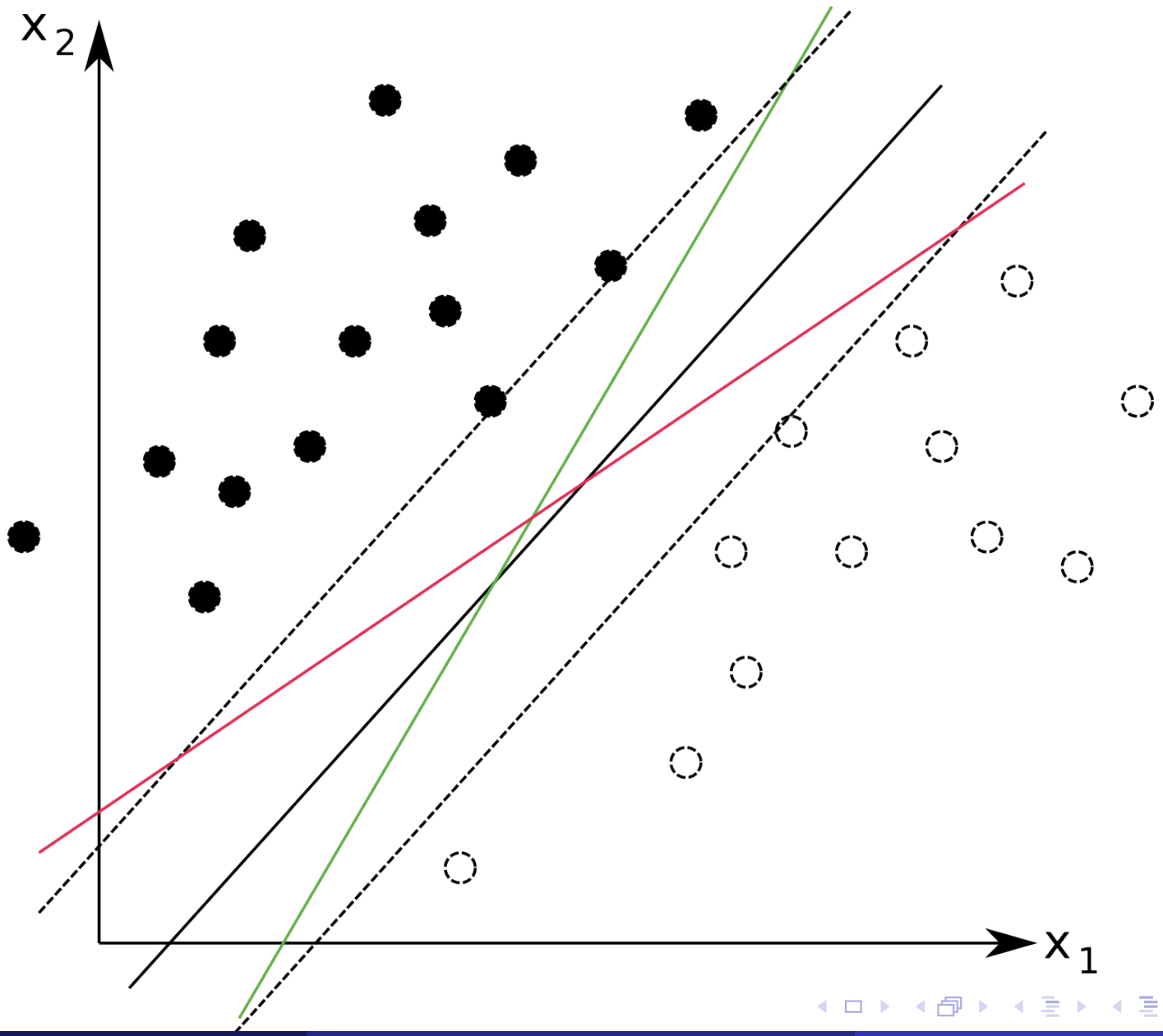
A linear classifier



A linear classifier



A linear classifier



Support Vector Machine, notations

Problem

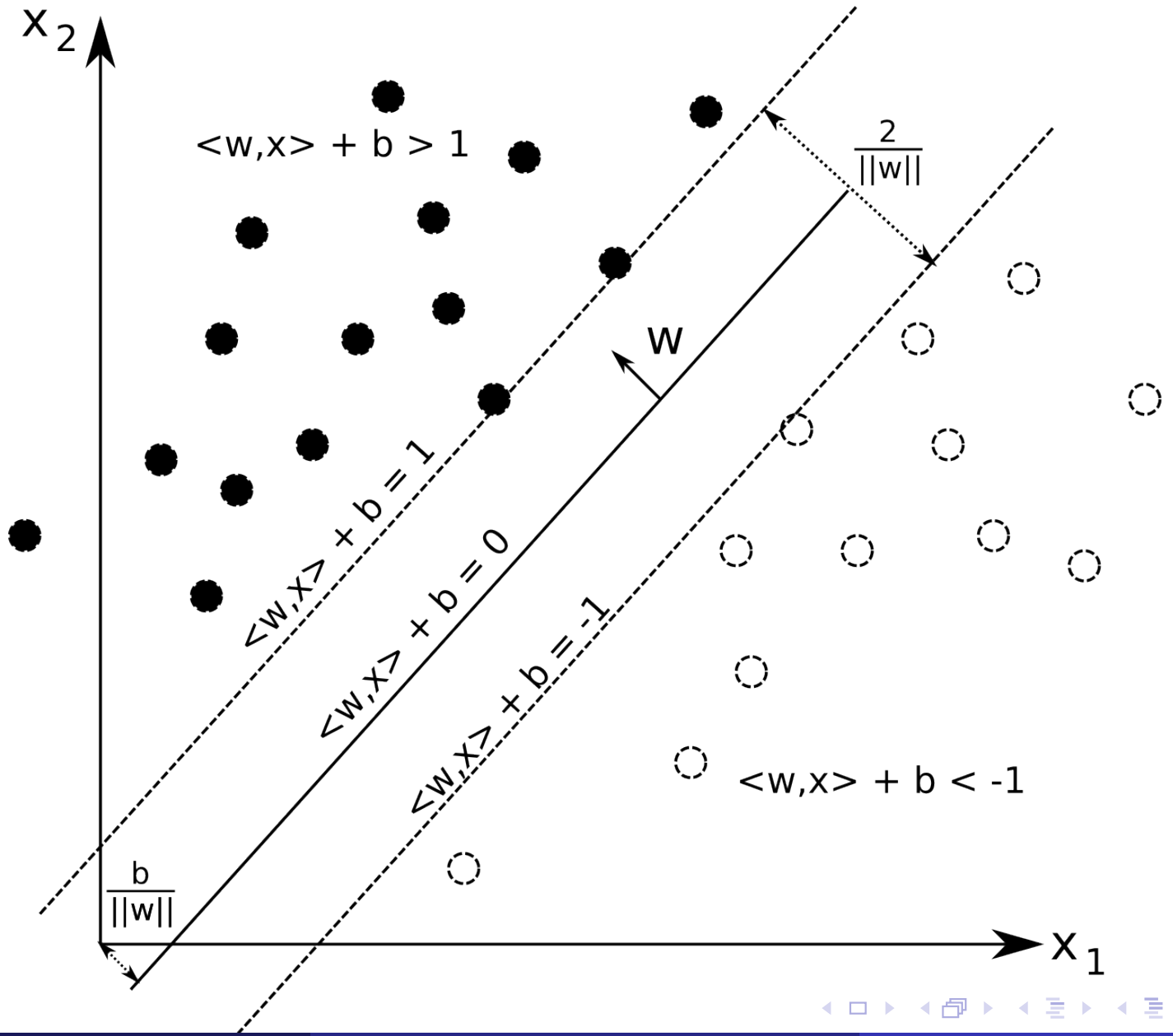
S, training set

$$\{(x_i, y_i), x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}\}_{i \in 0..D}$$

Find a linear function $\langle w, x_i \rangle + b$ such that :

$$sign(\langle w, x_i \rangle + b) = y_i$$

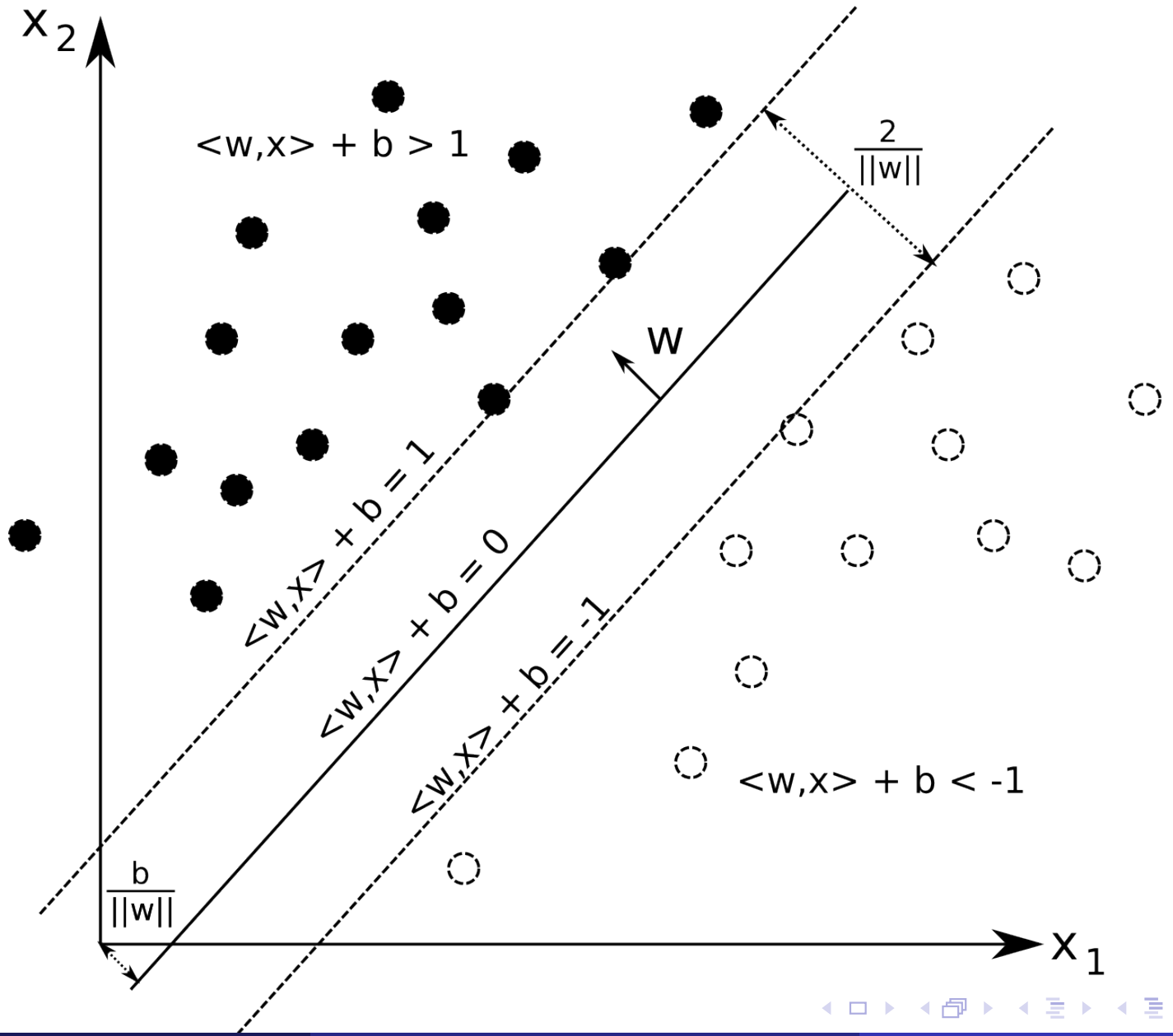
SVM, maximum margin classifier



Margin

$$
\begin{aligned}
\text{distance}(x_+, x_-) &= \langle \frac{w}{||w||}, x_+ - x_- \rangle \\
&= \frac{1}{||w||}(\langle w, x_+ \rangle - \langle w, x_- \rangle) \\
&= \frac{1}{||w||}((\langle w, x_+ \rangle + b) - (\langle w, x_- \rangle + b)) \\
&= \frac{1}{||w||}(1 - (-1)) \\
&= \frac{2}{||w||}
\end{aligned}
$$

SVM, maximum margin classifier



Solving an optimization problem using the Lagrangien

Primal problem

$$minimize_{w,b} f(w,b)$$

Under the constraints, $h_i(w,b) \geq 0$

Lagrange function

$$L(w,b,\alpha) = f(w,b) - \sum_i \alpha_i h_i(w,b)$$

Let, $g(\alpha) = \inf_{(w,b)} L(w,b,\alpha)$
$\forall w,b, g(\alpha) \leq L(w,b,\alpha)$
Moreover, $L(w,b,\alpha) \leq f(w,b)$
Thus, $\forall \alpha_i \geq 0$, $g(\alpha) \leq \min_{w,b} f(w,b)$
And with Karush Kuhn Tucker (KKT) optimality condition,

$$\max_{\alpha} g(\alpha) = \min_{w,b} f(w,b) \Leftrightarrow \alpha_i h_i(w,x) = 0$$

Support Vector Machine, problem

Primal problem

$$minimize_{(w,b)} \frac{||w||^2}{2}$$

Under the constraints, $\forall 0 < i \leq D, y_i(\langle w, x_i \rangle + b) \geq 1$

Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_i \alpha_i(y_i(\langle w, x_i \rangle + b) - 1)$$

Dual problem :

$$maximize_{(w,b,\alpha)} L(w, b, \alpha)$$

with $\alpha_i \geq 0$
Optimality in w, b is a saddle point with α

Support Vector Machine, problem

Derivative in w, b need to vanish

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_i \alpha_i y_i x_i = 0$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = \sum_i \alpha_i y_i = 0$$

Dual problem

$$maximize_\alpha \; -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

under the constraints,

$$\begin{cases} \sum_i \alpha_i y_i = 0 \\ \alpha_i \geq 0 \end{cases}$$

Support Vectors

Support vectors

$$w = \sum_i y_i \alpha_i x_i$$

Karush Kuhn Tucker (KKT) optimality condition

Lagrange multiplier times constraint equals zero

$$\alpha_i (y_i (\langle w, x_i \rangle + b) - 1) = 0$$

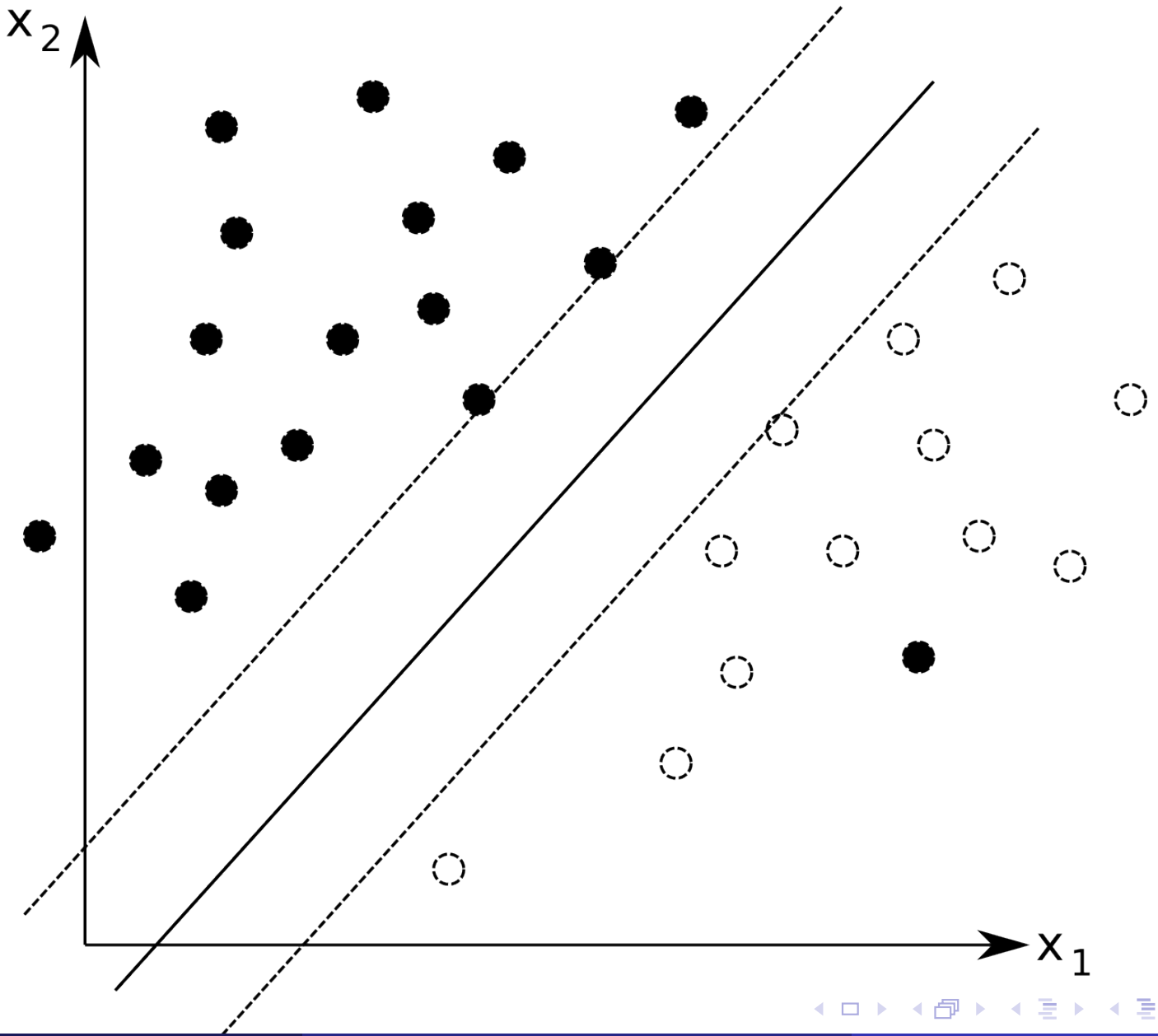
Thus,

$$\begin{cases} \alpha_i = 0 \\ \alpha_i > 0 \Rightarrow y_i (\langle w, x_i \rangle + b) = 1 \end{cases}$$

Experiments with separable space

SVMvaryingC.ipynb

What happens if space is not separable



Adding slack variable

Problem was

$$minimize_{(w,b)} \frac{||w||^2}{2}$$

With,

$$y_i(w.x_i + b) \geq 1$$

With slack

$$minimize_{(w,b)} \frac{||w||^2}{2} + C \sum_i \xi_i$$

With,

$$\begin{cases} y_i(w.x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

Support Vector Machine, without slack

Primal problem

$$minimize_{(w,b)} \frac{||w||^2}{2}$$

With,

$$y_i(w.x_i + b) \geq 1$$

Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_i \alpha_i(y_i(\langle w, x_i \rangle + b) - 1)$$

Dual problem :

$$maximize_{(w,b,\alpha)} L(w, b, \alpha)$$

Optimality in w, b, is a saddle point with α

Support Vector Machine, with slack

Primal problem

$$minimize_{(w,b)} \frac{||w||^2}{2} + C \sum_i \xi_i$$

With,

$$\begin{cases} y_i(w.x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

Lagrange function

$$L(w, b, \xi, \alpha, \eta) = \frac{1}{2}||w||^2 + C \sum_i \xi_i - \sum_i \alpha_i (y_i(\langle x_i, w \rangle + b) + \xi_i - 1) - \sum_i \eta_i \xi_i$$

Dual problem :

$$maximize_{(w,b,\xi,\alpha,\eta)} L(w, b, \xi, \alpha, \eta)$$

Optimality in w, b, ξ is a saddle point with α, η

Support Vector Machine, problem

Derivative in w, b, ξ need to vanish

$$\frac{\partial}{\partial w} L(w, b, \xi, \alpha, \eta) = w - \sum_i \alpha_i y_i x_i = 0$$

$$\frac{\partial}{\partial b} L(w, b, \xi, \alpha, \eta) = \sum_i \alpha_i y_i = 0$$

$$\frac{\partial}{\partial \xi} L(w, b, \xi, \alpha, \eta) = C - \alpha_i - \eta_i = 0 \Rightarrow \eta_i = C - \alpha_i$$

Dual problem

$$maximize_\alpha - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

under the constraints, $\sum_i \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$

Support Vectors

Support vectors

$$w = \sum_i y_i \alpha_i x_i$$

Karush Kuhn Tucker (KKT) optimality condition

Lagrange multiplier times constraint equals zero

$$\alpha_i (y_i (\langle w, x_i \rangle + b) + \xi_i - 1) = 0$$

$$\eta_i \xi_i = 0 \Leftrightarrow (C - \alpha_i)\xi_i = 0$$

Thus,

$$\begin{cases} \alpha_i = 0 \Rightarrow y_i (\langle w, x_i \rangle + b) \geq 1 \\ 0 < \alpha_i < C \Rightarrow y_i (\langle w, x_i \rangle + b) = 1 \\ \alpha_i = C \Rightarrow y_i (\langle w, x_i \rangle + b) \leq 1 \end{cases}$$

Support Vector Machine, Loss functions

Primal problem

$$minimize_{(w,b)} \frac{||w||^2}{2} + C \sum_i \xi_i$$

With,

$$\begin{cases} y_i(w.x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

With loss function

$$minimize_{(w,b)} \frac{||w||^2}{2} + C \sum_i \max(0, 1 - y_i(w.x_i + b))$$

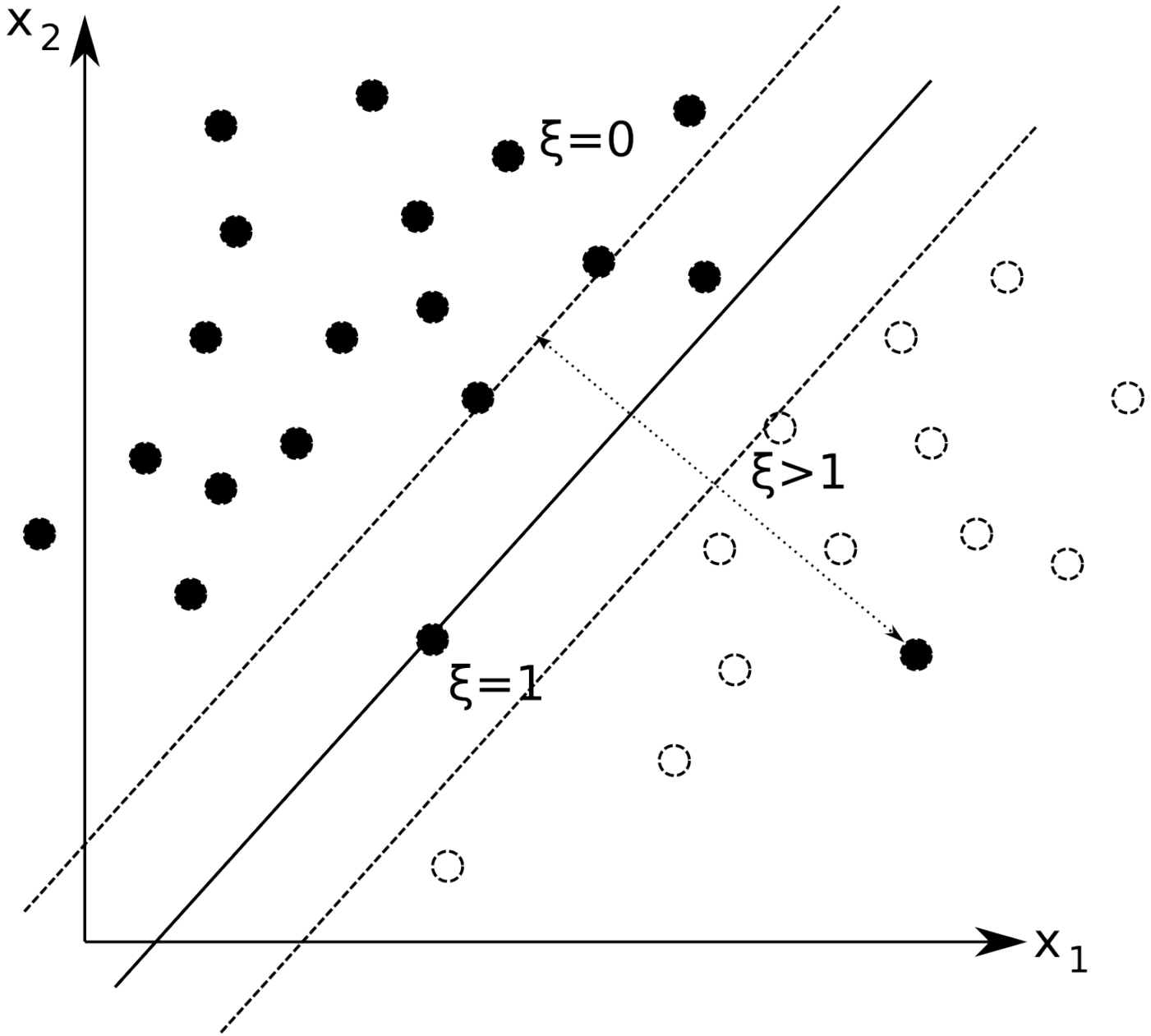
here,

$$loss(x_i, y_i) = \max(0, 1 - y_i(w.x_i + b)) = \max(0, 1 - f(x_i))$$

Support Vector Machine, Common loss functions

Common loss functions

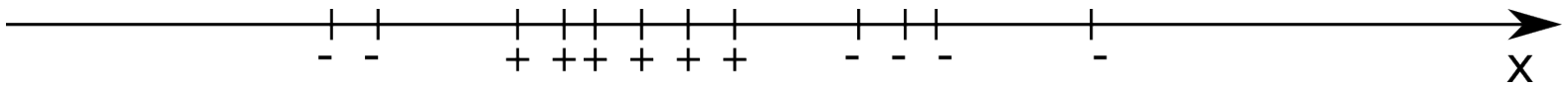
- hinge loss, L_1-loss : $\max(0, 1 - y_i(w.x_i + b))$
- squares hinge L_2-loss : $\max(0, (1 - y_i(w.x_i + b))^2)$
- logistic loss : $\log(1 + \exp(-y_i(w.x_i + b)))$


x 2
ξ=0
ξ>1
ξ=1
x 1

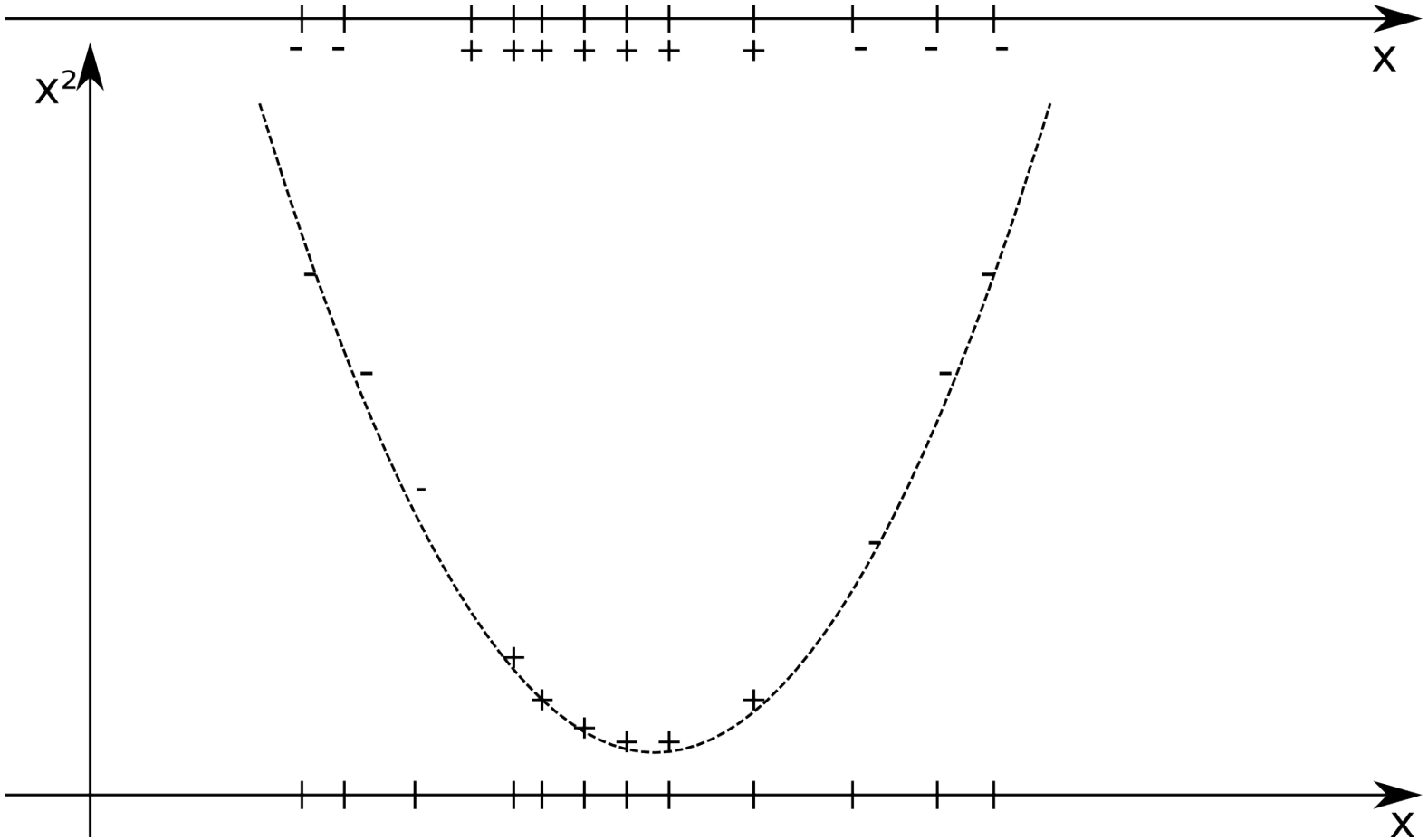
Expermiments with different values for C

SVMvaryingC.ipynb#Varying-C-parameter

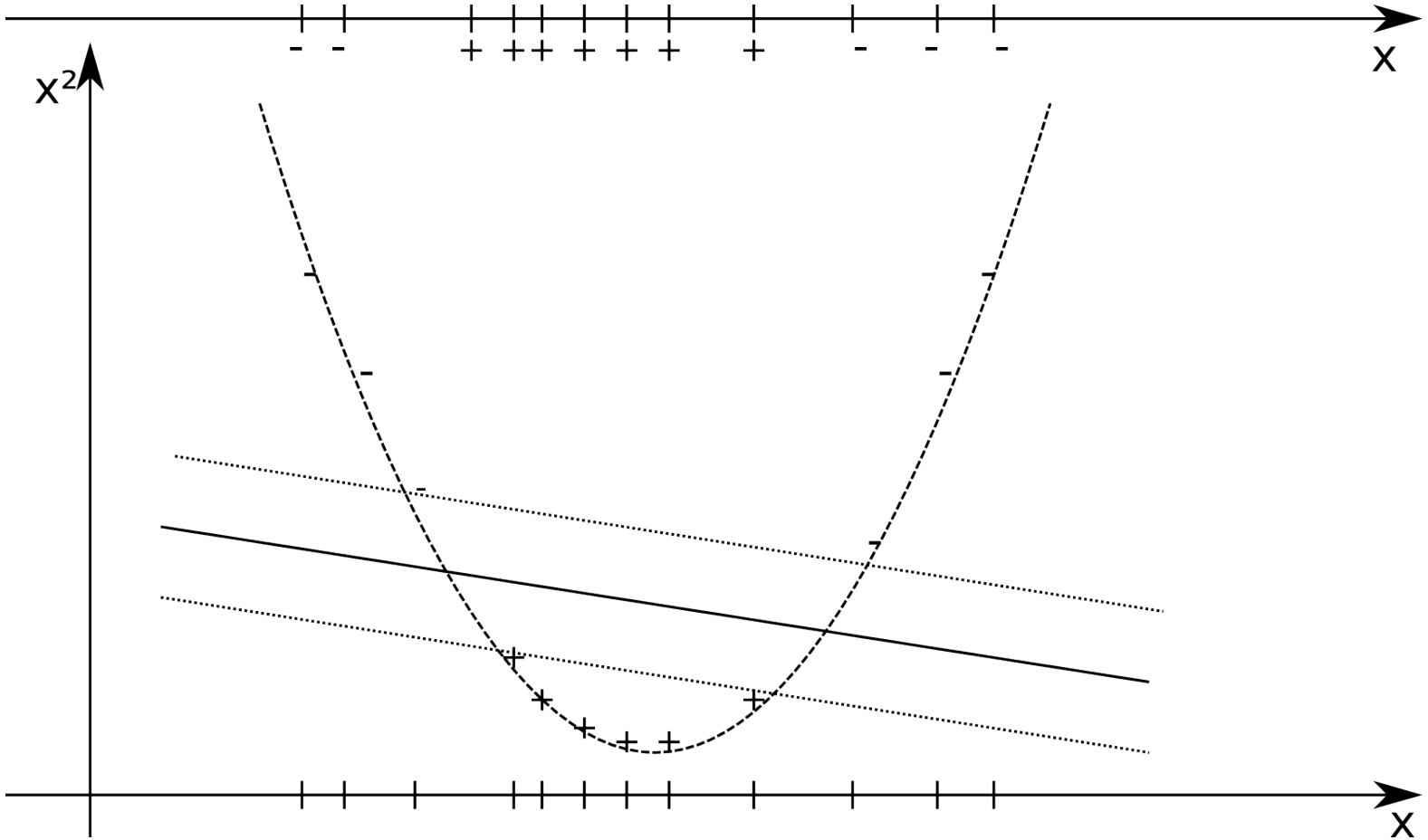
Non linearly separable data



Non linearly separable data, $\Phi(x) = (x, x^2)$



Non linearly separable data, $\Phi(x) = (x, x^2)$



Linear case

Primal Problem

$$minimize_{w,b} \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

subject to, $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

Dual Problem

$$maximize_{\alpha} \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_i \alpha_i$$

subject to, $\sum_i \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$

Support vector expansion

$$f(x) = \sum_i \alpha_i y_i \langle x_i, x\rangle + b$$

With a transformation $\Phi : x \mapsto \Phi(x)$

Primal Problem

$$minimize_{w,b} \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

subject to, $y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

Dual Problem

$$maximize_{\alpha} \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j)\rangle + \sum_i \alpha_i$$

subject to, $\sum_i \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$

Support vector expansion

$$f(x) = \sum_i \alpha_i y_i \langle \Phi(x_i), \Phi(x)\rangle + b$$

The kernel trick

Kernel function

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

We just need to compute the dot product in the new space

Dual Problem

$$maximize_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_i \alpha_i$$

subject to, $\sum_i \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$

Support vector expansion

$$f(x) = \sum_i \alpha_i y_i k(x_i, x) + b$$

Kernels

Kernel functions

- linear : $k(x, x') = \langle x, x' \rangle$
- polynomial : $k(x, x') = (\gamma \langle x, x' \rangle + r)^d$
- rbf : $k(x, x') = \exp(-\gamma |x - x'|^2)$

RBF Kernel imply an infinite space

Here we're in dimension 1, $x \in \mathbb{R}$

$$\begin{aligned} k(x, x') &= exp(-(x - x')^2) \\ &= exp(-x^2)exp(-x'^2)exp(2xx') \end{aligned}$$

With Taylor transformation,

$$\begin{aligned} k(x, x') &= exp(-x^2)exp(-x'^2) \sum_{k=0}^{\infty} \frac{2^k x^k x'^k}{k!} \\ &= \langle (\cdots, \frac{2^{k-1}}{\sqrt{k!}} \exp(-x^2)x^k, \cdots), \\ &\quad (\cdots, \frac{2^{k-1}}{\sqrt{k!}} \exp(-x'^2)x'^k, \cdots) \rangle \end{aligned}$$

Experiments with different kernels

www.antidot.net/wiss2015/SVMvaryingC.html#Non-linear-kernels

SVM in multiclass

one-vs-the rest

- N_C binary classifiers (but each involving all dataset)
- At prediction time, choose the class with maximum decision value

one-vs-one

- $\frac{N_C(N_C-1)}{2}$ binary classifiers
- At prediction time, vote

SVM in scikit-learn

SVC : Support Vector Classification

sklearn.svm.linearSVC

- based on Liblinear library
- strategy : one-vs-the rest
- only linear kernel
- loss can be : 'hinge' or 'squared_hinge'

sklearn.svm.SVC

- based on libSVM
- multiclass strategy : one-vs-one
- kernel can be : linear, polynomial, RBF, sigmoid, precomputed
- only hinge loss

Sommaire

1. Problem definition
2. Extracting features from text files
3. Algorithms for classification
   - Naïve Bayes
   - Support Vector Machine (SVM)
   - Tuning parameters
     - Cross validation
     - Grid search
4. Conclusion

Cross validation I

http://scikit-learn.org/stable/modules/cross_validation.html

Overfitting

Estimation of parameters on the test set can lead to overfitting : parameters are the best for this test set but not in the general case.

Train, test and validation dataset

A solution :

- tweak the parameters on the test set
- validate on a validation dataset
- only few data in training dataset

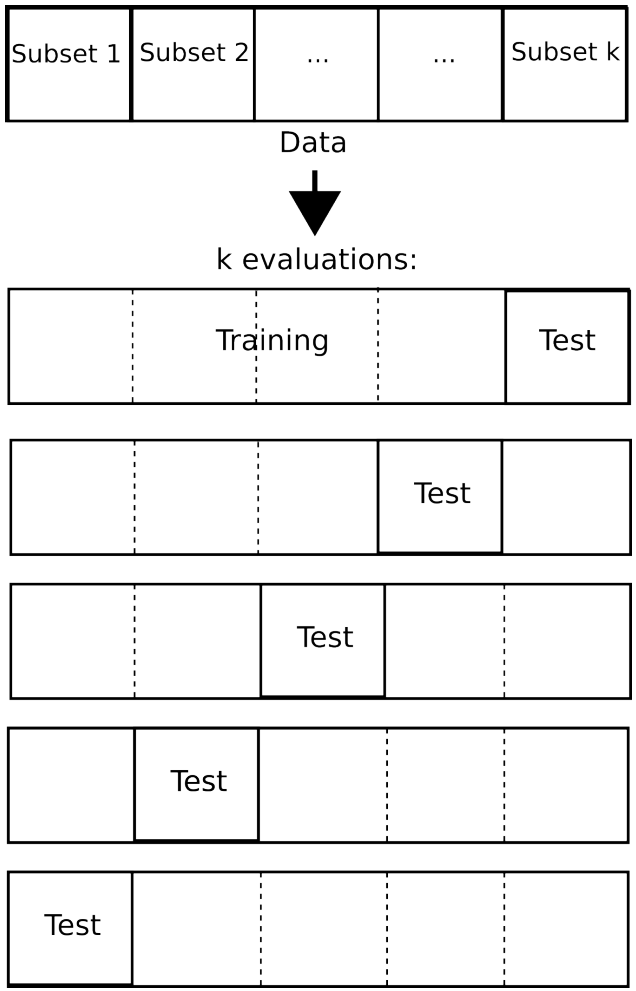
Cross validation II

Cross validation

k-fold cross validation

- Split training data in k partitions of the same size
- train the model on $k-1$ partitions
- then, evaluate on the kth partition

Cross validation III



Grid Search

http://scikit-learn.org/stable/modules/grid_search.html

Grid search

- Test each value for each parameter
- brut force algorithm to find the best value for each parameter

In scikit-learn

- Automatically runs $k\times$ number of parameters' values trainings
- Keeps the best model

Demo with scikit-learn

http://www.antidot.net/wiss2015/grid_search_20newsgroups.html

Sommaire

1. Problem definition
2. Extracting features from text files
3. Algorithms for classification
4. Conclusion
   - Methodology

1. Problem definition
   - Supervised classification
   - Evaluation metrics
2. Extracting features from text files
   - Bag of words model
   - Term frequency inverse document frequency (tfidf)
3. Algorithms for classification
   - Naïve Bayes
   - Support Vector Machine (SVM)
   - Tuning parameters
     - Cross validation
     - Grid search
4. Conclusion
   - Methodology

Methodology

To solve a problem using Machine Learning, you have to :

1. Understand the data
2. Choose an evaluation measure
3. Be able to test the model
4. Find the main features
5. Try the algorithms, with different parameters

Conclusion

- Machine Learning has a lot of applications
- With libraries like scikit-learn, no need to implement algorithms yourself

Questions ?

References

Machine Learning in Python :

`http://scikit-learn.org`

Alex Smola very good lecture on Machine Learning at CMU :

- `http://alex.smola.org/teaching/10-701-15/`
- Kernels : `https://www.youtube.com/watch?v=0Nis-oMLbDs`
- SVM : `https://www.youtube.com/watch?v=bsbpqNIKQzU`

Bernoulli Naïve Bayes

Features

$x_i = 1$ iff word i is present in document

Else, $x_i = 0$

The number of occurrences of word i doesn't matter

Bernoulli

For each feature i,

$P(x_i|y = k) = P(i|y = k)x_i + (1 - P(i|y = k))(1 - x_i)$

Absence of a feature is explicitly taken into account

Estimation of $P(i|y = k)$

$$P(i|y = k) = \frac{1 + \text{nb of documents in } k \text{ that contains word } i}{\text{nb of documents in } k}$$