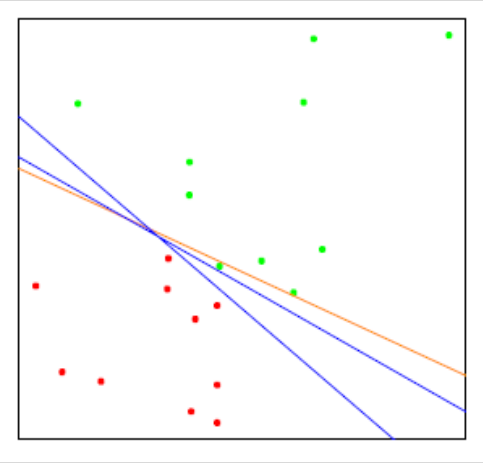


Parallel & Scalable Data Analysis

Introduction to Machine Learning Algorithms

Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 4

Classification Challenges & Solutions

November 24th, 2017
Ghent, Belgium

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH
FORSCHUNGSZENTRUM

Review of Lecture 3

- Supervised Classification
  - Finding a target distribution for realistic learning situations
  - Assume unknown probability distribution over the input space
  - Hypothesis search with M models and we pick one
- Statistical Learning Theory

(ideal function)

$\mathbf{x} = (x_1, ..., x_d)$ ← Probability Distribution P on X

Hypothesis Set $\mathcal{H} = \{h\};\ g \in \mathcal{H}$

Final Hypothesis $g \approx f$

(set of candidate formulas)

'Approximately' 'Probably' 'Number of samples N important for learning'

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\ \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

Learning Algorithm ('train a system') $\mathcal{A}$

(set of known algorithms)

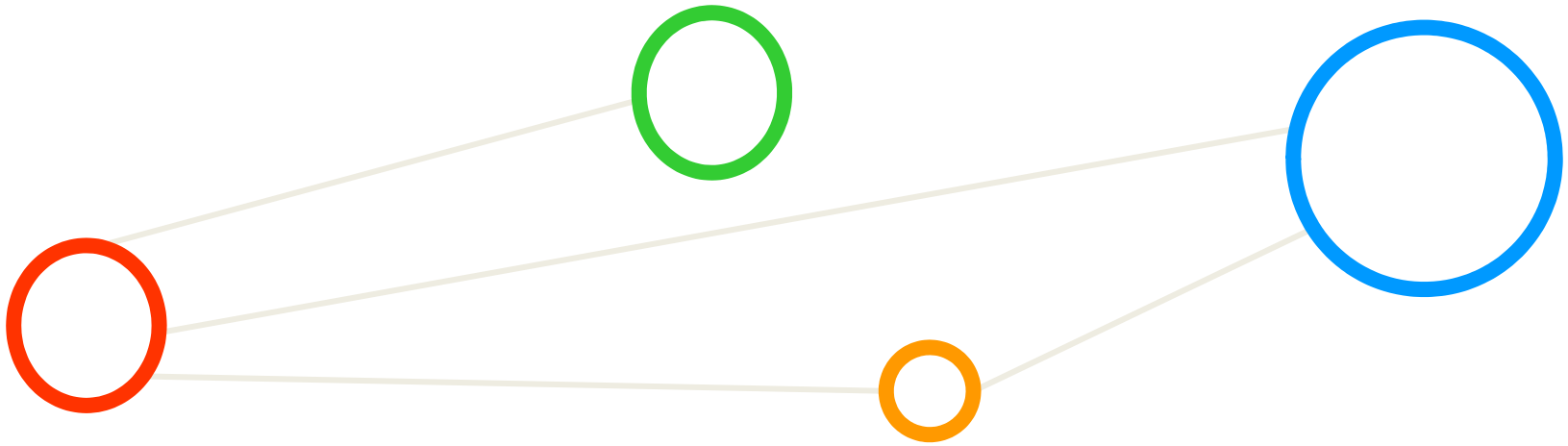
'learn: get error smaller' $E_{in}(g) \approx 0$

'generalize well for unseen data' $E_{out}(g) \approx E_{in}(g)$

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\ \epsilon\ ]\ <=\ 4m_{\mathcal{H}}(2N)e^{-1/8\epsilon^2 N}$$

Shift the view to the data and we exchange M with growth function that is indeed depending on N

Outline



Outline of the Course

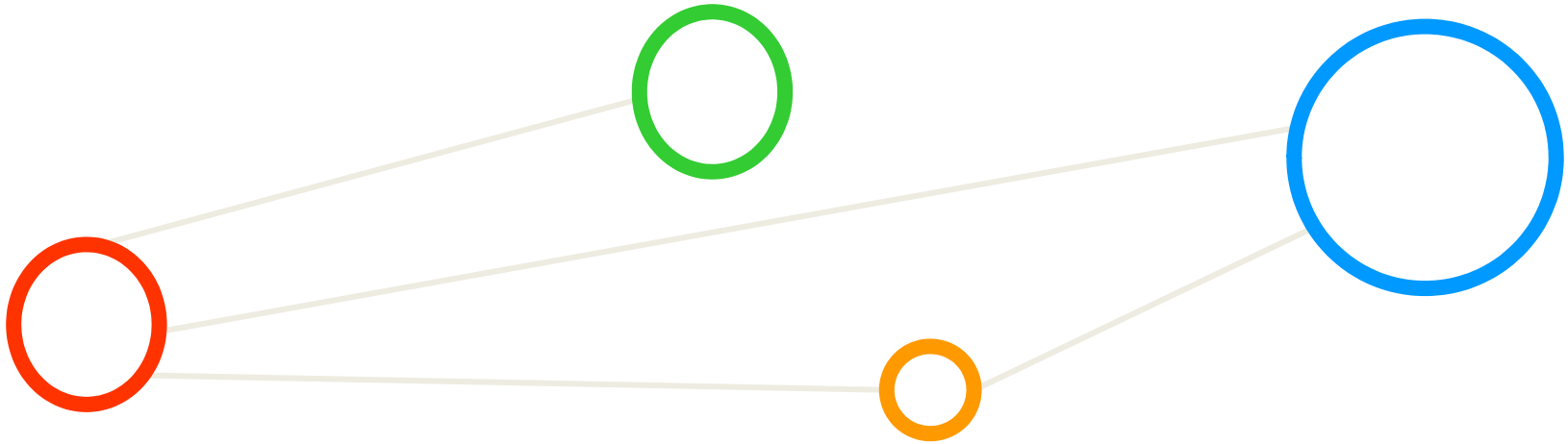
1. Machine Learning Fundamentals
2. Unsupervised Clustering and Applications
3. Supervised Classification and Applications
4. Classification Challenges and Solutions
5. Regularization and Support Vector Machines
6. Validation and Parallelization Benefits

Outline

- Classification Challenges & Solutions
  - Review Practice Experience & Applications
  - Challenge One: Non-Linear Seperable Datasets
  - Challenge Two: Problem of Overfitting
  - Regularization Approach
  - Supervised Classification Models Overview

- Maximal Margin Classifier
  - Term Support Vector Machines Refined
  - Apply Classifier to Datasets
  - Margin as Geometric Interpretation
  - Optimization Problem & Implementation
  - Solving and Limitations of Classifier

Classification Challenges & Solutions



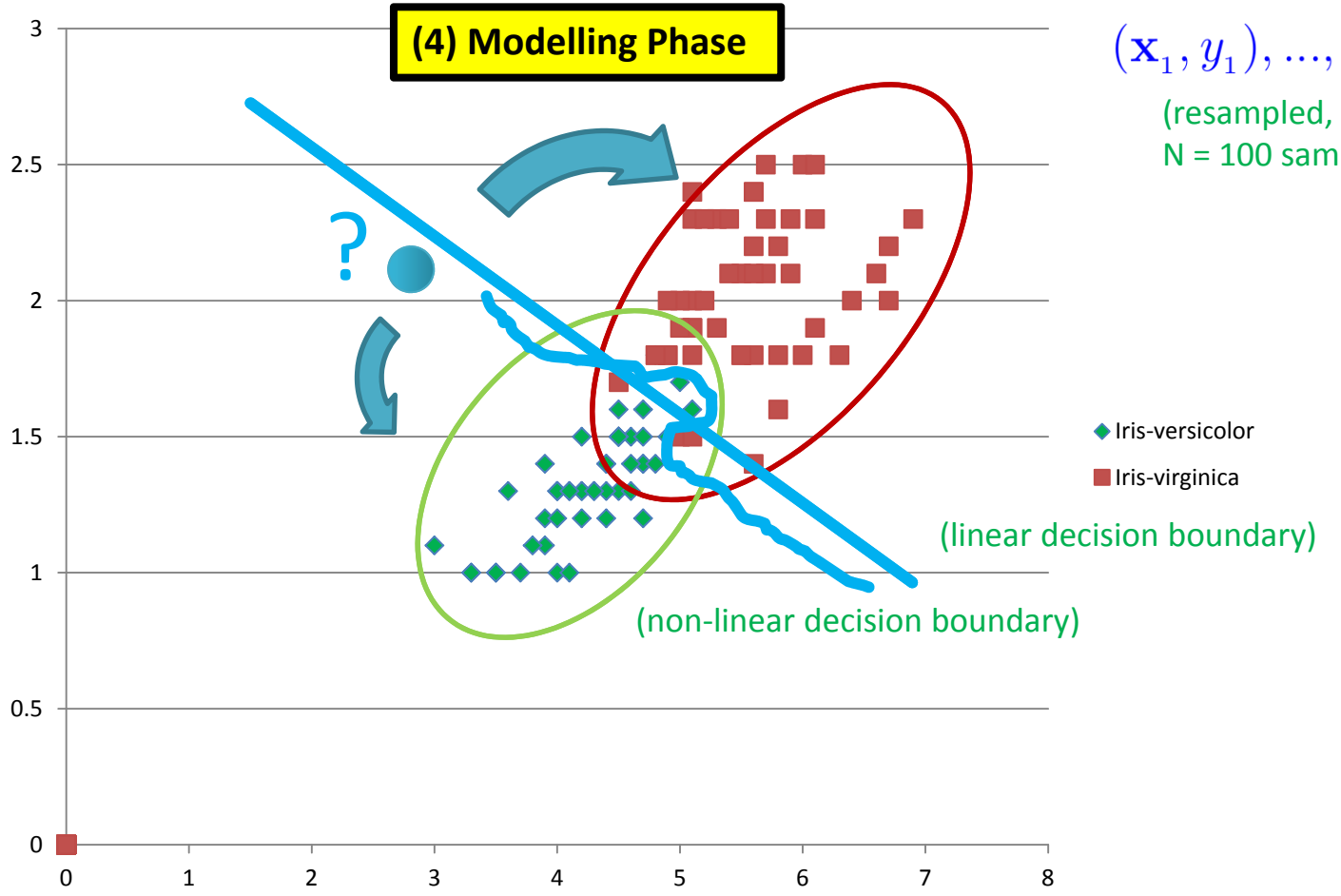
Key Challenges: Why is it not so easy in practice?

- Scalability
  - Gigabytes, Terabytes, and Petabytes datasets that fit not into memory
  - E.g. algorithms become necessary with out-of-core/CPU strategies
- High Dimensionality
  - Datasets with hundreds or thousand attributes become available
  - E.g. bioinformatics with gene expression data with thousand of features
- Heterogenous and Complex Data
  - More complex data objects emerge and unstructured data sets
  - E.g. Earth observation time-series data across the globe
- Data Ownership and Distribution
  - Distributed datasets are common (e.g. security and transfer challenges)

- **Key challenges faced when doing traditional data analysis and machine learning are scalability, high dimensionality of datasets, heterogenous and complex data, data ownership & distribution**

[1] Introduction to Data Mining

Challenge One - Non-linearly Seperable Data in Practice



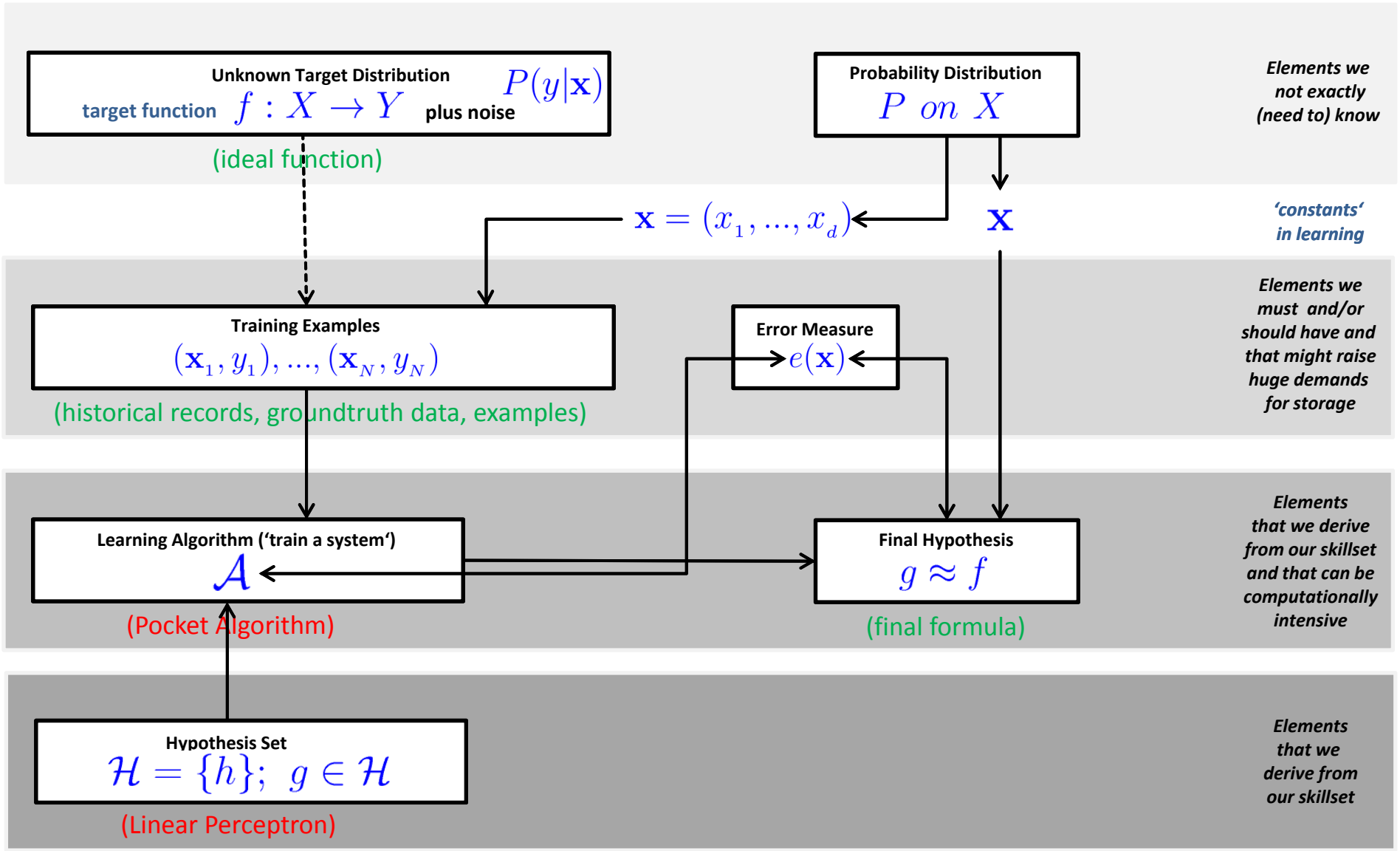
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(resampled, again
N = 100 samples)

(lessons learned from practice: requires soft-thresholds to allow for some errors being overall better for new data)

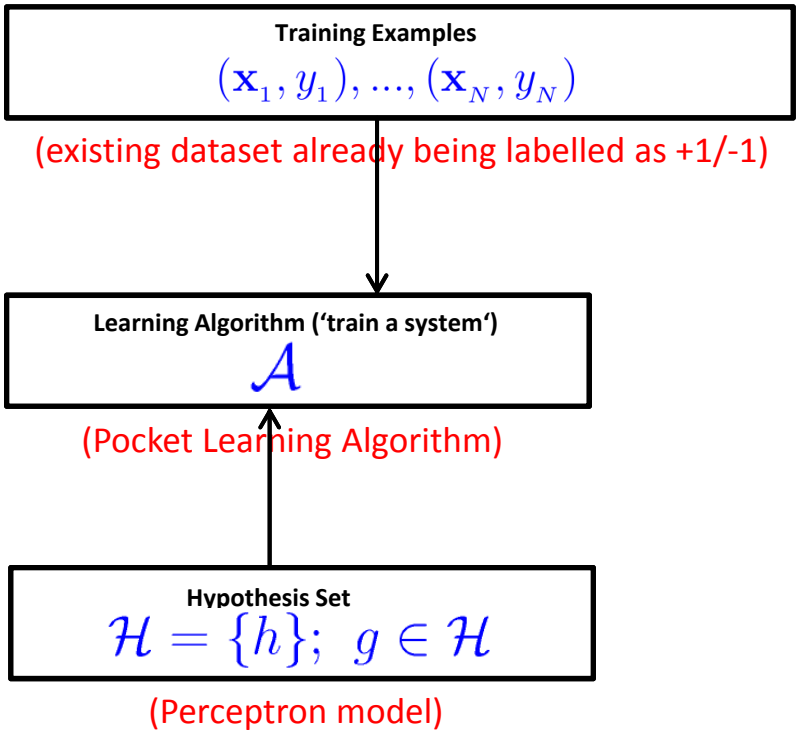
(lessons learned from practice: requires non-linear decision boundaries)

Solution Tools: Linear Perceptron Hypothesis Set & Pocket



Smart Adhancement of PLA – Pocket Algorithm

- When: If we believe there is a linear pattern to be detected
  - No assumption: can be non-linearly seperable data

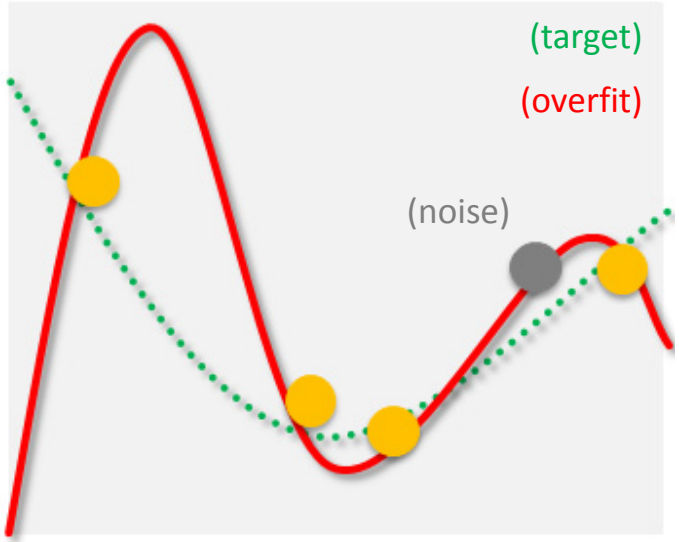


- Basis is still the PLA
- Idea: Put the best solution so far 'in a pocket'
- Best means: Error measure checks per iterations
- Works with non-linearly seperable data
- Needs fixed iterations number (otherwise no convergence of algorithm)

Challenge Two – Problem of Overfitting

- **Overfitting refers to fit the data too well – more than is warranted – thus may misguide the learning**
- **Overfitting is not just 'bad generalization' - e.g. the VC dimension covers noiseless & noise targets**
- **Theory of Regularization are approaches against overfitting and prevent it using different methods**

- Key problem: noise in the target function leads to overfitting
  - Effect: 'noisy target function' and its noise misguides the fit in learning
  - There is always 'some noise' in the data
  - Consequence: poor target function ('distribution') approximation
- Example: Target functions is second order polynomial (i.e. parabola)
  - Using a higher-order polynomial fit
  - Perfect fit: low $E_{in}(g)$, but large $E_{out}(g)$



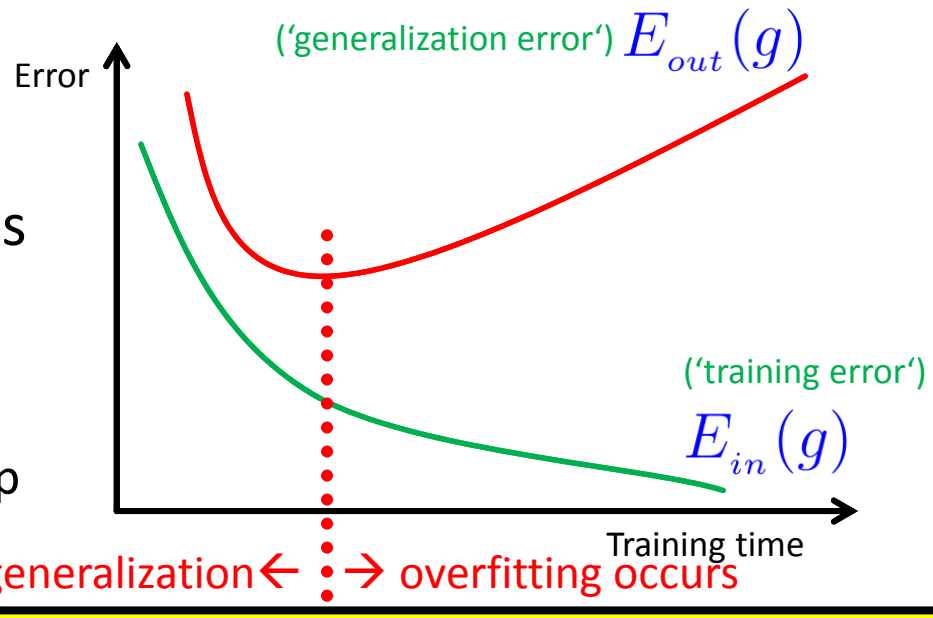
(but simple polynomial works good enough)
('over': here meant as 4th order, a 3rd order would be better, 2nd best)

Problem of Overfitting – Clarifying Terms

- **A good model must have low training error (E_{in}) and low generalization error (E_{out})**
- **Model overfitting is if a model fits the data too well (E_{in}) with a poorer generalization error (E_{out}) than another model with a higher training error (E_{in})**

[1] Introduction to Data Mining

- Overfitting & Errors
  - $E_{in}(g)$ goes down
  - $E_{out}(g)$ goes up
- 'Bad generalization area' ends
  - Good to reduce $E_{in}(g)$
- 'Overfitting area' starts
  - Reducing $E_{in}(g)$ does not help
  - Reason 'fitting the noise'



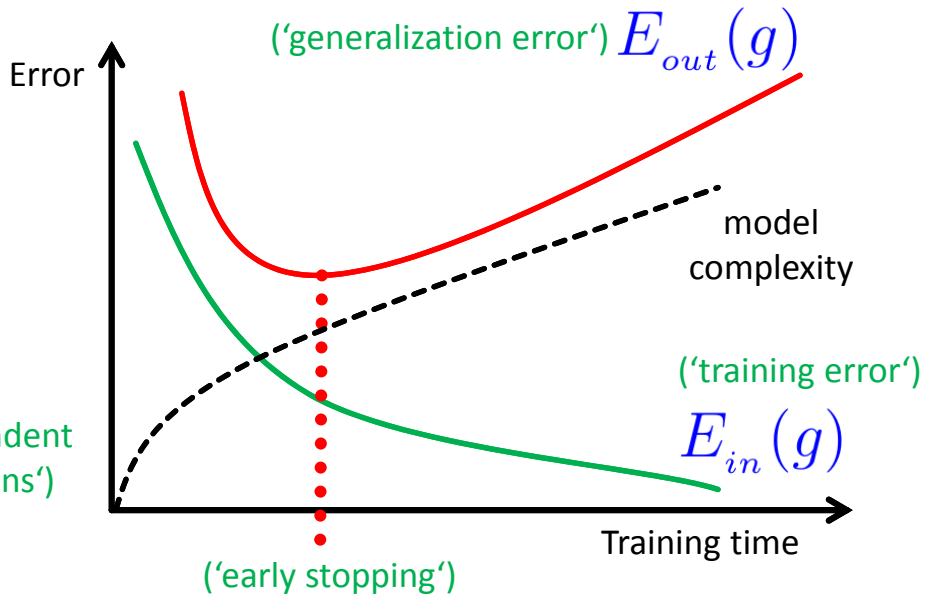
- **The two general approaches to prevent overfitting are (1) regularization and (2) validation**

➢ **Lecture 6 provides details on validation to be considered as another method against overfitting**

Problem of Overfitting – Model Relationships

- Review ‘overfitting situations‘
  - When comparing ‘various models‘ and related to ‘model complexity‘
  - Different models are used, e.g. 2nd and 4th order polynomial
  - Same model is used with e.g. two different instances (e.g. two neural networks but with different parameters)
- Intuitive solution
  - Detect when it happens
  - ‘Early stopping regularization term‘ to stop the training
  - Early stopping method (later)

(‘model complexity measure: the VC analysis was independent of a specific target function – bound for all target functions‘)



- **‘Early stopping‘ approach is part of the theory of regularization, but based on validation methods**

Problem of Overfitting – Noise Term Revisited

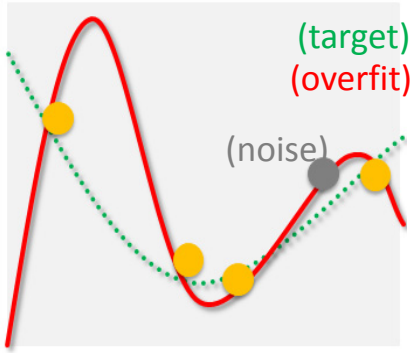
- ‘(Noisy) Target function‘ is not a (deterministic) function
  - Getting with ‘same x in‘ the ‘same y out‘ is not always given in practice
  - Idea: Use a ‘target distribution‘ instead of ‘target function‘

Unknown Target Distribution

target function $f : X \rightarrow Y$ plus noise $P(y|\mathbf{x})$

(ideal function)

- **Fitting some noise in the data is the basic reason for overfitting and harms the learning process**
- **Big datasets tend to have more noise in the data so the overfitting problem might occur even more intense**



- ‘Different types of some noise‘ in data
  - Key to understand overfitting & preventing it
  - ‘Shift of view‘: refinement of noise term
  - Learning from data: ‘matching properties of # data‘

‘shift the view’

(‘function view‘)

(‘# data view‘)

‘# samples‘

Problem of Overfitting – Stochastic Noise

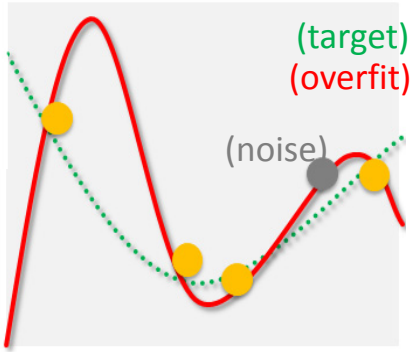
- Stoachastic noise is a part ‘on top of‘ each learnable function
  - Noise in the data that can not be captured and thus not modelled by f
  - Random noise : aka ‘non-deterministic noise‘
  - Conventional understanding established early in this course
  - Finding a ‘non-existing pattern in noise not feasible in learning‘

Unknown Target Distribution

target function $f : X \rightarrow Y$ plus noise $P(y|\mathbf{x})$

(ideal function)

- Practice Example
  - Random fluctuations and/or measurement errors in data (cf. Lecture 1, PANGAEA)
  - Fitting a pattern that not exists ‘out-of-sample‘
  - Puts learning progress ‘off-track‘ and ‘away from f‘

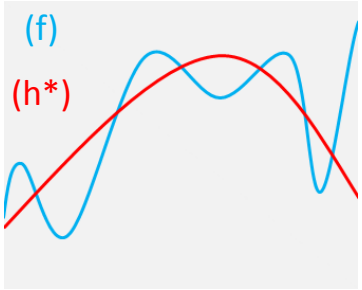


- **Stochastic noise here means noise that can‘t be captured, because it‘s just pure ‘noise as is‘ (nothing to look for) – aka no pattern in the data to understand or to learn from**

Problem of Overfitting – Deterministic Noise

- Part of target function f that H can not capture: $f(\mathbf{x}) - h^*(\mathbf{x})$
  - Hypothesis set H is limited so best h* can not fully approximate f
  - h* approximates f, but fails to pick certain parts of the target f
  - 'Behaves like noise', existing even if data is 'stochastic noiseless'
- Different 'type of noise' than stochastic noise
  - Deterministic noise depends on $\mathcal{H}$ (determines how much more can be captured by h*)
  - E.g. same f, and more sophisticated $\mathcal{H}$: noise is smaller
    (stochastic noise remains the same,
    nothing can capture it)
  - Fixed for a given $\mathbf{x}$, clearly measurable
    (stochastic noise may vary for values of $\mathbf{x}$)

(learning deterministic noise is outside the ability to learn for a given h*)



- **Deterministic noise here means noise that can't be captured, because it is a limited model (out of the league of this particular model), e.g. 'learning with a toddler statistical learning theory'**

Problem of Overfitting – Impacts on Learning

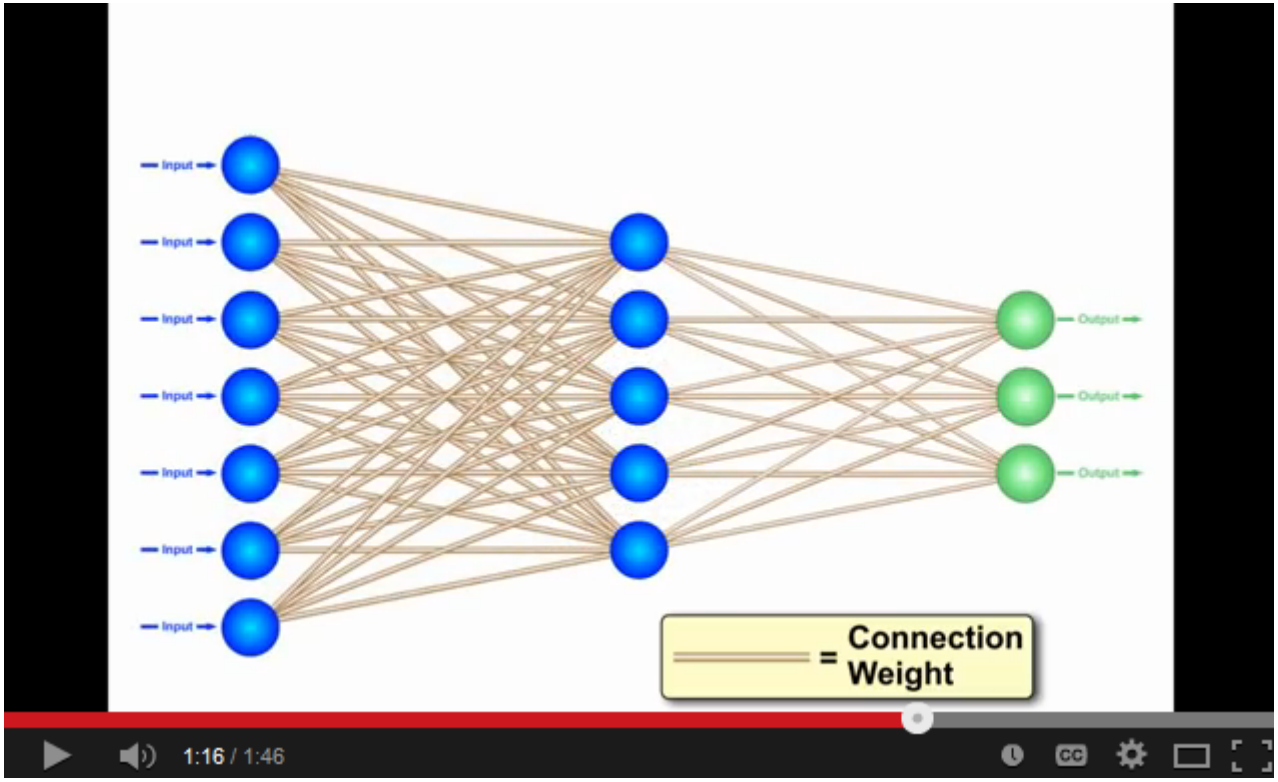
- **The higher the degree of the polynomial (cf. model complexity), the more degrees of freedom are existing and thus the more capacity exists to overfit the training data**

- Understanding deterministic noise & target complexity
  - Increasing target complexity increases deterministic noise (at some level)
  - Increasing the number of data N decreases the deterministic noise
- Finite N case: $\mathcal{H}$ tries to fit the noise
  - Fitting the noise straightforward (e.g. Perceptron Learning Algorithm)
  - Stochastic (in data) and deterministic (simple model) noise will be part of it
- Two 'solution methods' for avoiding overfitting
  - Regularization: 'Putting the brakes in learning', e.g. early stopping (more theoretical, hence 'theory of regularization')
  - Validation: 'Checking the bottom line', e.g. other hints for out-of-sample (more practical, methods on data that provides 'hints')

➢ **Lecture 6 provides details on validation to be considered as another method against overfitting**

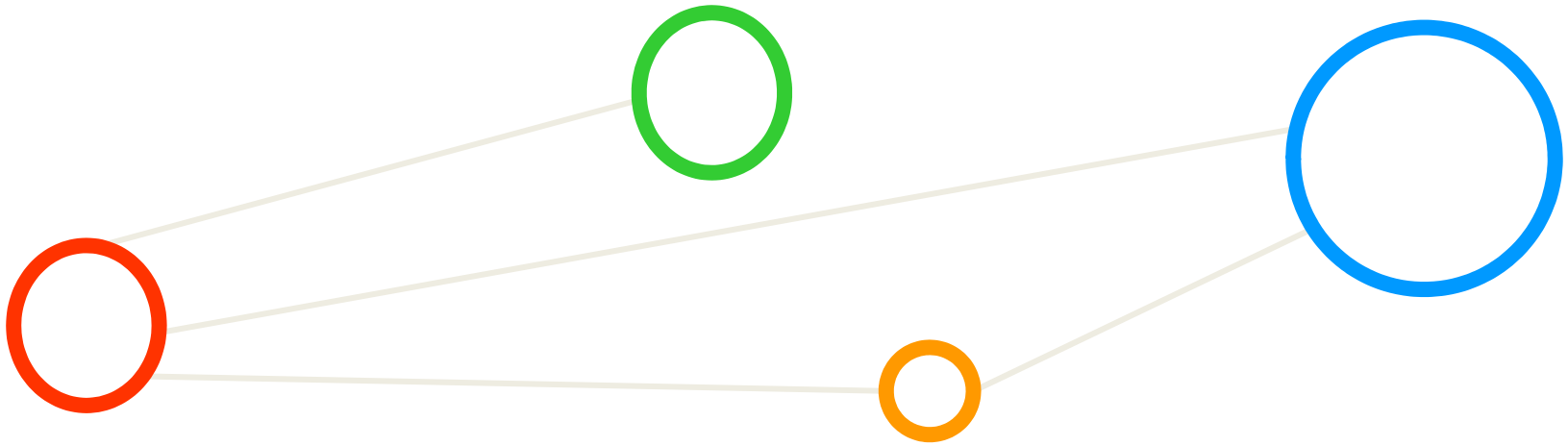
Exercises

[Video] Towards Multi-Layer Perceptrons



[3] YouTube Video, Neural Networks – A Simple Explanation

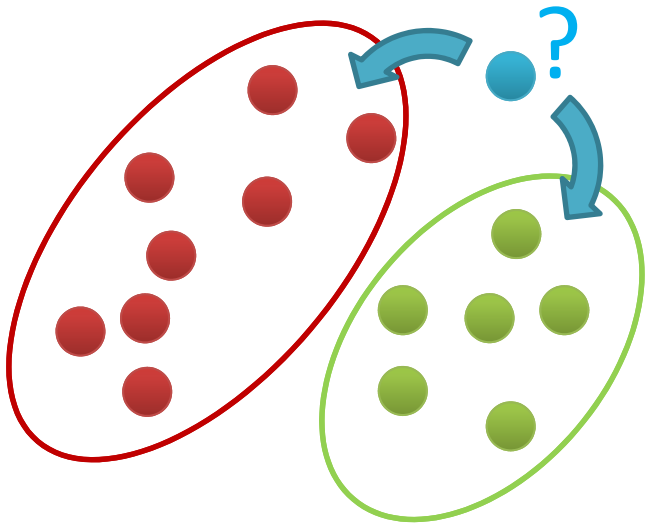
Maximum Margin Classifier



Methods Overview – Focus in this Lecture

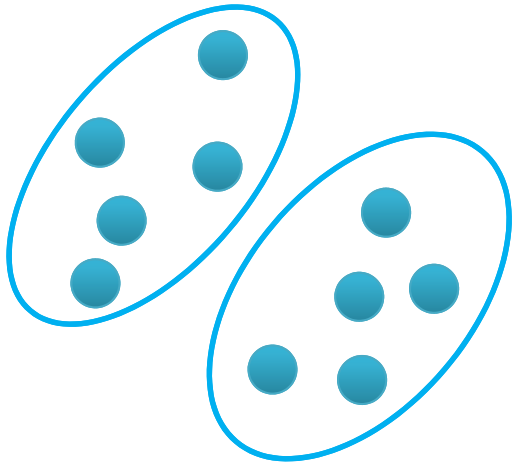
- **Statistical data mining methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



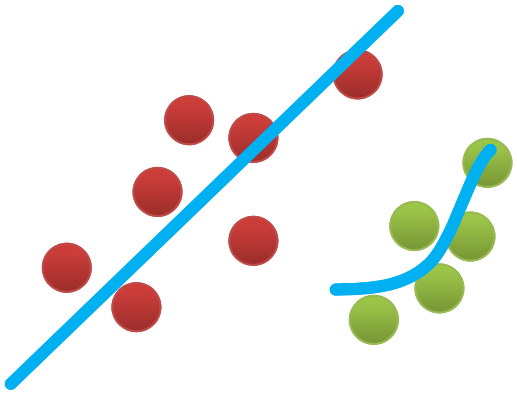
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

Regression



- Identify a line with a certain slope describing the data

Term Support Vector Machines Refined

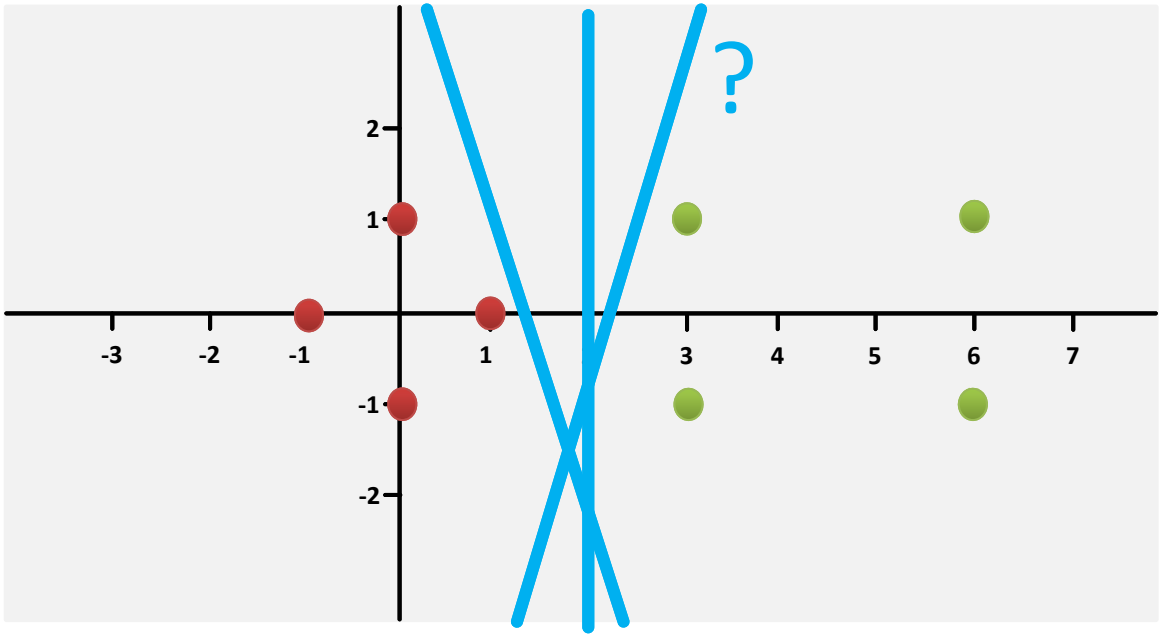
- **Support Vector Machines (SVMs) are a classification technique developed ~1990**
- **SVMs perform well in many settings & are considered as one of the best 'out of the box classifiers'**

[2] An Introduction to Statistical Learning

- Term detailed refinement into 'three separate techniques'
  - Practice: applications mostly use the SVMs with kernel methods
- 'Maximal margin classifier'
  - A simple and intuitive classifier with a 'best' linear class boundary
  - Requires that data is 'linearly separable'
- 'Support Vector Classifier'
  - Extension to the maximal margin classifier for non-linearly seperable data
  - Applied to a broader range of cases, idea of 'allowing some error'
- 'Support Vector Machines' → Using Non-Linear Kernel Methods
  - Extension of the support vector classifier
  - Enables non-linear class boundaries & via kernels;

Expected Out-of-Sample Performance for ‘Any Line‘

- We believe there is a (linear) pattern to be detected
  - Assumption: linearly seperable data (later non-seperable cases)
  - Performance question: What is the optimal line (decision boundary)?
  - E.g. green data: $\left\{\begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}3\\-1\end{pmatrix}, \begin{pmatrix}6\\1\end{pmatrix}, \begin{pmatrix}6\\-1\end{pmatrix}\right\}$ red data: $\left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\-1\end{pmatrix}, \begin{pmatrix}-1\\0\end{pmatrix}\right\}$



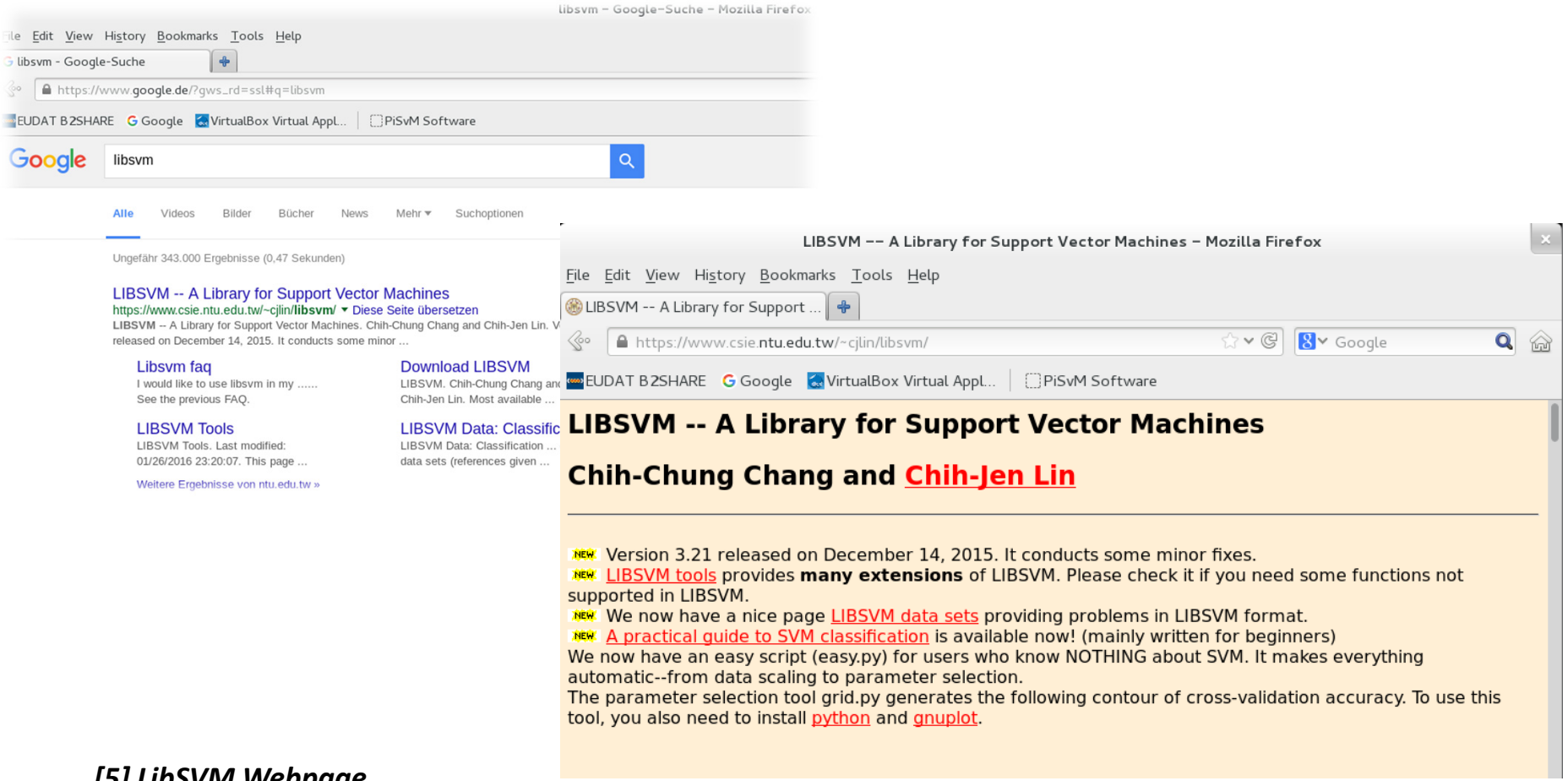
(PLA gives us just any line as soon as all samples are correctly classified)

(How can we craft a margin expressing ‘furthest away‘)

- **Intuition tells us just ‘furthest away’ from the closest points is a good position for the line – why?**

LibSVM – Defacto Standard SVM Implementation

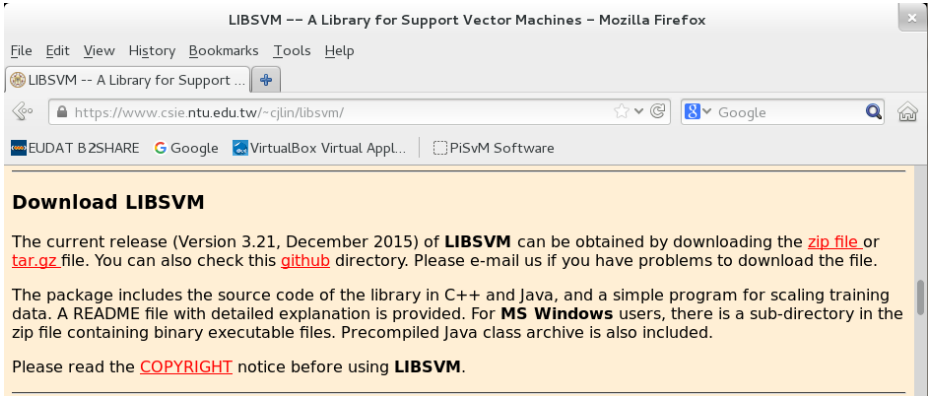
- Free available tool
  - Includes Sequential Minimal Optimization (SMO) implementation



[5] LibSVM Webpage

LibSVM Installation – Download

- Download tar.gz (or in Windows zip bundle)



[5] LibSVM Webpage

- Put package in a folder of your choice
  - Alternatively copy file to your usual working environment

```
adminuser@linux-8djg:~/tools> scp libsvm-3.21.tar.gz mriedel@jureca.fz-juelich.de:/homeb/zam/mriedel
libsvm-3.21.tar.gz                                          100%  827KB 827.4KB/s   00:00
```

```
-bash-4.2$ ls -al
total 64
drwxr-xr-x  2 mriedel zam    512 Jul  6 20:00 .
drwxr-xr-x 29 mriedel zam  32768 Jul  6 19:58 ..
-rw-r--r--  1 mriedel zam 847291 Jul  6 20:00 libsvm-3.21.tar.gz
-bash-4.2$ pwd
/homeb/zam/mriedel/serialtools
```

LibSVM Installation – Unpack the Bundle

- Untar (or Unzip in Windows)

```
/homeb/zam/mriedel/serialtools
-bash-4.2$ tar -zxvf libsvm-3.21.tar.gz
libsvm-3.21/
libsvm-3.21/COPYRIGHT
libsvm-3.21/svm-predict.c
libsvm-3.21/svm.cpp
libsvm-3.21/README
libsvm-3.21/Makefile.win
libsvm-3.21/svm.h
libsvm-3.21/heart_scale
libsvm-3.21/java/
libsvm-3.21/java/svm_toy.java
libsvm-3.21/java/svm_scale.java
libsvm-3.21/java/libsvm/
libsvm-3.21/java/libsvm/svm_model.java
libsvm-3.21/java/libsvm/svm.m4
libsvm-3.21/java/libsvm/svm_problem.java
libsvm-3.21/java/libsvm/svm.java
libsvm-3.21/java/libsvm/svm_node.java
libsvm-3.21/java/libsvm/svm_parameter.java
libsvm-3.21/java/libsvm/svm_print_interface.java
libsvm-3.21/java/svm_train.java
libsvm-3.21/java/Makefile
libsvm-3.21/java/test_applet.html
libsvm-3.21/java/libsvm.jar
libsvm-3.21/java/svm_predict.java
libsvm-3.21/Makefile
libsvm-3.21/windows/
libsvm-3.21/windows/svm-toy.exe
libsvm-3.21/windows/svm-scale.exe
libsvm-3.21/windows/svmtrain.mexw64
libsvm-3.21/windows/libsvmwrite.mexw64
libsvm-3.21/windows/libsvm.dll
```

```
/homeb/zam/mriedel/serialtools/libsvm-3.21
-bash-4.2$ ls -al
total 544
drwxr-xr-x 8 mriedel zam 32768 Dec 14  2015 .
drwxr-xr-x 3 mriedel zam   512 Jul  6 20:03 ..
-rw-r--r-- 1 mriedel zam  1497 Dec 14  2015 COPYRIGHT
-rw-r--r-- 1 mriedel zam 83089 Dec 14  2015 FAQ.html
-rw-r--r-- 1 mriedel zam 27670 Dec 14  2015 heart_scale
drwxr-xr-x 3 mriedel zam   512 Dec 14  2015 java
-rw-r--r-- 1 mriedel zam   732 Dec 14  2015 Makefile
-rw-r--r-- 1 mriedel zam  1136 Dec 14  2015 Makefile.win
drwxr-xr-x 2 mriedel zam   512 Dec 14  2015 matlab
drwxr-xr-x 2 mriedel zam   512 Dec 14  2015 python
-rw-r--r-- 1 mriedel zam 28679 Dec 14  2015 README
-rw-r--r-- 1 mriedel zam 64836 Dec 14  2015 svm.cpp
-rw-r--r-- 1 mriedel zam   477 Dec 14  2015 svm.def
-rw-r--r-- 1 mriedel zam  3382 Dec 14  2015 svm.h
-rw-r--r-- 1 mriedel zam  5536 Dec 14  2015 svm-predict.c
-rw-r--r-- 1 mriedel zam  8539 Dec 14  2015 svm-scale.c
drwxr-xr-x 5 mriedel zam   512 Dec 14  2015 svm-toy
-rw-r--r-- 1 mriedel zam  8986 Dec 14  2015 svm-train.c
drwxr-xr-x 2 mriedel zam   512 Dec 14  2015 tools
drwxr-xr-x 2 mriedel zam   512 Dec 14  2015 windows
```

LibSVM Installation – Make (only in UNIX)

- Use make to generate executables (needs g++ compiler)

```
-bash-4.2$ pwd
/homeb/zam/mriedel/serialtools/libsvm-3.21
-bash-4.2$ make
g++ -Wall -Wconversion -O3 -fPIC -c svm.cpp
g++ -Wall -Wconversion -O3 -fPIC svm-train.c svm.o -o svm-train -lm
g++ -Wall -Wconversion -O3 -fPIC svm-predict.c svm.o -o svm-predict -lm
g++ -Wall -Wconversion -O3 -fPIC svm-scale.c -o svm-scale
```

- Check executables important for us

```
-bash-4.2$ pwd
/homeb/zam/mriedel/serialtools/libsvm-3.21
-bash-4.2$ ls -al
total 896
drwxr-xr-x 8 mriedel zam  32768 Jul  6 20:05 .
drwxr-xr-x 3 mriedel zam    512 Jul  6 20:03 ..
-rw-r--r-- 1 mriedel zam   1497 Dec 14  2015 COPYRIGHT
-rw-r--r-- 1 mriedel zam  83089 Dec 14  2015 FAQ.html
-rw-r--r-- 1 mriedel zam  27670 Dec 14  2015 heart_scale
drwxr-xr-x 3 mriedel zam    512 Dec 14  2015 java
-rw-r--r-- 1 mriedel zam    732 Dec 14  2015 Makefile
-rw-r--r-- 1 mriedel zam   1136 Dec 14  2015 Makefile.win
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 matlab
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 python
-rw-r--r-- 1 mriedel zam  28679 Dec 14  2015 README
-rw-r--r-- 1 mriedel zam  64836 Dec 14  2015 svm.cpp
-rw-r--r-- 1 mriedel zam    477 Dec 14  2015 svm.def
-rw-r--r-- 1 mriedel zam   3382 Dec 14  2015 svm.h
-rwxr-xr-x 1 mriedel zam  78270 Jul  6 20:05 svm-predict
-rwxr-xr-x 1 mriedel zam  18587 Jul  6 20:05 svm-scale
-rw-r--r-- 1 mriedel zam   8539 Dec 14  2015 svm-scale.c
-rwxr-xr-x 1 mriedel zam  78509 Jul  6 20:05 svm-train
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 tools
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 windows
```

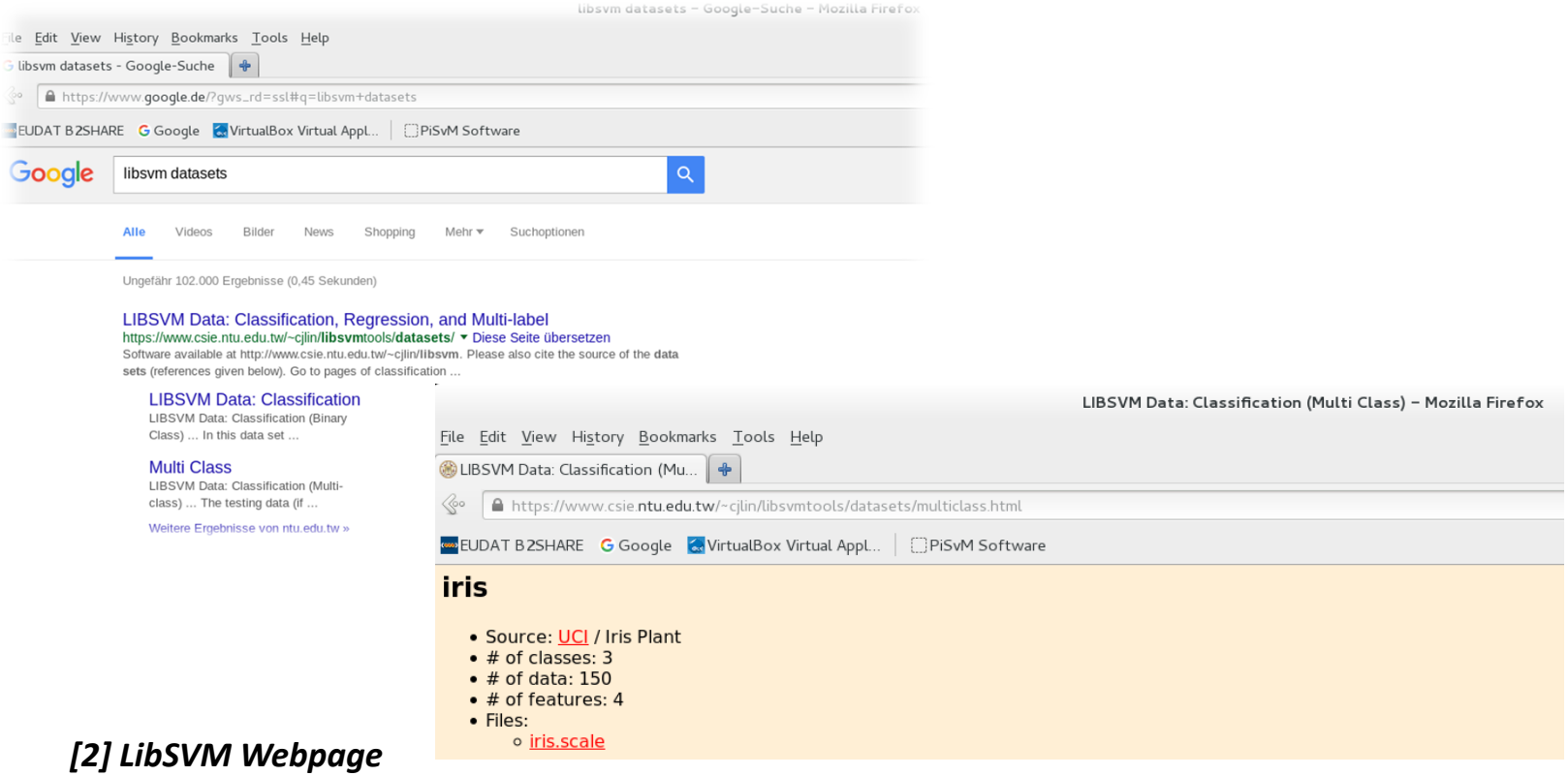
(use in testing phase)

(use in training phase)

[5] LibSVM Webpage

Scaling

- Scaled version of our data (cf. Lecture 1 & 3): iris.scale
  - Scaling is used in order that the optimization does not have to work with large numbers – so one can scale, but it is not a requirement
  - Sometimes the performance improved with scaling



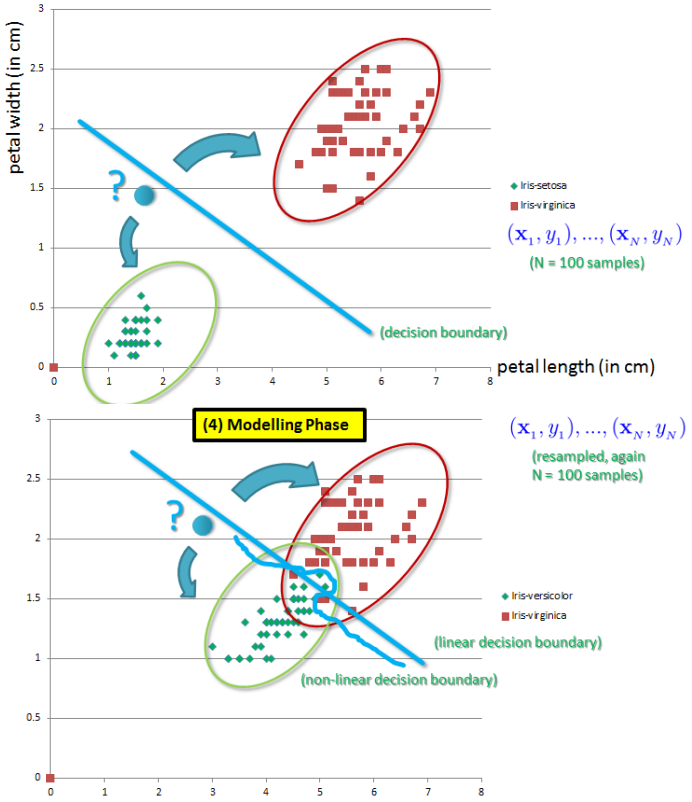
[2] LibSVM Webpage

Data Preparation Phase

- Copy IRIS Dataset in your working environment

```
adminuser@linux-8djg:~/data> scp iris.scale mriedel@jureca.fz-juelich.de:/homeb/zam/mriedel
iris.scale                                                    100% 6954     6.8KB/s   00:00

/homeb/zam/mriedel/datasets
-bash-4.2$ ls -al
total 64
drwxr-xr-x  2 mriedel zam   512 Jul  6 21:53 .
drwxr-xr-x 30 mriedel zam 32768 Jul  6 21:53 ..
-rw-r--r--  1 mriedel zam  6954 Jul  6 21:53 iris.scale
```

- Dataset Two-class problem, linearly seperable
  - Dataset Iris Setosa (class 1) and Iris Virginica (class 3)
  - Iris-class1and3-training(20)/testing(30)
- Dataset Two-class problem, not linearly seperable
  - Dataset Iris Veriscolor (class 2) and Iris Virginica (class3)
  - iris-class2and3-training(20)/testing(30)



Exercises

Iris Dataset

- Iris dataset is already available in the tuturial directory

Iris Dataset LibSVM Format Preprocessing

Morris Riedel

;

03 July 2016

http://b2share.eudat.eu

Abstract: UCI Machine Learning Repository
IRIS Dataset
iris.scale.original and iris.scale
- 3 classes, 50 samples each class
iris-class1and3
- only linearly seperable data
- class 1 and 3 sampling
- 100 samples
iris-class1and3-training/testing
- 20 for training, 30 for testing
- per class 1 and 3
iris-class2and3-training/testing
- 20 for training, 30 for testing
- per class 2 and 3

Keyword(s): LibSVM ; Iris ; Flowers ; UCI

The record appears in these collections:
Generic

(persistent handle link for publication into papers)

B2SHARE
Store and Share Research Data

[6] Iris Dataset LibSVM Format Preprocessing

```
/apps/gent/tutorials/machine_learning/classification/Iris
[vsc42544@gligar02 Iris]$ ls -al
total 256
drwxr-xr-x 2 vsc40003 vsc40003 4096 Nov 22 15:42 .
drwxr-xr-x 6 vsc40003 vsc40003 4096 Nov 22 15:44 ..
-rw-r--r-- 1 vsc40003 vsc40003 2736 Nov  9 21:41 iris-class1and3-testing.txt
-rw-r--r-- 1 vsc40003 vsc40003 1806 Nov  9 21:41 iris-class1and3-training.txt
-rw-r--r-- 1 vsc40003 vsc40003 4542 Nov  9 21:41 iris-class1and3.txt
-rw-r--r-- 1 vsc40003 vsc40003 2841 Nov  9 21:41 iris-class2and3-testing.txt
-rw-r--r-- 1 vsc40003 vsc40003 3184 Nov  9 21:41 iris-class2and3-training.txt
-rw-r--r-- 1 vsc40003 vsc40003 4658 Nov  9 21:42 iris-class2and3.txt
-rw-r--r-- 1 vsc40003 vsc40003 6954 Nov  9 21:42 iris.scale.original.original
-rw-r--r-- 1 vsc40003 vsc40003 6954 Nov  9 21:42 iris.scale.scale
```

Training Phase: linearly seperable case (iris-class1and3)

- Use svm-train (c<=0 not allowed)

```
-bash-4.2$ more svm-train1-3.sh
./svm-train -t 0 -c 1 /homeb/zam/mriedel/datasets/iris-class1and3-training

-bash-4.2$ ./svm-train1-3.sh
*
optimization finished, #iter = 11
nu = 0.035490
obj = -0.709742, rho = 0.447384
nSV = 4, nBSV = 0
Total nSV = 4
```

- Check model file

```
-bash-4.2$ more iris-class1and3-training.model
svm_type c_svc
kernel_type linear
nr_class 2
total_sv 4
rho 0.447384
label 1 3
nr_sv 3 1
SV
0.1374686716356165 1:-0.666667 2:-0.166667 3:-0.864407 4:-0.916667
0.1011680329343598 1:-0.388889 2:0.583333 3:-0.762712 4:-0.75
0.4711540941141194 1:-0.944444 2:-0.25 3:-0.864407 4:-0.916667
-0.7097907986840956 1:-0.666667 2:-0.583333 3:0.186441 4:0.333333
```

```
-bash-4.2$ ls -al
total 896
drwxr-xr-x 8 mriedel zam  32768 Jul  6 22:25 .
drwxr-xr-x 3 mriedel zam    512 Jul  6 20:03 ..
-rw-r--r-- 1 mriedel zam   1497 Dec 14  2015 COPYRIGHT
-rw-r--r-- 1 mriedel zam  83089 Dec 14  2015 FAQ.html
-rw-r--r-- 1 mriedel zam  27670 Dec 14  2015 heart_scale
-rw-r--r-- 1 mriedel zam    354 Jul  6 22:25 iris-class1and3-training.model
drwxr-xr-x 3 mriedel zam    512 Dec 14  2015 java
-rw-r--r-- 1 mriedel zam    732 Dec 14  2015 Makefile
-rw-r--r-- 1 mriedel zam   1136 Dec 14  2015 Makefile.win
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 matlab
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 python
-rw-r--r-- 1 mriedel zam  28679 Dec 14  2015 README
-rw-r--r-- 1 mriedel zam  64836 Dec 14  2015 svm.cpp
-rw-r--r-- 1 mriedel zam    477 Dec 14  2015 svm.def
-rw-r--r-- 1 mriedel zam   3382 Dec 14  2015 svm.h
-rw-r--r-- 1 mriedel zam 100224 Jul  6 20:05 svm.o
-rwxr-xr-x 1 mriedel zam  78270 Jul  6 20:05 svm-predict
-rw-r--r-- 1 mriedel zam   5536 Dec 14  2015 svm-predict.c
-rwxr-xr-x 1 mriedel zam  18587 Jul  6 20:05 svm-scale
-rw-r--r-- 1 mriedel zam   8539 Dec 14  2015 svm-scale.c
drwxr-xr-x 5 mriedel zam    512 Dec 14  2015 svm-toy
-rwxr-xr-x 1 mriedel zam  78509 Jul  6 20:05 svm-train
-rwxr-xr-x 1 mriedel zam     76 Jul  6 22:24 svm-train1-3.sh
-rw-r--r-- 1 mriedel zam   8986 Dec 14  2015 svm-train.c
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 tools
drwxr-xr-x 2 mriedel zam    512 Dec 14  2015 windows
```

Testing Phase: linearly seperable case (iris-class1and3)

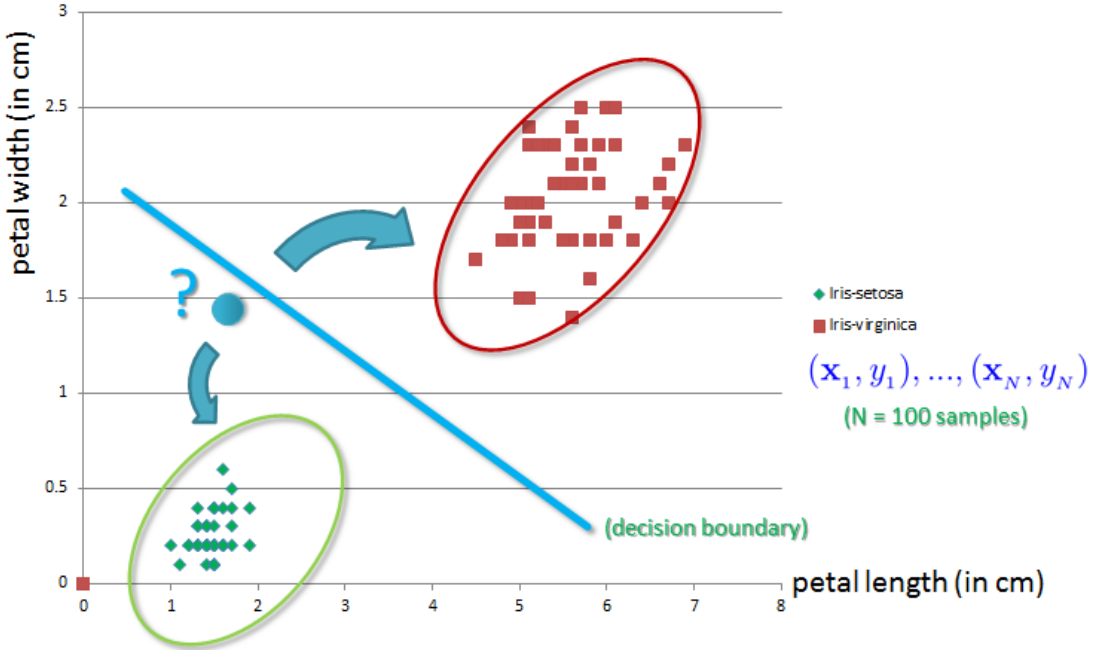
- Use svm-predict (using newly created model file & testing data)

```
-bash-4.2$ more svm-predict1-3.sh
./svm-predict /homeb/zam/mriedel/datasets/iris-class1and3-testing ./iris-class1and3-training.model ./results.txt
```

```
-bash-4.2$ ./svm-predict1-3.sh
Accuracy = 100% (60/60) (classification)
```

```
-bash-4.2$ head results.txt
1
1
1
1
1
1
1
1
1
1
```

(consistent with our graph: 100% here possible since very easy problem, in practice rarely)



LibSVM – svm-train Parameters

- Important parameters (training phase)

```
-bash-4.2$ ./svm-train
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
        0 -- C-SVC              (multi-class classification)
        1 -- nu-SVC             (multi-class classification)
        2 -- one-class SVM
        3 -- epsilon-SVR        (regression)
        4 -- nu-SVR             (regression)
-t kernel_type : set type of kernel function (default 2)
        0 -- linear: u'*v
        1 -- polynomial: (gamma*u'*v + coef0)^degree
        2 -- radial basis function: exp(-gamma*|u-v|^2)
        3 -- sigmoid: tanh(gamma*u'*v + coef0)
        4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)
```

(we need a training set file)

(take default here = C-SVC)

(in this lecture we have just ‘linear kernels‘)

(Regularization Parameter)

Training Examples

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

[5] LibSVM Webpage

LibSVM – svm-predict Parameters

- Important parameters (testing phase)

```
-bash-4.2$ ./svm-predict
Usage: svm-predict [options] test_file model_file output_file
options:
-b probability_estimates: whether to predict probability estimates, 0 or 1 (default 0); for one-class SVM only 0 is supported
-q : quiet mode (no outputs)
```

(the model file is generated in the training phase → the support vectors found in optimization)

(test file is a testing dataset set aside to be used once training is finished)

(output file gives us indications how each sample was classified)

Testing Examples

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

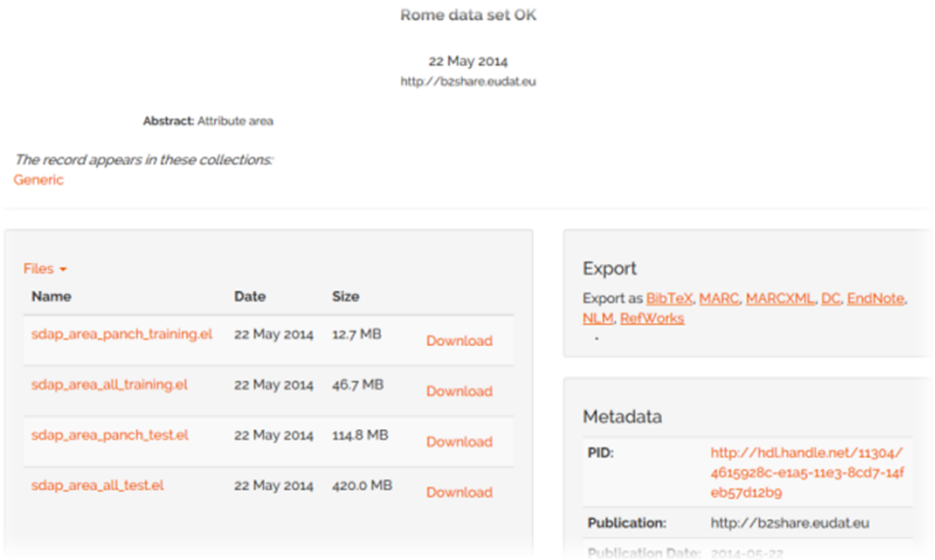
Interactive Session on Golett for Serial LibSVM

- Use "qsub -I -W x=FLAGS:ADVRES:<reservationID>"
  - (the -I option here is 'minus capital i')
  - Running 'module swap cluster/golett‘ first
  - qsub -I -W x=FLAGS:ADVRES:machine_learning.317 (Friday)
  - Note that job scripts and interactive sessions will by default be allocated 1 hour of walltime and 1 single processor core
  - Module load LIBSVM/3.22-intel-2016b is our serial SVM implementation

```
[vsc42544@gligar02 ~]$ module swap cluster/golett

The following have been reloaded with a version change:
  1) cluster/delcatty => cluster/golett

[vsc42544@gligar02 ~]$ qsub -I -W x=FLAGS:ADVRES:machine_learning.317
qsub: waiting for job 1174756.master19.golett.gent.vsc to start

[vsc42544@gligar02 ~]$ qsub -I
qsub: waiting for job 1174757.master19.golett.gent.vsc to start
qsub: job 1174757.master19.golett.gent.vsc ready

[vsc42544@node2400 ~]$ module load LIBSVM/3.22-intel-2016b
```

Rome Remote Sensing Dataset (cf. Lecture 3)

- Data is already available in the tutorial directory



(persistent handle link for publication into papers)

B2SHARE
Store and Share Research Data

[8] Rome Image dataset

```
[vsc42544@gligar02 Rome]$ pwd
/apps/gent/tutorials/machine_learning/classification/Rome
[vsc42544@gligar02 Rome]$ ls -al
total 1160512
drwxr-xr-x 2 vsc40003 vsc40003      4096 Nov 22 15:43 .
drwxr-xr-x 6 vsc40003 vsc40003      4096 Nov 22 15:44 ..
-rw-r--r-- 1 vsc40003 vsc40003 419974873 Nov 22 15:39 sdap_area_all_test.el
-rw-r--r-- 1 vsc40003 vsc40003  46652874 Nov 22 15:40 sdap_area_all_training.el
-rw-r--r-- 1 vsc40003 vsc40003 114763982 Nov 22 15:42 sdap_area_panch_test.el
-rw-r--r-- 1 vsc40003 vsc40003  12745692 Nov 22 15:42 sdap_area_panch_training.el
```

Indian Pines Remote Sensing Dataset (cf. Lecture 3)

- *Indian Pines Dataset Raw and Processed*

(persistent handle link for publication into papers)

Abstract: 1) Indian raw: 1417x614x200 (training 10% and test)
2) Indian processed:1417x614x30 (training 10% and test)

B2SHARE
Store and Share Research Data

[9] Indian Pine Image dataset

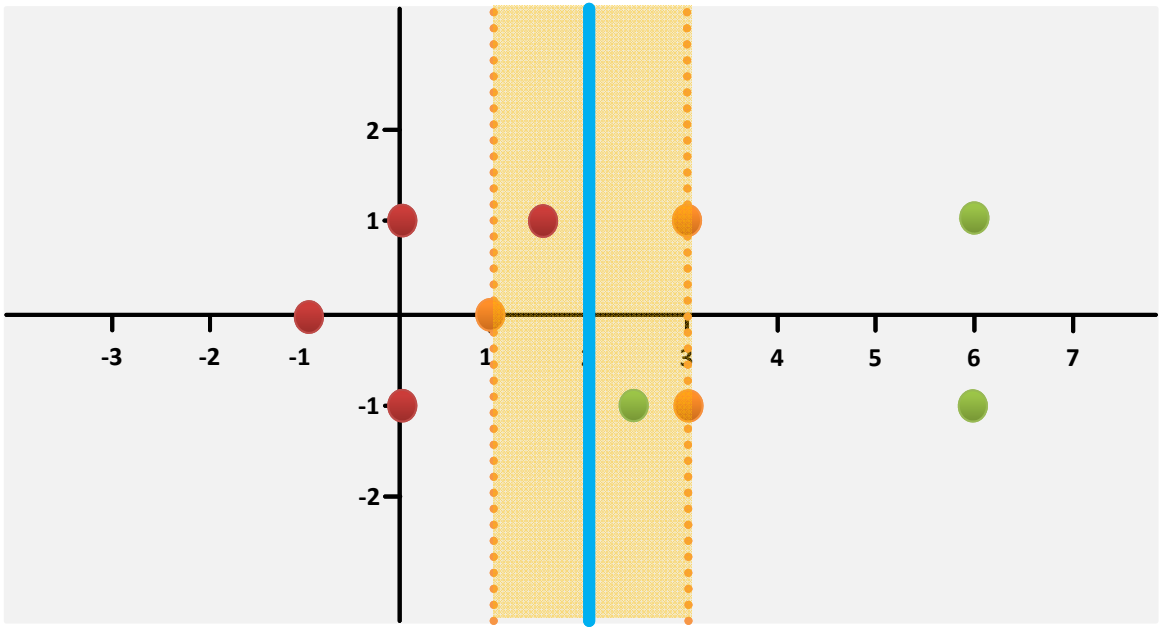
Files

Name	Date	Size	
indian_processed_training.el	05 Feb 2015	11.7 MB	Download
indian_raw_test.el	05 Feb 2015	747.1 MB	Download
indian_raw_training.el	05 Feb 2015	83.0 MB	Download
indian_processed_test.el	05 Feb 2015	105.6 MB	Download

```
[vsc42544@gligar02 Indian]$ pwd
/apps/gent/tutorials/machine_learning/classification/Indian
[vsc42544@gligar02 Indian]$ ls -al
total 1850688
drwxr-xr-x 2 vsc40003 vsc40003      4096 Nov 22 15:42 .
drwxr-xr-x 6 vsc40003 vsc40003      4096 Nov 22 15:44 ..
-rw-r--r-- 1 vsc40003 vsc40003 105594346 Nov  9 21:44 indian_processed_test.el
-rw-r--r-- 1 vsc40003 vsc40003  11732509 Nov  9 21:44 indian_processed_training.el
-rw-r--r-- 1 vsc40003 vsc40003 747125597 Nov  9 21:46 indian_raw_test.el
-rw-r--r-- 1 vsc40003 vsc40003  83014311 Nov  9 21:47 indian_raw_training.el
```

Expected Out-of-Sample Performance for ‘Best Line‘

- The line with a ‘bigger margin‘ seems to be better – but why?
  - Intuition: chance is higher that a new point will still be correctly classified
  - Fewer hypothesis possible: constrained by sized margin (cf. Lecture 3)
  - Idea: achieving good ‘out-of-sample‘ performance is goal (cf. Lecture 3)



(e.g. better performance compared to PLA technique)

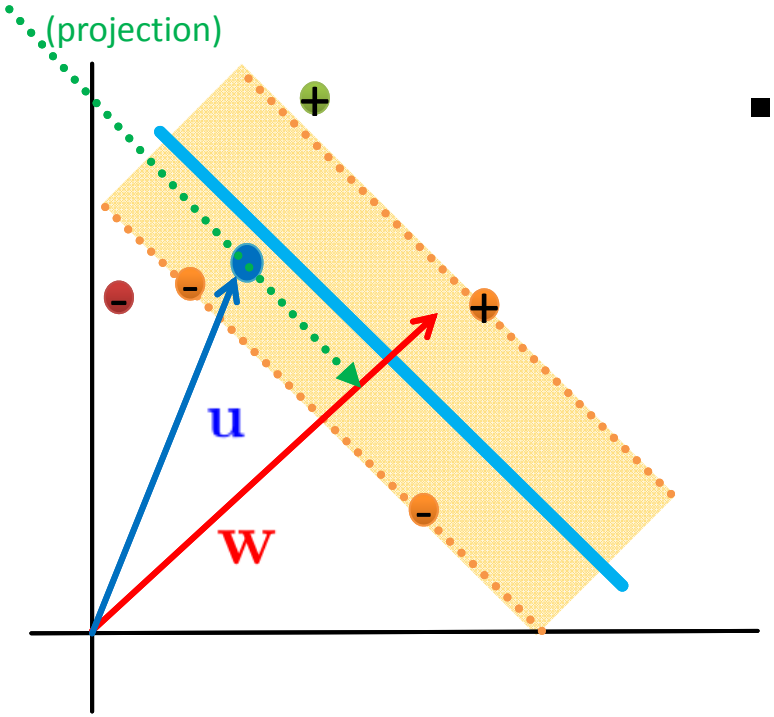
(simple line in a linear setup as intuitive decision boundary)

(Question remains: how we can achieve a bigger margin)

- **Support Vector Machines (SVMs) use maximum margins that will be mathematically established**

Geometric SVM Interpretation and Setup (1)

- Think ‘simplified coordinate system‘ and use ‘Linear Algebra‘
  - Many other samples are removed (red and green not SVs)
  - Vector $\mathbf{w}$ of ‘any length‘ perpendicular to the decision boundary
  - Vector $\mathbf{u}$ points to an unknown quantity (e.g. new sample to classify)
  - Is $\mathbf{u}$ on the left or right side of the decision boundary?



- Dot product $\mathbf{w} \cdot \mathbf{u} \geq C; C = -b$
  - With $\mathbf{u}$ takes the projection on the $\mathbf{w}$
  - Depending on where projection is it is left or right from the decision boundary
  - Simple transformation brings decison rule:

(1) $\mathbf{w} \cdot \mathbf{u} + b \geq 0$ → means +

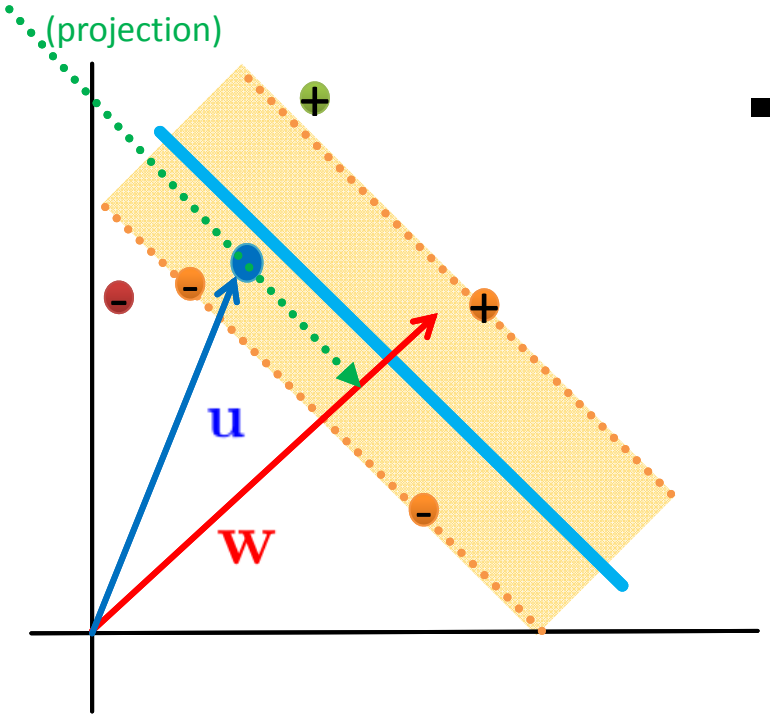
  - (given that b and $\mathbf{w}$ are unknown to us)

(constraints are not enough to fix particular b or w, need more constraints to calculate b or w)

Geometric SVM Interpretation and Setup (2)

- Creating our constraints to get b or $\mathbf{w}$ computed
  - First constraint set for positive samples ⊕ $\mathbf{w} \cdot \mathbf{x}_+ + b \geq 1$
  - Second constraint set for negative samples ⊖ $\mathbf{w} \cdot \mathbf{x}_- + b \leq 1$
  - For mathematical convenience introduce variables (i.e. labelled samples)

$y_i = +$ for ⊕ and $y_i = -$ for ⊖



- Multiply equations by y_i
  - Positive samples: $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$
  - Negative samples: $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$
  - Both same due to $y_i = +$ and $y_i = -$

(brings us mathematical convenience often quoted)

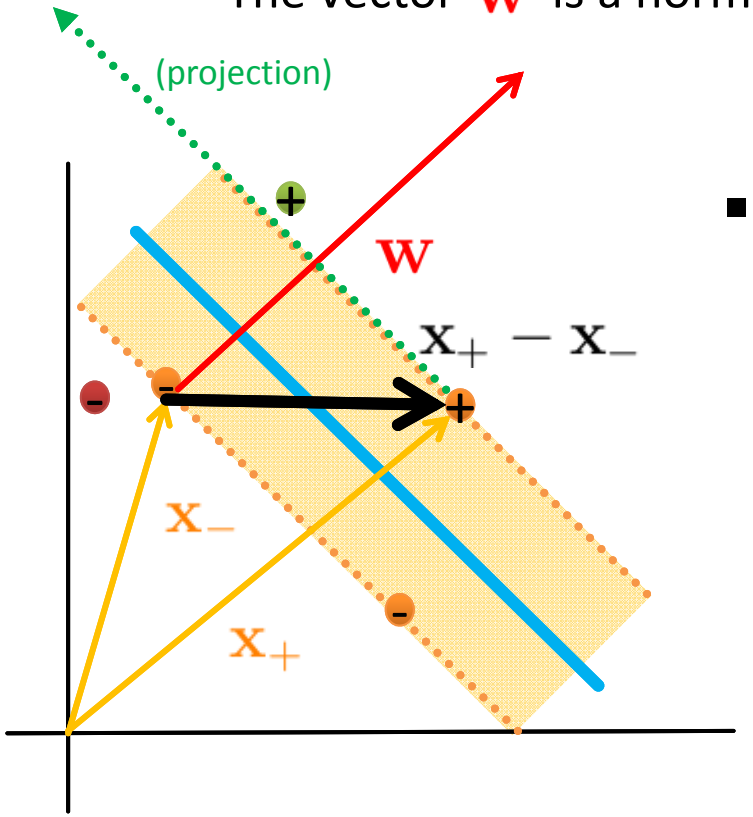
$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0$$

(additional constraints just for support vectors itself helps)

(2) $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$

Geometric SVM Interpretation and Setup (3)

- Determine the 'width of the margin'
  - Difference between positive and negative SVs: $\mathbf{x}_+ - \mathbf{x}_-$
  - Projection of $\mathbf{x}_+ - \mathbf{x}_-$ onto the vector $\mathbf{w}$
  - The vector $\mathbf{w}$ is a normal vector, magnitude is $\|\mathbf{w}\|$



(Dot product of two vectors is a scalar, here the width of the margin)

- Unit vector is helpful for 'margin width'
  - Projection (dot product) for margin width:

$$\mathbf{x}_+ - \mathbf{x}_- \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$ (unit vector)

$1 - b \quad 1 + b$

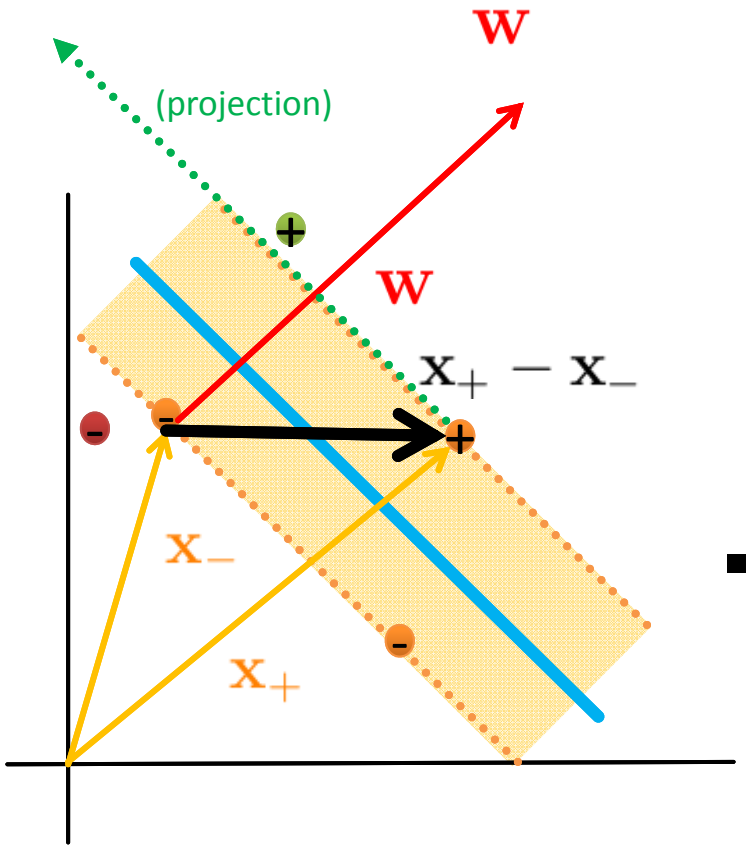
$$\frac{2}{\|\mathbf{w}\|}$$ (3)

  - When enforce constraint: $y_i = +$ $y_i = -$

(2) $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$

Constrained Optimization Steps SVM (1)

- Use ‘constraint optimization‘ of mathematical toolkit
  - Idea is to ‘maximize the width‘ of the margin: $max \frac{2}{\|\mathbf{w}\|}$ (drop the constant 2 is possible here)



$\Rightarrow max \frac{1}{\|\mathbf{w}\|}$ (equivalent)

$\Rightarrow min \|\mathbf{w}\|$ (equivalent for max)

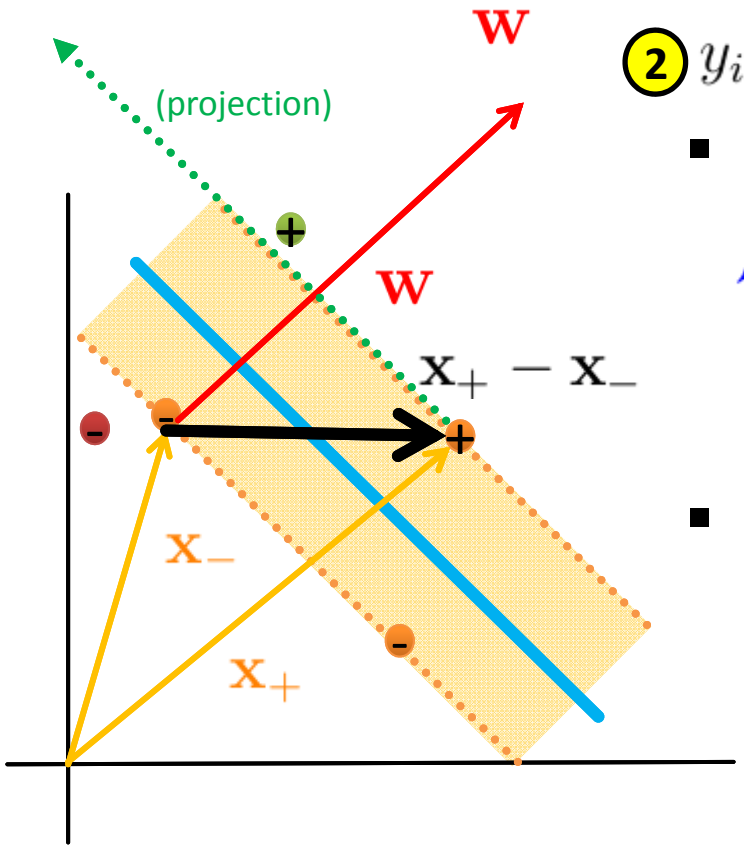
$\Rightarrow min \frac{1}{2}\|\mathbf{w}\|^2$ (mathematical convenience) ③

- Next: Find the extreme values
  - Subject to constraints

② $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$

Constrained Optimization Steps SVM (2)

- Use ‘Lagrange Multipliers‘ of mathematical toolkit
  - Established tool in ‘constrained optimization‘ to find function extremum
  - ‘Get rid‘ of constraints by using Lagrange Multipliers (4)



(2) $y_i(\mathbf{x}_i \cdot \mathbf{w} + b - 1) = 0$

- Introduce a multiplier for each constraint

$$\mathcal{L}(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum \alpha_i [y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1]$$

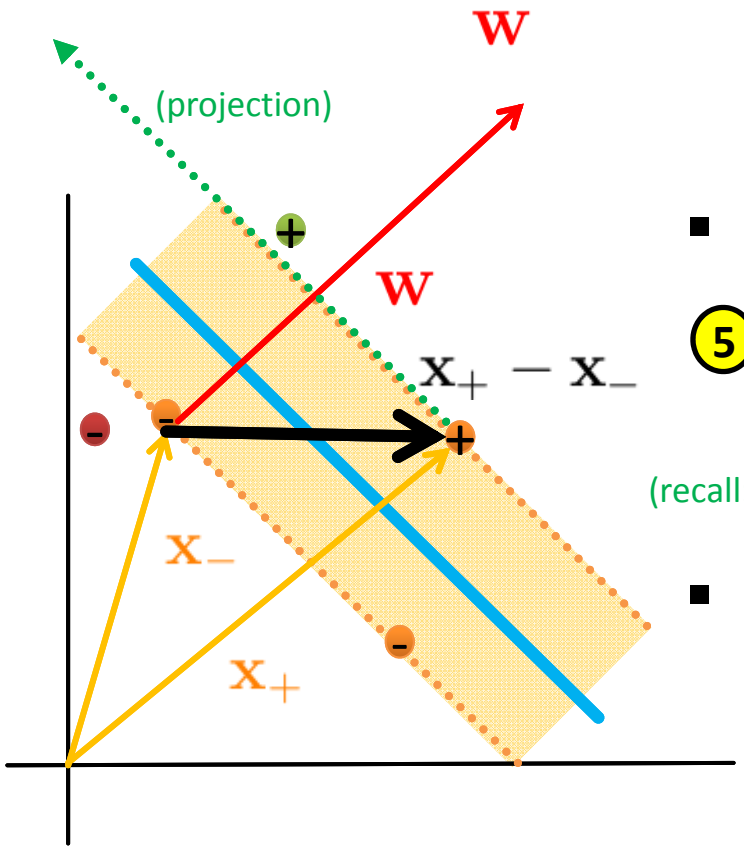
(interesting: non zero for support vectors, rest zero)

- Find derivatives for extremum & set 0
  - But two unknowns that might vary
  - First differentiate w.r.t. $\mathbf{w}$
  - Second differentiate w.r.t. b

(derivative gives the gradient, setting 0 means extremum like min)

Constrained Optimization Steps SVM (3)

- Lagrange gives: $\mathcal{L}(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum \alpha_i [y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1]$



- First differentiate w.r.t $\mathbf{w}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum \alpha_i y_i \mathbf{x}_i = 0$$

(derivative gives the gradient, setting 0 means extremum like min)

- Simple transformation brings:

(5) $$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$

(i.e. vector is linear sum of samples)

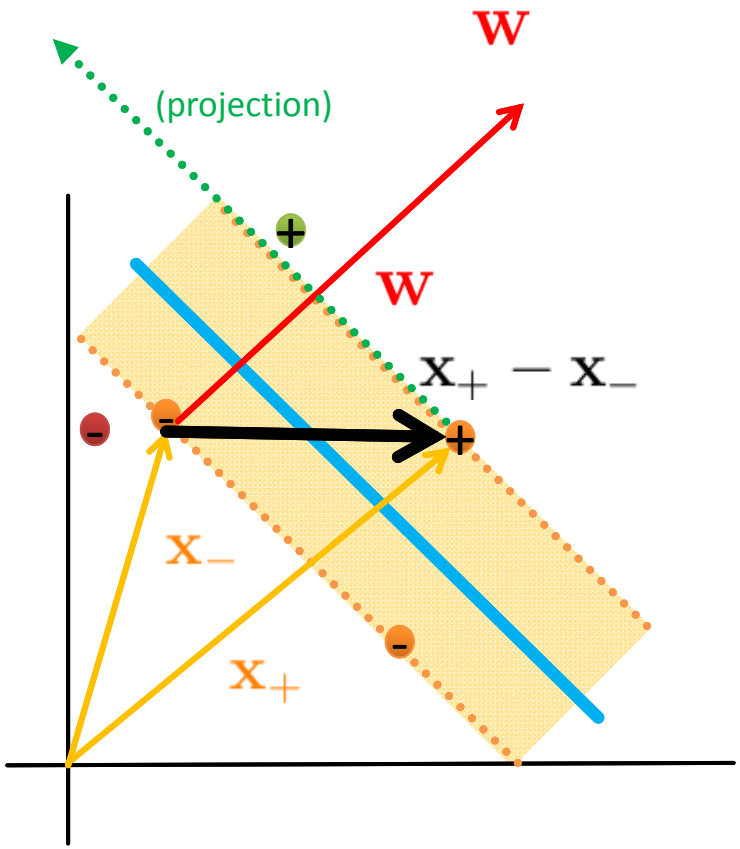
(recall: non zero for support vectors, rest zero → even less samples)

- Second differentiate w.r.t. b

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0$$ (5)

Constrained Optimization Steps SVM (4)

- Lagrange gives: $\mathcal{L}(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum \alpha_i [y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1]$



- Find minimum
  - Quadratic optimization problem
  - Take advantage of (5) $\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$

(plug into)

$$\mathcal{L} = \frac{1}{2}\left(\sum \alpha_i y_i \mathbf{x}_i\right) \cdot \left(\sum \alpha_j y_j \mathbf{x}_j\right)$$

$$- \sum \alpha_i y_i \mathbf{x}_i \cdot \left(\sum \alpha_j y_j \mathbf{x}_j\right)$$

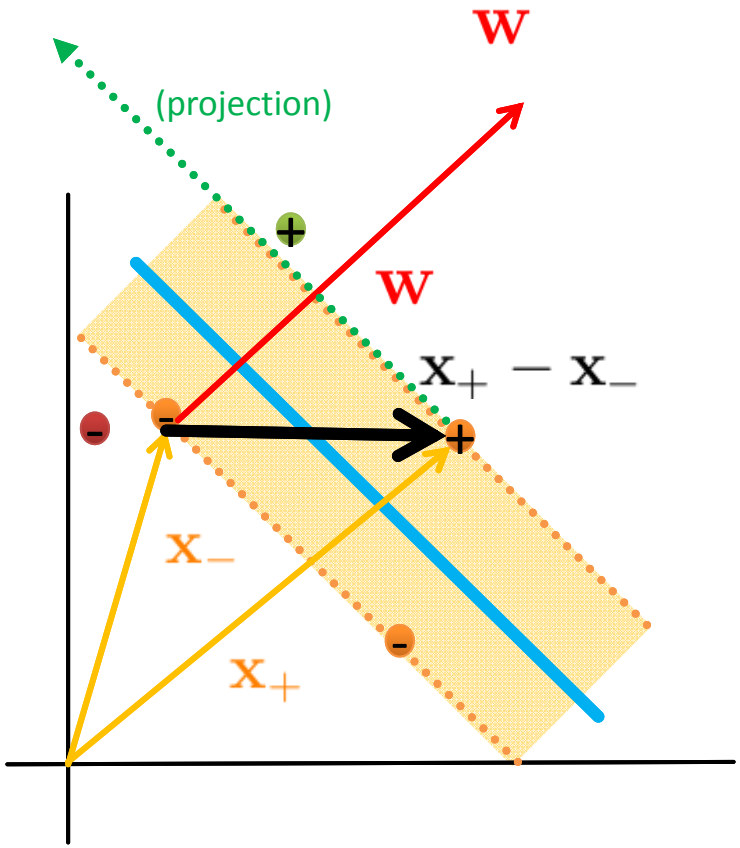
$$- \sum \alpha_i y_i b + \sum \alpha_i$$

(b constant in front sum)

(5) $$\sum \alpha_i y_i = 0$$

Constrained Optimization Steps SVM (5)

- Rewrite formula: $\mathcal{L} = \frac{1}{2}\left(\sum \alpha_i y_i \mathbf{x}_i\right) \cdot \left(\sum \alpha_j y_j \mathbf{x}_j\right) - \sum \alpha_i y_i \mathbf{x}_i \cdot \left(\sum \alpha_j y_j \mathbf{x}_j\right)$ (the same)



$$-\sum \alpha_i y_i b + \sum \alpha_i$$ (was 0)

(results in)

(optimization depends only on dot product of samples)

$$\mathcal{L} = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (6)$$

- Equation to be solved by some quadratic programming package

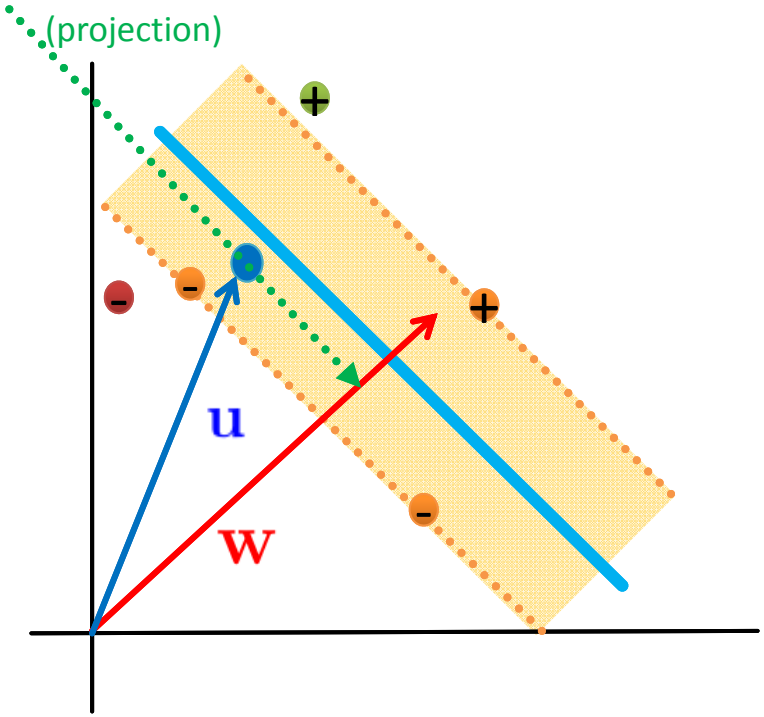
Use of SVM Classifier to Perform Classification

- Use findings for decision rule

(5) $\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$

(1) $\mathbf{w} \cdot \mathbf{u} + b \geq 0$ ⊕ ⇒ $\sum \alpha_i y_i \mathbf{x}_i \cdot \mathbf{u}_i + b \geq 0$ ⊕

(decision rule also depends on dotproduct)



Maximal Margin Classifier – Training Set and Test Set

- Classification technique
  - Given ‘labelled dataset‘
  - Data matrix X (n x p)
  - Training set:
    n training samples
  - p-dimensional space
  - Linearly seperable data
  - Binary classification problem
    (two class classification)
  - Test set:
    a vector x* with test observations

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}$$

(n x p-dimensional vectors)

$$y_1, \ldots, y_n \in \{-1, 1\}$$

(class labels) (two classes)

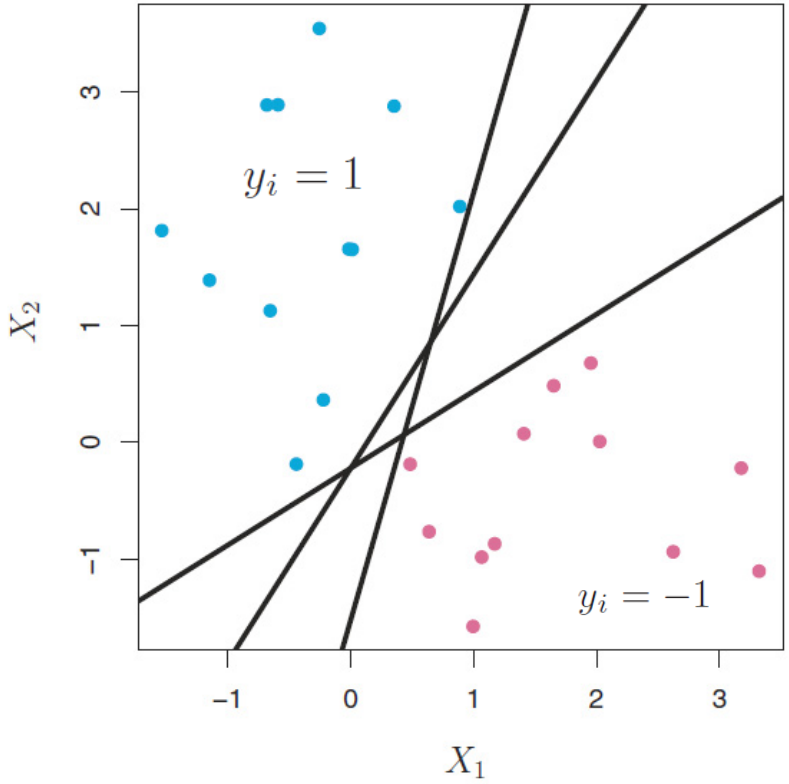
$$x^* = \begin{pmatrix} x_1^* & \ldots & x_p^* \end{pmatrix}^T$$

- **Maximal Margin Classifiers create a seperating hyperplane that seperates the training set samples perfectly according to their class labels following a reasonable way of which hyperplane to use**

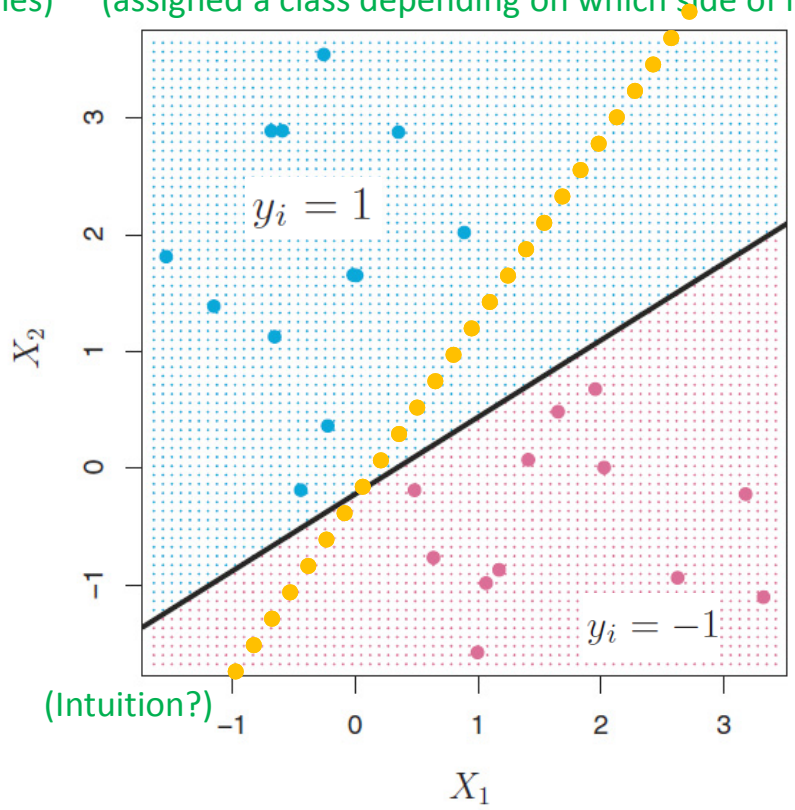
[2] An Introduction to Statistical Learning

Maximal Margin Classifier – Use of Seperating Hyperplanes

(three possible hyperplanes – any line out of infinite ones)



(assigned a class depending on which side of hyperplane)



(linear decision boundary)

(but is this the best line…?)

(Intuition?)

(properties of the seperating hyperplane)

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \text{ if } y_i = 1$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \text{ if } y_i = -1$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0 \text{ for all } i = 1, \ldots, n$$

modified from [2] An Introduction to Statistical Learning

(using testset to predict and assign labels via sign)

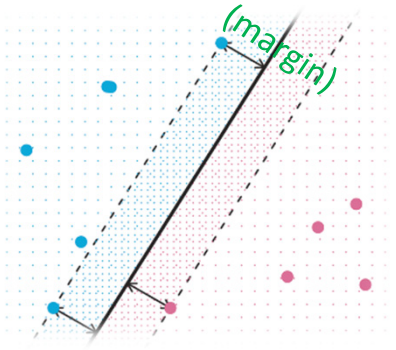
$$\text{sign of } f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$$

$$f(x^*) \text{ is negative assign it to class } -1$$

$$f(x^*) \text{ is positive assign it to class } -1$$

Maximal Margin Classifier – Reasoning and Margin Term

- Reasoning to pick the ‘best line‘
  - There exists a ‘maximal margin hyperplane‘ (optimal seperating hyperplane)
  - Hyperplane that is ‘farthest away‘ from the training set samples
- Identify the ‘margin‘ itself
  - Compute the ‘perpendicular distance’ (point ‘right angle 90 degrees‘ distance to the plane)
  - From each training sample to a given separating hyperplane
  - The smallest such distance is the ’minimal distance’ from the observations to the hyperplane – the margin
- Identify ‘maximal margin‘
  - Identify the hyperplane that has the ‘farthest minimum distance’ to the training observations
  - Also named the ‘optimal seperating hyperplane‘



- **The maximal margin hyperplane is the seperating hyperplane for which the margin is largest**

[2] An Introduction to Statistical Learning

Maximal Margin Classifier – Margin Performance

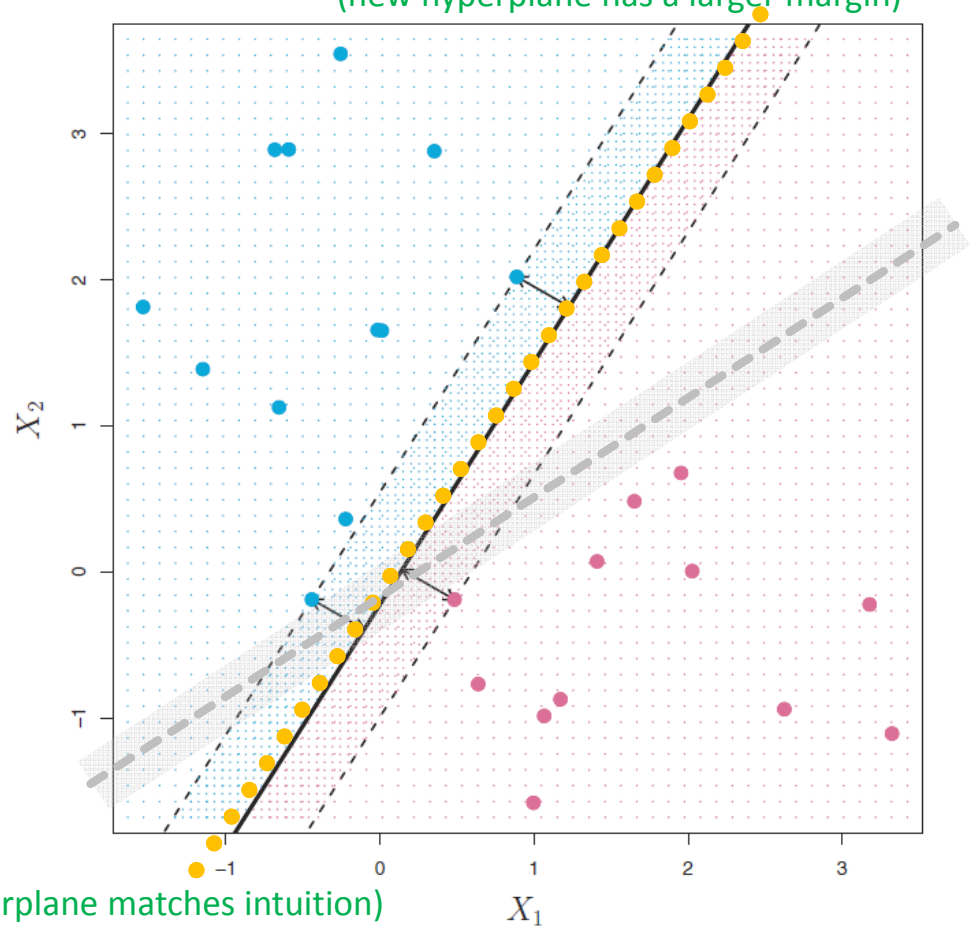
- Classification technique
  - Classify testset samples based on which side of the maximal margin hyperplane they are lying

$$\text{sign of } f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$$

(ßs are the coefficients of the maximal margin hyperplane)

  - Assuming that a classifier that has a large margin on the training data will also have a large margin on the test data (cf. also 'the intuitive notion')
- Testset samples will be thus correctly classified

(new hyperplane has a larger margin)



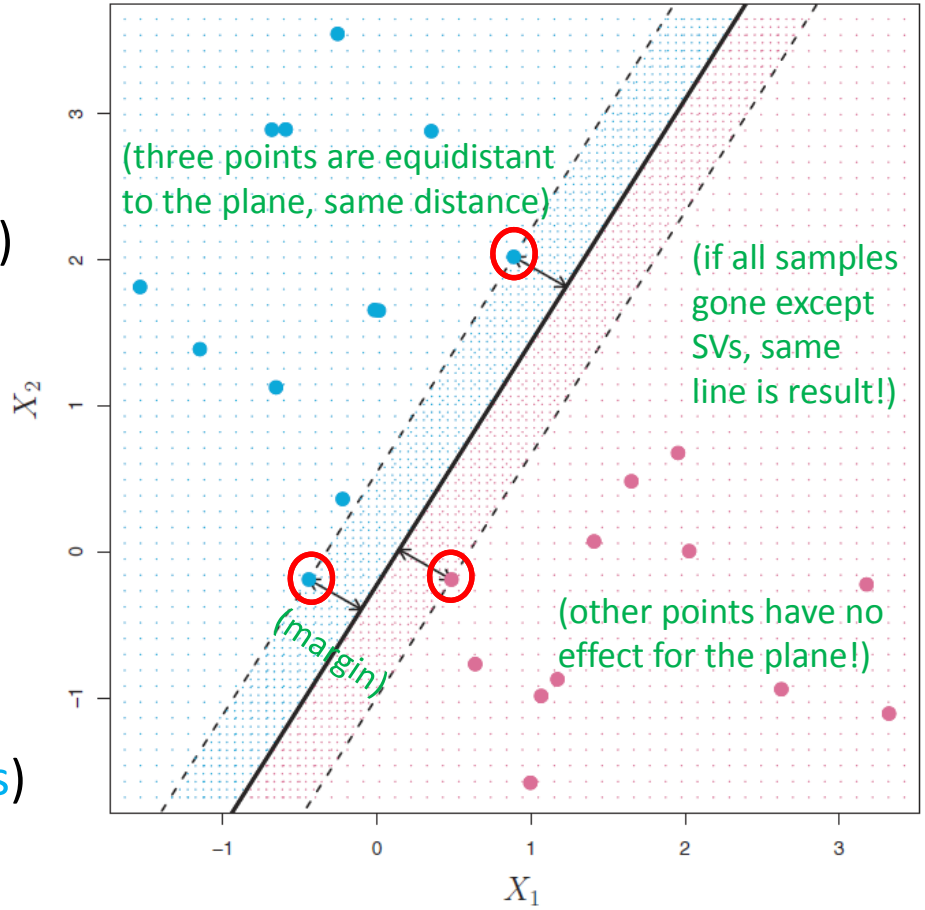
(hyperplane matches intuition)

(Compared to grey hyperplane: a 'greater minimal distance' between the data points and the seperating hyperplane)

modified from [2] An Introduction to Statistical Learning

Maximal Margin Classifier – Support Vector Term

- Observation
  - Three data points lie on the edge of margin (somewhat special data points)
  - Dashed lines indicating the width of the margin (very interesting to know)
  - Margin width is the distance from the special data points to the hyperplane (hyperplane depends directly on small data subset: SV points)



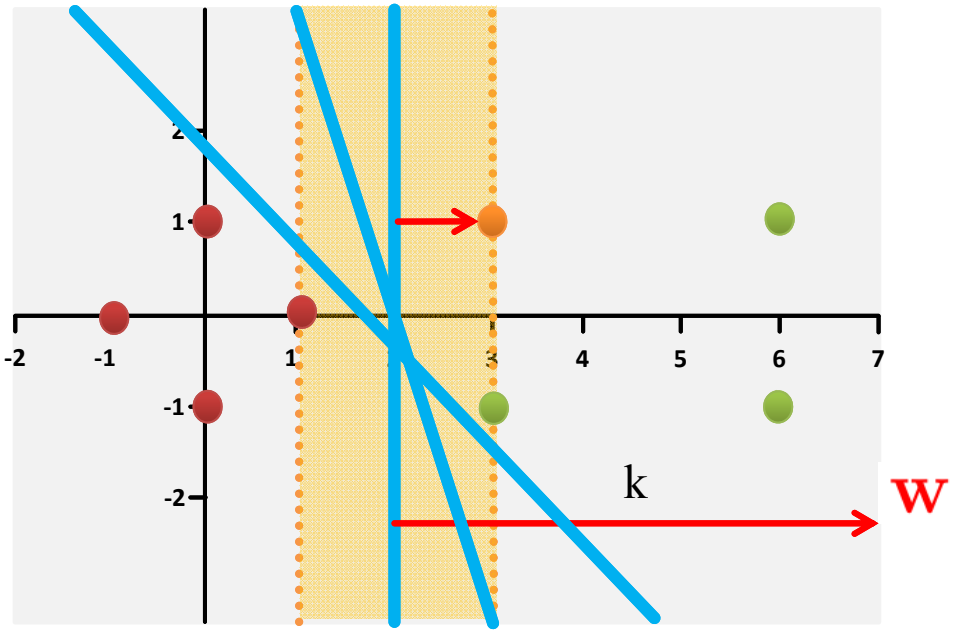
modified from [2] An Introduction to Statistical Learning

- **Points that lie on the edge of the margin are named support vectors (SVs) in p-dimensional space**
- **SVs 'support' the maximal margin hyperplane: if SVs are moved → the hyperplane moves as well**

Maximal Margin Classifier – Optimization and W Vector

- Which weight $\mathbf{w}$ maximizes the margin?
  - Margin is just a distance from 'a line to a point', goal is to minimize $\mathbf{w}$
  - Pick x_n as the nearest data point to the line (or hyper-plane)…

(distance between two dashed planes is 2 / ||w||)



- Reduce the problem to a 'constraint optimization problem' (vector $\mathbf{w}$ are the ß coefficients)

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1$$

(for points on plane w must be 0, interpret k as length of w)

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} = 0 \qquad k(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) = 0 \text{ for any } k \neq 0$$

- **Support vectors achieve the margin and are positioned exactly on the boundary of the margin**

Maximal Margin Classifier – Optimization and N Samples

- Approach: Maximizing the margin
  - Equivalent to minimize objective function
    (original and modified have same w and b)

$$\min_{w,b} \left\{ \frac{1}{2} \| \mathbf{w} \|^2 \right\}$$

(substituted for plain || w || for mathematical convenience)

subject to y_i ($\mathbf{w}$ • x_i – b) >= 1

- 'Lagrangian Dual problem' (chain of math turns optimization problem into solving this)
  - Use of already established Lagrangian method :

$$\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^T \mathbf{x}_m$$

(big data impact, important dot product)

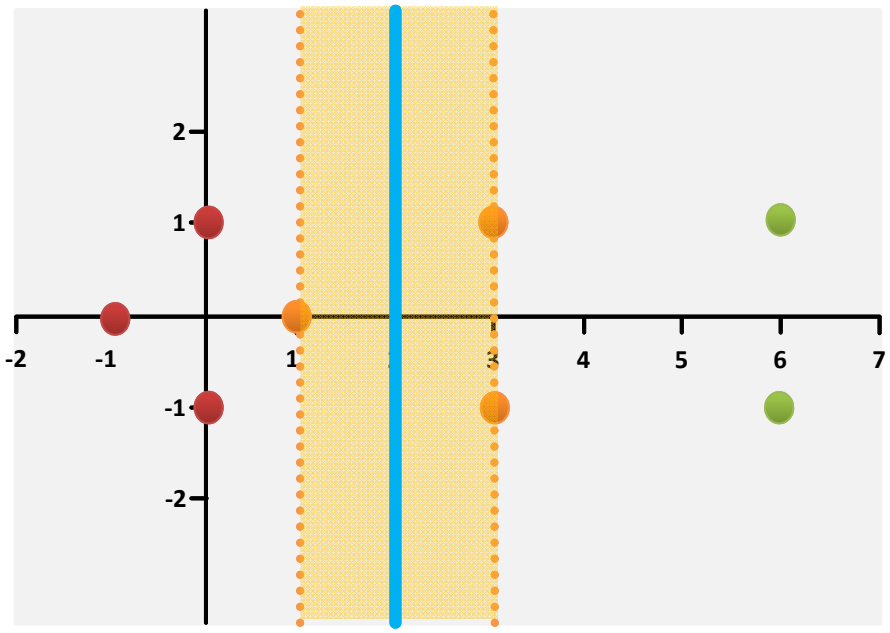
- Interesting properties
  - Simple function: Quadratic in alpha
  - Simple constraints in this optimization problem (not covered here)
  - Established tools exist: Quadratic Programming (qp) (rule of thumb)

- **Practice shows that #N moderate is ok, but large #N ('big data') are problematic for computing**
- **Quadratic programming and computing the solving depends on number of samples N in the dataset**

Maximal Margin Classifier – Optimization and # SV Impacts

- Interpretation of QP results (vector of alpha is returned)
  - The obtained values of alpha (lagrange multipliers) are mostly 0
  - Only a couple of alphas are > 0 and special: the support vectors (SVs)

(three support vectors create optimal line)



- N x N, usually not sparse
- Computational complexity relies in the following:

(big data challenge) (e.g. all datasets vs. sampling)

$$\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^T \mathbf{x}_m$$

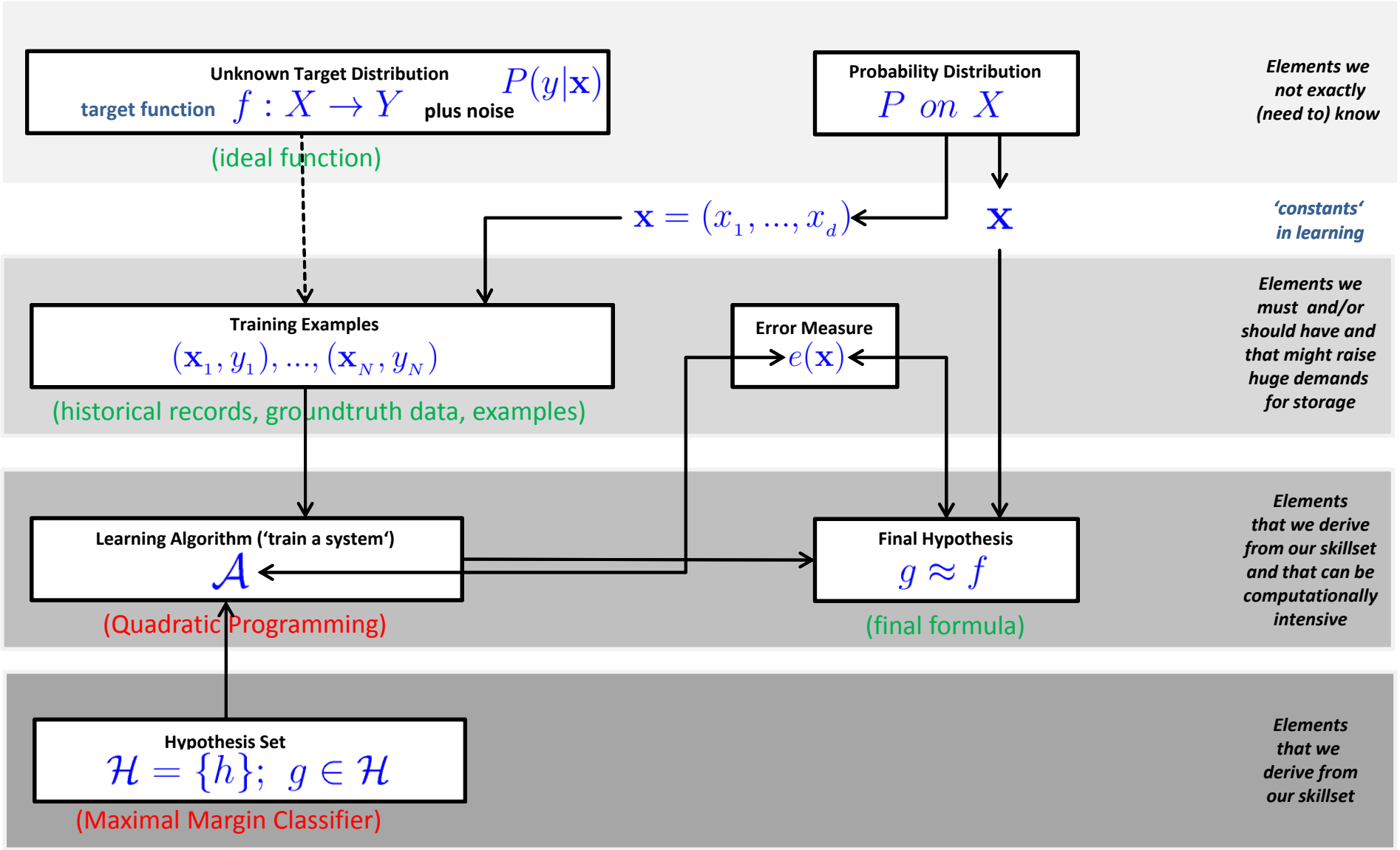
$$\begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 & \dots y_1 y_N x_1^T x_N \\ \dots & \dots & \ddots \\ y_N y_1 x_N^T x_1 & y_N y_2 x_N^T x_2 & \dots y_N y_N x_N^T x_N \end{bmatrix}$$

(quadratic coefficients, alphas are result from QP)

(rule of thumb)

- **Generalization measure: #SVs as 'in-sample quantity' → 10SVs/1000 samples ok, 500SVs/1000 bad**
- **Reasonsing towards overfitting due to a large number of SVs (fit many, small margin, gives bad E_{out})**

Solution Tools: Maximal Margin Classifier & QP Algorithm



Maximal Margin Classifier – Solving and Limitations

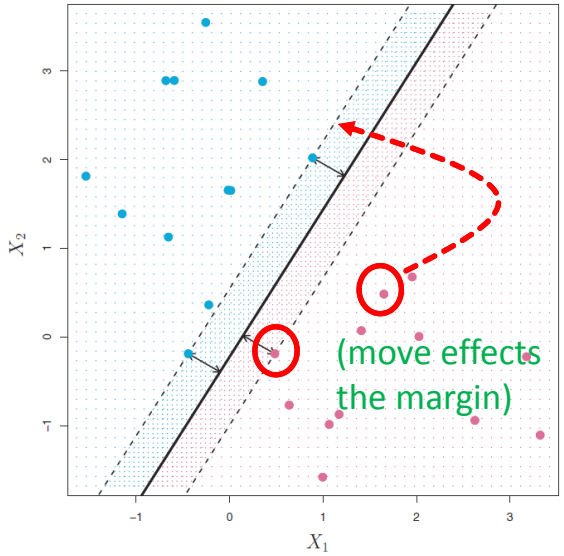
- **Solving constraint optimization problem chooses coefficients that maximize M & gives hyperplane**
- **Solving this problem efficiently is possible techniques like sequential minimal optimization (SMO)**
- **Maximal margin classifiers use a hard-margin & thus only work with exact linearly seperable data**

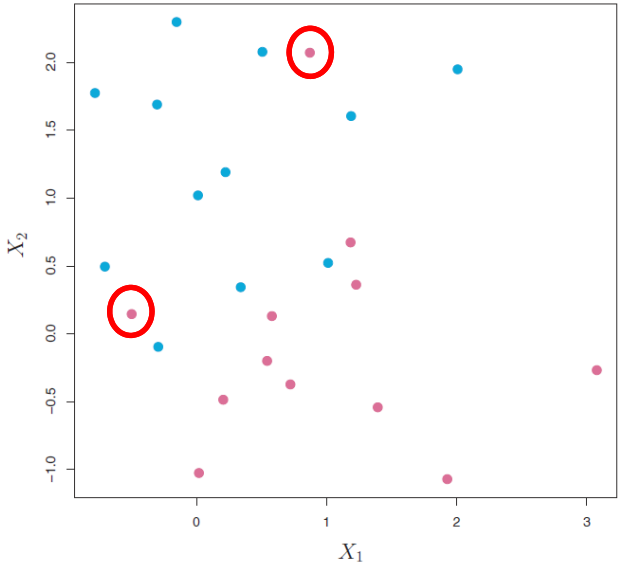
modified from [2] An Introduction to Statistical Learning

- Limitation
  - Non linearly separable data (given mostly in practice)
  - Optimization problem has no solution M > 0 (think point moves over plane)
  - No separating hyperplane can be created (classifier can not be used)

(no error allowed, a 'hard margin')

(allow some error the margin will be bigger, … maybe better E_{out})





(no exact seperation possible)

(… but with allowing some error maybe, a 'soft margin'…)

Exercises

Training Indian Pines on Golett – Job Script

- Use Indian Pines and start changing parameters
  - Parameters are equal to the serial libsvm and some additional parameters for paralellization

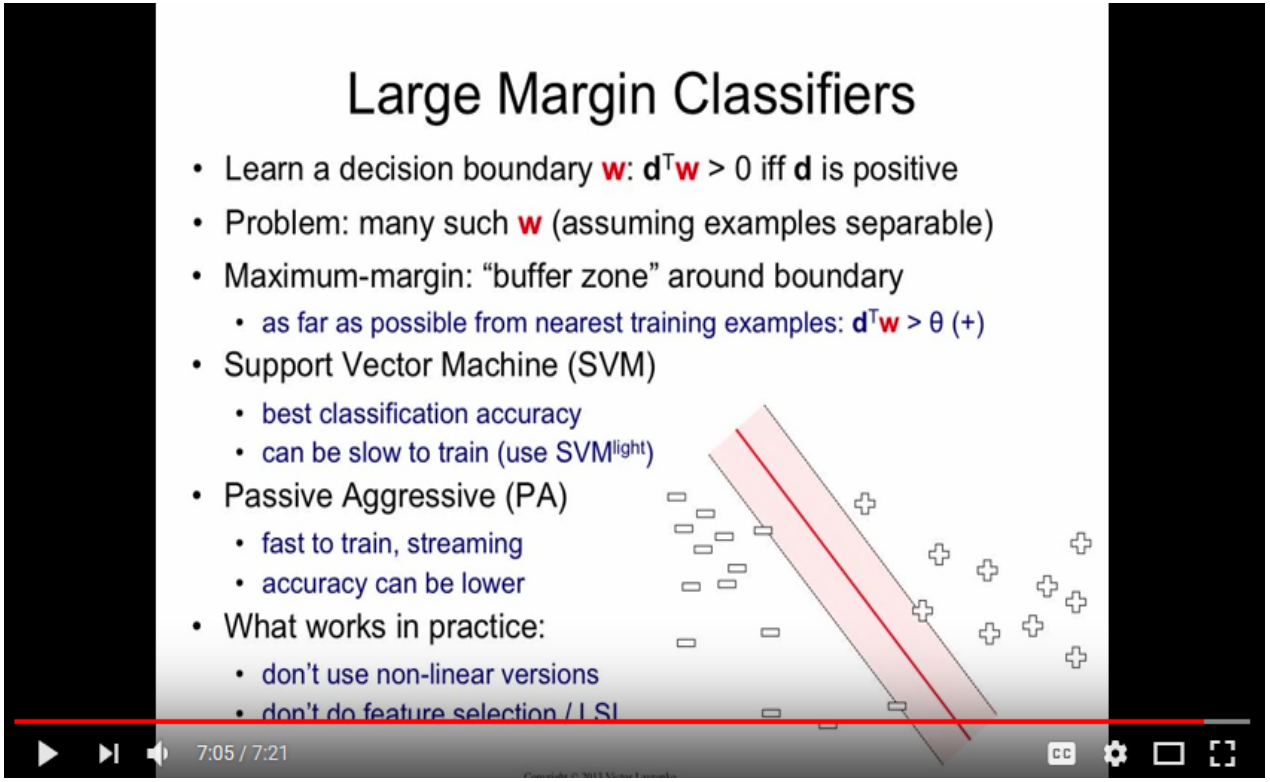
```
#!/bin/bash
#PBS -l walltime=1:0:0
#PBS -l nodes=1:ppn=all

#module load HPDBSCAN/20171110-foss-2017b
module load piSvM-JSC/1.2-20150622-intel-2017b
module load vsc-mympirun

export WORKDIR=$VSC_SCRATCH/$PBS_JOBID
mkdir -p $WORKDIR
cd $WORKDIR

# Train data
cp /apps/gent/tutorials/machine_learning/classification/Indian/indian_processed_training.el .

# by default, mympirun will use all available cores
# use --hybrid to only use a certain number of cores (per workernode)
# mympirun --hybrid 6 dbscan -e 300 -m 100 -t 12 bremenSmall.h5.h5
mympirun --hybrid 32 pisvm-train -D -o 1025 -q 512 -c 10 -g 8 -t 2 -m 1024 -s 0 indian_processed_training.el

echo "Results available in $WORKDIR"
```

Testing Indian Pines on Golett – Job Script

- Use Indian Pines and using your model files
  - Parameters are equal to the serial libsvm and some additional parameters for paralellization

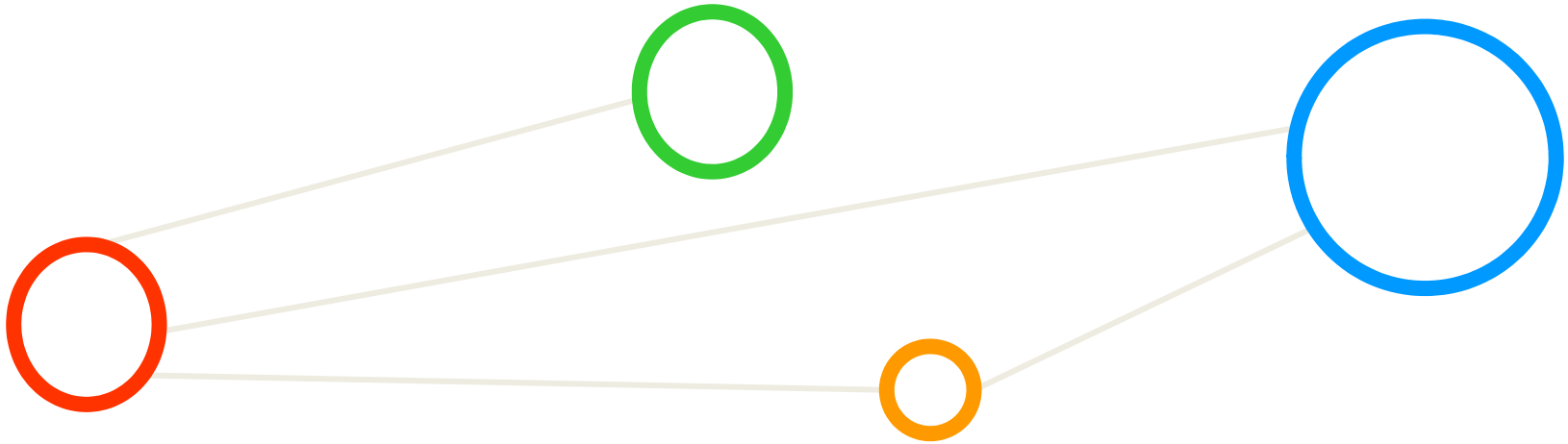
```
#!/bin/bash
#PBS -l walltime=1:0:0
#PBS -l nodes=1:ppn=all

#module load HPDBSCAN/20171110-foss-2017b
module load piSvM-JSC/1.2-20150622-intel-2017b
module load vsc-mympirun

export WORKDIR=$VSC_SCRATCH/$PBS_JOBID
mkdir -p $WORKDIR
cd $WORKDIR

# Test data
cp /apps/gent/tutorials/machine_learning/classification/Indian/indian_processed_test.el .

# Model data
cp /user/home/gent/vsc425/vsc42544/indian_processed_training.el.model .

# by default, mympirun will use all available cores
# use --hybrid to only use a certain number of cores (per workernode)
# mympirun --hybrid 6 dbscan -e 300 -m 100 -t 12 bremenSmall.h5.h5
mympirun --hybrid 32 pisvm-predict indian_processed_test.el indian_processed_training.el.model results.txt

echo "Results available in $WORKDIR"
```

[Video] Maximum Margin



[7] YouTube Video, Text Classification 2: Maximum Margin Hyperplane'

Lecture Bibliography



Lecture Bibliography

- [1] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison Wesley, ISBN 0321321367, English, ~769 pages, 2005
- [2] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [3] YouTube Video, 'Neural Networks, A Simple Explanation', Online: http://www.youtube.com/watch?v=gcK_5x2KsLA
- [5] LibSVM Webpage, Online: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- [6] EUDATB2SHARE Iris Dataset LibSVM Format Preprocessing (Record 397), Online: http://hdl.handle.net/11304/10e216d4-0a98-4ab4-86ea-75ed05ee0f46
- [7] YouTube Video, 'Text Classification 2: Maximum Margin Hyperplane', Online: https://www.youtube.com/watch?v=dQ68FW7p97A
- [8] Rome Dataset, B2SHARE, Online: http://hdl.handle.net/11304/4615928c-e1a5-11e3-8cd7-14feb57d12b9
- [9] Indian Pine Image Dataset, B2SHARE, Online: http://hdl.handle.net/11304/7e8eec8e-ad61-11e4-ac7e-860aa0063d1f


services measurement within fusion new funding
Policy-based concepts device Analysis
forms cross-disciplinary resources Centre climate Computer expertise Approaches construction Fusion
year like Computational Test E.g Computing disciplines results Enable
Cross-Disciplinary big nodes PB community Environment Modelling different
EU methods High centers Key scientific important
Science etc DKRZ Use storage basis Infrastructure files
model international simulations ENES technologies
formats physical
engineering References brain increasing via EUDAT processing IPCC
SMAQ Big Project Data Common TB
MapReduce create Methods
computing Using European
databases GB Supercomputing Work images
codes
analysis stored Resources analyze Large Summary data-intensive Service Scientific levels often
use DLCL Level external
JSC computational Research Matters Concrete performance
research e.g science HPC Juelich datasets
acquisition large-scale system Landscape
System Climate modelling AR5 Hardware distributed cores compute access hundreds Nutshell Services management
Understanding Technologies Earth Structure current network
DLCLs structures directory project General
Germany Simulation
Provide NASA Energy systems day Processing Health