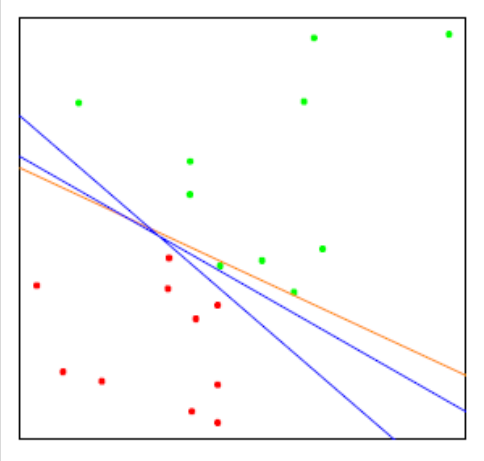


Parallel & Scalable Machine Learning

Introduction to Machine Learning Algorithms

Dr. Gabriele Cavallaro
Postdoctoral Researcher, Juelich Supercomputing Centre, Germany
Ernir Erlingsson
PhD Student, University of Iceland & DEEP-EST Application Engineer

LECTURE 12

Parallelization Benefits and Cross-Validation Practicals

March 8th, 2018
JSC, Germany

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE
JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE
DEEP Projects

Outline

Outline of the Course

1. Introduction to Machine Learning Fundamentals
2. PRACE and Parallel Computing Basics
3. Unsupervised Clustering and Applications
4. Unsupervised Clustering Challenges & Solutions
5. Supervised Classification and Learning Theory Basics
6. Classification Applications, Challenges, and Solutions
7. Support Vector Machines and Kernel Methods
8. Practicals with SVMs
9. Validation and Regularization Techniques
10. Practicals with Validation and Regularization
11. Deep Learning Introduction
12. Parallelization Benefits and Cross-Validation Practicals

Day One – beginner

Day Two – moderate

Day Three – expert

Outline

- Parallelization Benefits
  - Regularization Parameters Revisited
  - Cross-Validation Technique Revisited
  - First Level of Parallelization in piSVM
  - Second Level of Parallelization in GridSearch

- Cross-Validation Practicals
  - Gridsearch as a team
  - Assign teams combinations of parameters
  - Parameter C and Gamma on Indian Pines
  - Gridsearch Benefits Revisited

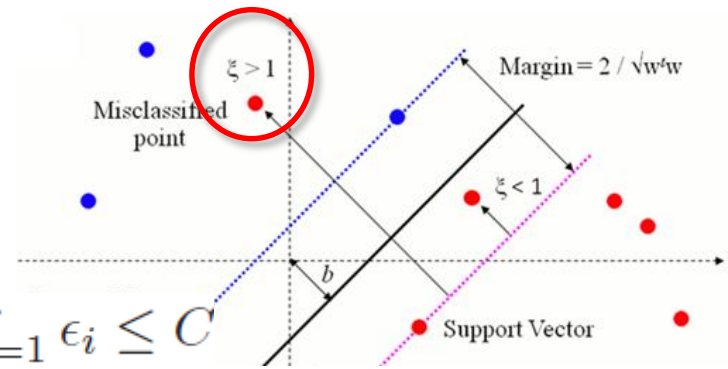
Outline

- Short Course Summary
  - Machine Learning Fundamentals
  - Unsupervised Clustering with HPDBSCAN
  - Supervised Classificaiton with PISVM
  - Statistical Learning Theory Basica
  - Regularization & Validation

Regularization Revisited & Kernel Parameter Gamma

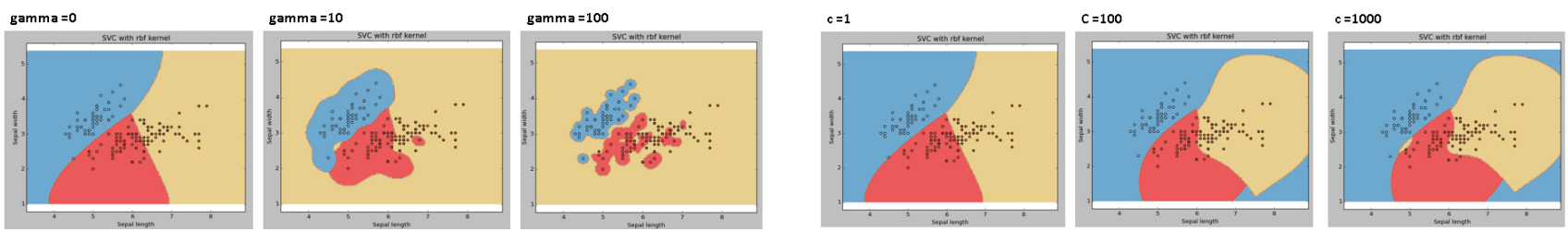
- C = 0 (too rectrictive, potentially bad for E_{out}) $\epsilon_1 = \ldots = \epsilon_n = 0$
  - No budget/costs for violations: comparable to maximal margin classifier
  - Further constraint: only works in linearly seperable cases (less in practice)
- C > 0 (flexible option, better for E_{out})
  - No more than C data points can be on the wrong side of the hyperplane ('how much misclassifications allowed')
  - Reasoning: if an observation is on the wrong side then $\epsilon_i > 1$

$$\sum_{i=1}^{n} \epsilon_i \leq C$$



- Kernel Parameter Gamma
  - RBF Kernel element, example Iris Dataset

[3] SVM and Parameters

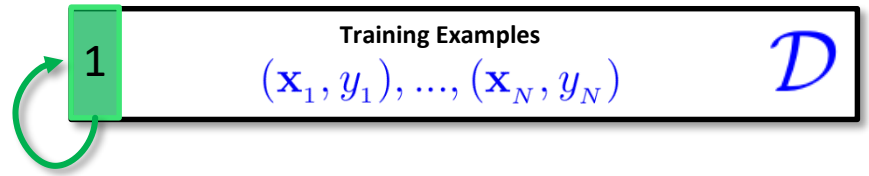


Validation Technique – Cross-Validation – Revisited

- **10-fold cross validation is mostly applied in practical problems by setting K = N/10 for real data**
- **Having N/K training sessions on N – K points each leads to long runtimes (→ use parallelization)**

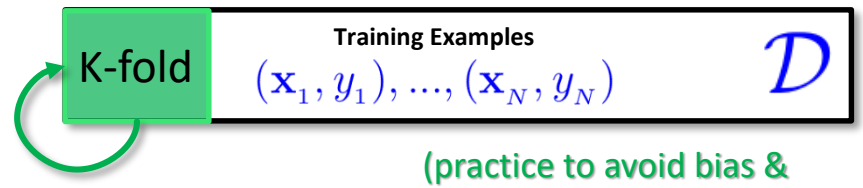
- Leave-one-out
  - N training sessions on
    N – 1 points each time

(leave 1 point out at each run → many runs)



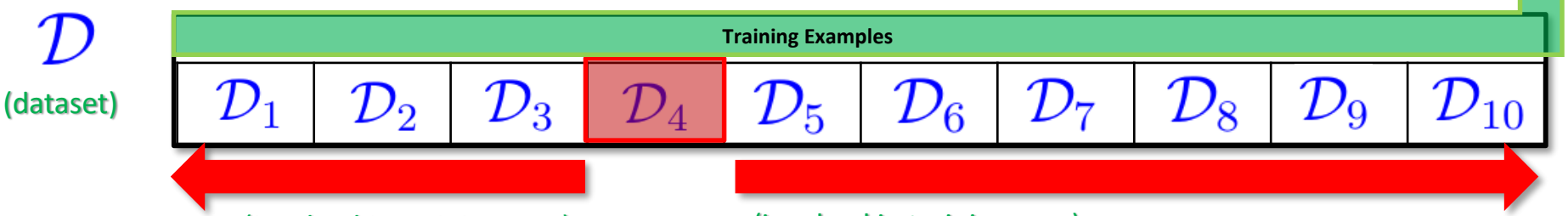
- Leave-more-out
  - Break data into number of folds
  - N/K training sessions on
    N – K points each time (fewer training sessions than above)

(generalization to leave k points out at each run)



(practice to avoid bias & contamination: some rest for test as 'unseen data')

  - Example: '10-fold cross-validation' with K = N/10 multiple times (N/K)
    (use 1/10 for validation, use 9/10 for training, then another 1/10 … N/K times)



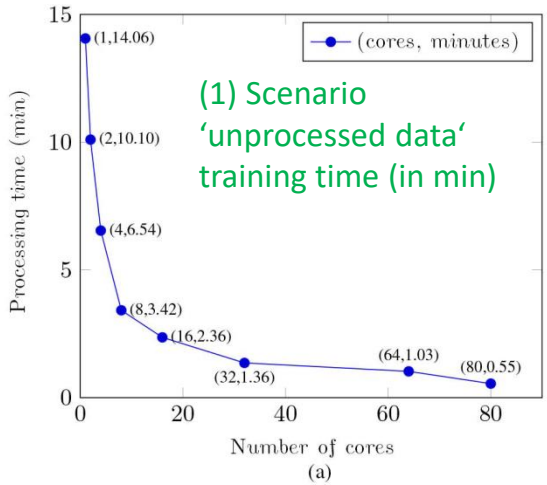
(involved in training now)

(involved in training now)

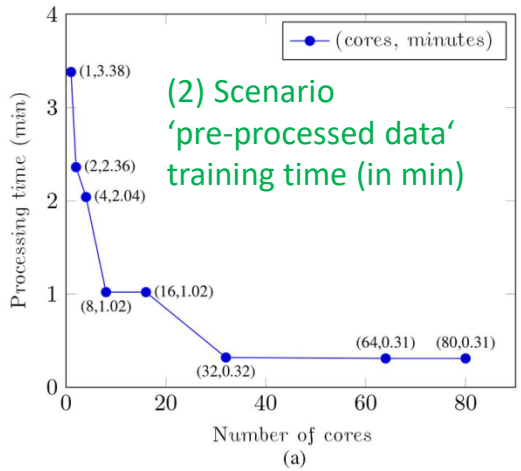
(now is the current example run)

Parallelization Benefit: Lower-Time-To-Solution

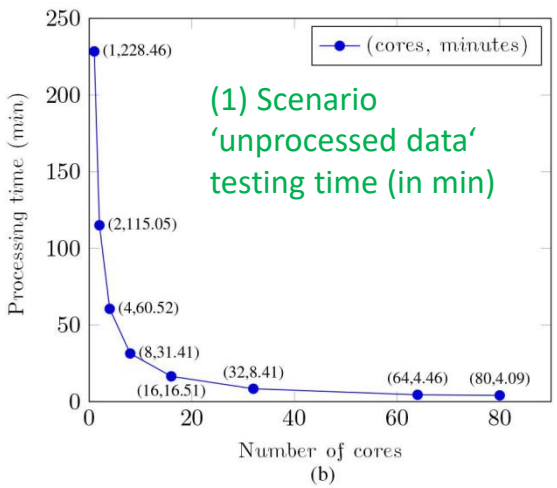
- Major speed-ups; ~interactive (<1 min); same accuracy;



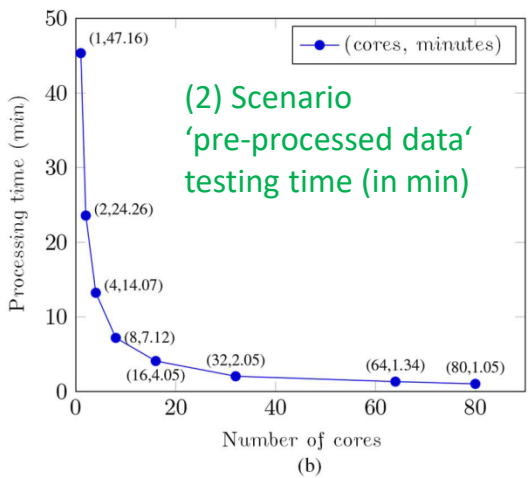
(1) Scenario
'unprocessed data'
training time (in min)



(2) Scenario
'pre-processed data'
training time (in min)



(1) Scenario
'unprocessed data'
testing time (in min)



(2) Scenario
'pre-processed data'
testing time (in min)

manual & serial activities (in min)

	kpca	esdap	nwfe	10x CSV	Training	Test	Total
(1) Scenario	0	0	0	4.47×10^3	10.45	71.08	4.55×10^3
(2) Scenario	5	15.38	1	529.55	1.37	23.25	575.55

'big data' is not always better data

	(1) Scenario	(2) Scenario
Number of features	200	30
Overall Accuracy (%)	40.68	77.96

(cf. Importance of feature engineering above)

[1] G. Cavallaro and M. Riedel, et al. , 2015

(aka first level of parallelism)

Parallelization Benefit: Parallel 10-Fold Cross-Validation

- Example: 2 Parameters, 10-fold cross-validation
  - 2 x benefits of parallelization possible in a so-called 'gridsearch'
  - (1) Compute parallel; (2) Do all cross-validation runs in parallel (all cells)
  - Evaluation between Matlab (aka 'serial laptop') & parallel (80 cores)

(2) Scenario 'pre-processed data', 10xCV **serial**: accuracy (min)

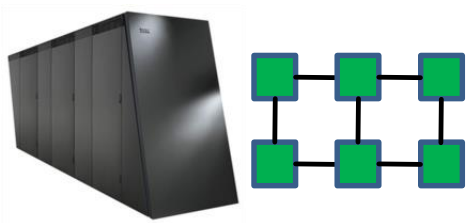
γ/C	1	10	100	1000	10 000
2					
4					
8					
16					
32					

In Matlab one after another

(1) First Result: best parameter set from 14.41 min to 1.02 min

(2) Second Result: all parameter sets from ~9 hours to ~35 min

[1] G. Cavallaro and M. Riedel, et al. , 2015



(2) Scenario 'pre-processed data', 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000

second level of parallelism

'(1) each cell inherent parallel'

'(2) all cells in parallel'

- **10-fold cross-validation achieves parallelization benefits (1) in each grid cell and (2) across all cells**

Parallelization Summary – aka ‘GridSearch‘

- Parallelization benefits are enormous for complex problems
  - Enables feasibility to tackle extremely large datasets & high dimensions
  - Provides functionality for a high number of classes (e.g. #k SVMs)
  - Achieves a massive reduction in time → lower time-to-solution

(1) Scenario ‘unprocessed data‘, 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	second level of parallelism				
4					
8					
16					
32					

(2) Scenario ‘pre-processed data‘, 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	second level of parallelism				
4					
8					
16					
32					

(1) Scenario ‘unprocessed data‘’10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	second level of parallelism				
4					
8					
16					
32					

First Result: best parameter set from 118.28 min to 4.09 min
Second Result: all parameter sets from ~3 days to ~2 hours

(2) Scenario ‘pre-processed data‘, 10xCV **parallel**: accuracy (min)

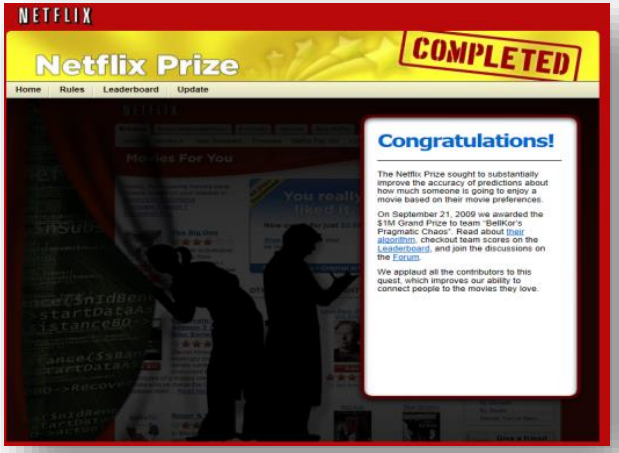
γ/C	1	10	100	1000	10 000
2	second level of parallelism				
4					
8					
16					
32					

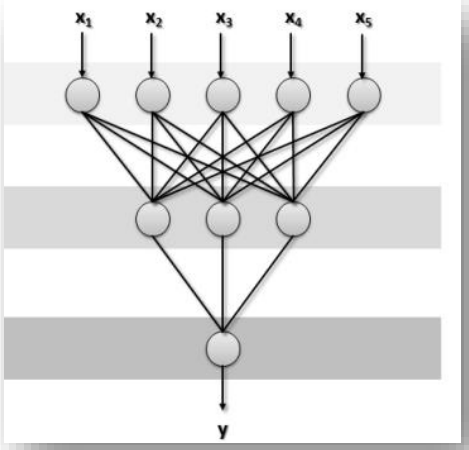
First Result: best parameter set from 14.41 min to 1.02 min
Second Result: all parameter sets from ~9 hours to ~35 min

[1] G. Cavallaro and M. Riedel, et al. , 2015

Complex Application Example in Industry – Netflix

- ~2009 - Netflix Prize Challenge 2009
  - Data: Netflix company provided data to learn from previous movie rentals
  - Challenge: Improve Netflix in-house movie recommender system
  - Prize: 1.000.000 US $ for team with 10% improvements
  - Approaches: Machine learning algorithms and collaborative filterings
  - Winner: Prize received by working with Artificial Neural Network (ANNs)

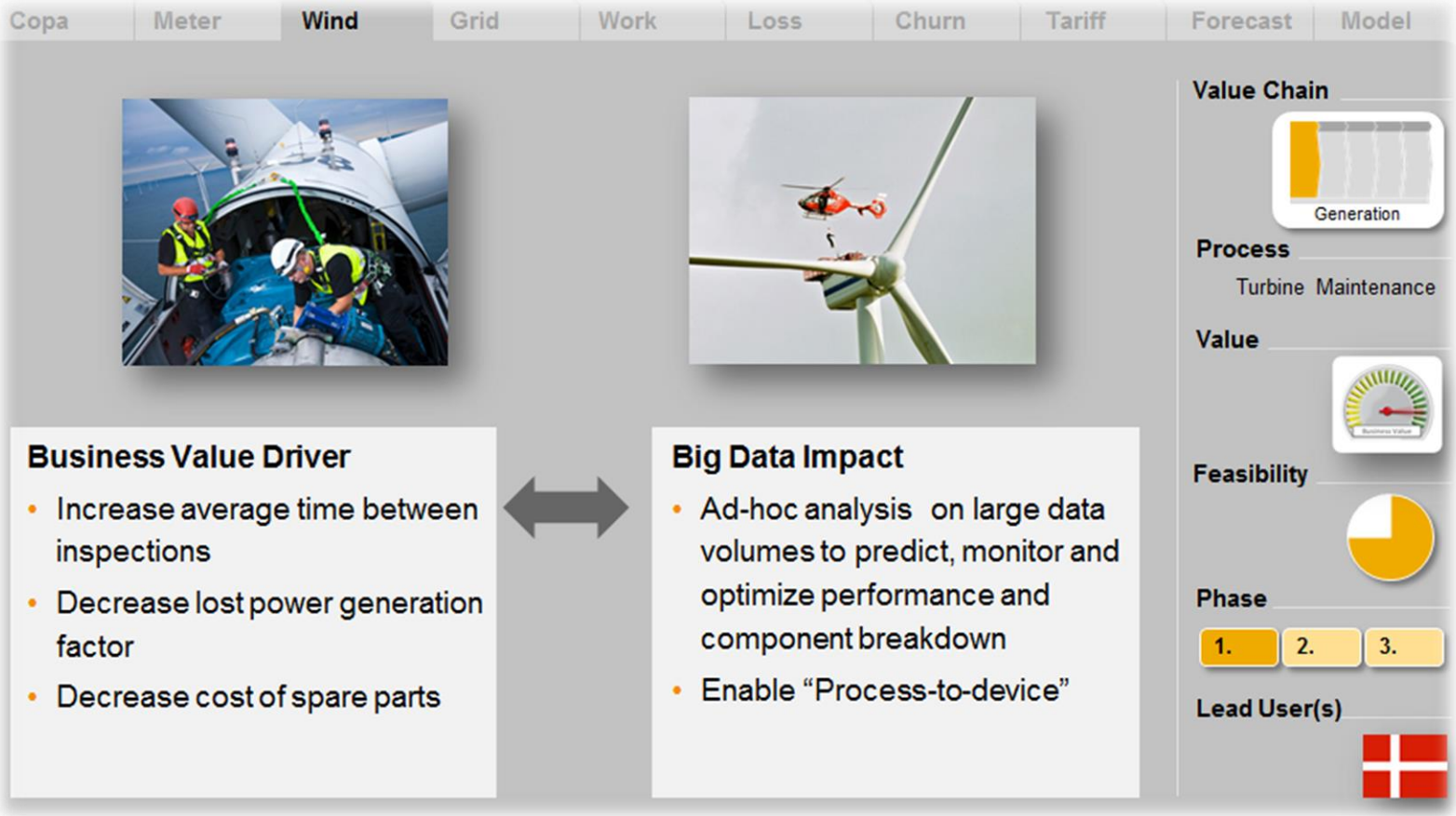




[4] A. Töscher and M. Jahrer, 'The BigChaos Solution to the Netflix Grand Prize', 2009

Complex Application Example in Industry – Windpower

- Predictive & Instant Maintenance Workforce Management

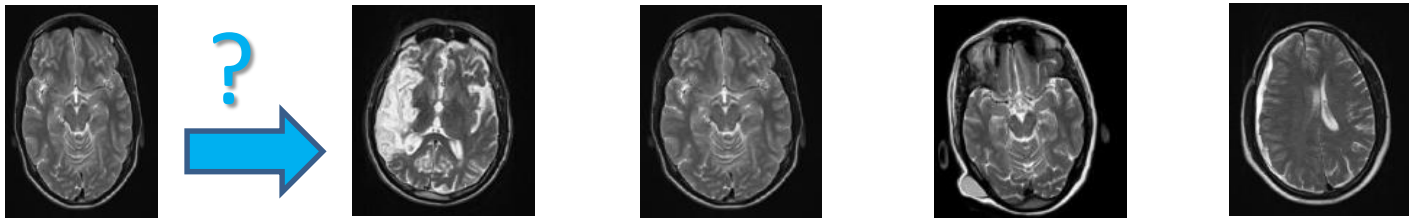


➢ **Slide courtesy of Dr. S. Fischer, Global Head of Applied Research – SAP AG, Germany SDIL**

Complex Application Examples in Science & Engineering

- Classification of Abnormalities in Brain MRI Images
  - Using Support Vector Machines (SVMs)
  - 'Classify images between normal and abnormal along with type of disease depending upon features.'

[5] D. Singh et al., 2012



input data | class Infected by clot | class normal brain | class Infected by tumor | class Infected by bleed

- Classification of buildings from multi-spectrial satellite data
  - Using Support Vector Machines (SVMs)
  - Classify land cover using image data & data preprocessing methods

Class	Training	Test
Buildings	18126	163129
Blocks	10982	98834
Roads	16353	147176
Light Train	1606	14454
Vegetation	6962	62655
Trees	9088	81792
Bare Soil	8127	73144
Soil	1506	13551
Tower	4792	43124
Total	77542	697859

?

different types of land cover

Cross-Validation Practicals

JURECA System – SSH Login

- Use your account train004 - train050
- Windows: use mobaxterm or putty
- UNIX: ssh trainXYZ@jureca.fz-juelich.de
- Example

```
adminuser@linux-8djg:~> ssh train001@jureca.fz-juelich.de
Warning: the ECDSA host key for 'jureca.fz-juelich.de' differs from the key for the IP address '134.94.33.
9'
Offending key for IP in /home/adminuser/.ssh/known_hosts:12
Matching host key in /home/adminuser/.ssh/known_hosts:19
Are you sure you want to continue connecting (yes/no)? yes
]Last login: Mon Aug 21 14:29:03 2017 from zam2036.zam.kfa-juelich.de
*****************************************************************************
*                            Welcome to JURECA                              *
*                                                                           *
*  Information about the system, latest changes, user documentation and FAQs:  *
*                  http://www.fz-juelich.de/ias/jsc/jureca                  *
*****************************************************************************
*                           ### Known Issues ###                            *
*                                                                           *
* An up-to-date list of known issues on the system is maintained at         *
*               http://www.fz-juelich.de/ias/jsc/jureca-known-issues        *
* Open issues:                                                              *
*   - Intel compiler error with std::valarray and                           *
*     optimized headers, added 2016-03-20                                   *
```

➢ **Remember to use your own trainXYZ account in order to login to the JURECA system**

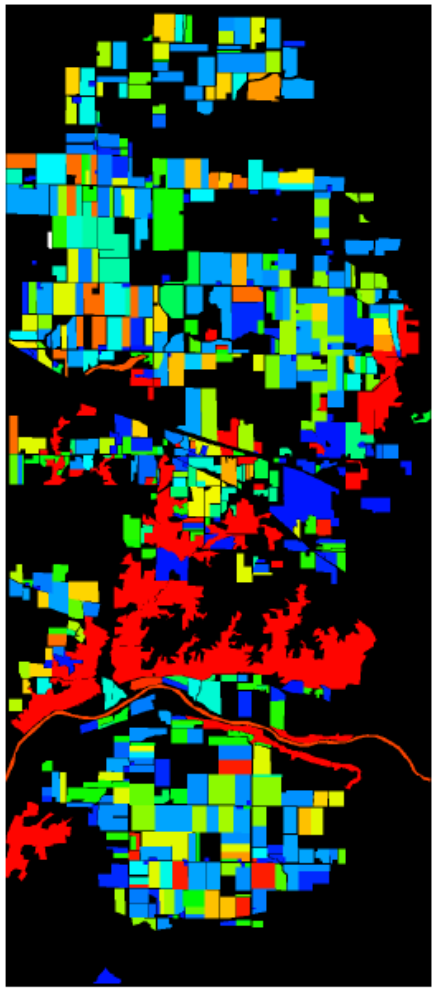
Indian Pines Dataset – Preprocessing

Corrected by JPL

- 1417×617 pixels (∼600 MB)
- 200 bands (20 discarded, with low SNR)
- 58 classes (6 discarded, with $\leq$ 100 samples)

Class		Number of samples		Class		Number of samples	
number	name	training	test	number	name	training	test
1	Buildings	1720	15475	27	Pasture	1039	9347
2	Corn	1778	16005	28	pond	10	92
3	Corn?	16	142	29	Soybeans	939	8452
4	Corn-EW	51	463	30	Soybeans?	89	805
5	Corn-NS	236	2120	31	Soybeans-NS	111	999
6	Corn-CleanTill	1240	11164	32	Soybeans-CleanTill	507	4567
7	Corn-CleanTill-EW	2649	23837	33	Soybeans-CleanTill?	273	2453
8	Corn-CleanTill-NS	3968	35710	34	Soybeans-CleanTill-EW	1180	10622
9	Corn-CleanTill-NS-Irrigated	80	720	35	Soybeans-CleanTill-NS	1039	9348
10	Corn-CleanTilled-NS?	173	1555	36	Soybeans-CleanTill-Drilled	224	2018
11	Corn-MinTill	105	944	37	Soybeans-CleanTill-Weedy	54	489
12	Corn-MinTill-EW	563	5066	38	Soybeans-Drilled	1512	13606
13	Corn-MinTill-NS	886	7976	39	Soybeans-MinTill	267	2400
14	Corn-NoTill	438	3943	40	Soybeans-MinTill-EW	183	1649
15	Corn-NoTill-EW	121	1085	41	Soybeans-MinTill-Drilled	810	7288
16	Corn-NoTill-NS	569	5116	42	Soybeans-MinTill-NS	495	4458
17	Fescue	11	103	43	Soybeans-NoTill	216	1941
18	Grass	115	1032	44	Soybeans-NoTill-EW	253	2280
19	Grass/Trees	233	2098	45	Soybeans-NoTill-NS	93	836
20	Hay	113	1015	46	Soybeans-NoTill-Drilled	873	7858
21	Hay?	219	1966	47	Swampy Area	58	525
22	Hay-Alfalfa	226	2032	48	River	311	2799
23	Lake	22	202	49	Trees?	58	522
24	NotCropped	194	1746	50	Wheat	498	4481
25	Oats	174	1568	51	Woods	6356	57206
26	Oats?	34	301	52	Woods?	14	130

[1] G. Cavallaro and M. Riedel, et al. , 2015



(non-linearly separable) dataset

Publicly Available Datasets – Location

- *Indian Pines Dataset Raw and Processed*

B2SHARE
Store and Share Research Data

[2] Indian Pines Raw and Processed

Indian pines: raw and processed

by [Unknown]

Dec 22, 2016

Last updated at Jan 11, 2018

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

PID: 11304/7e8eec8e-ad61-11e4-ac7e-860aa0063d1f Copy

EUDAT

Files

Name	Size
indian_processed_test.el	105.59MB
indian_processed_training.el	11.73MB
indian_raw_test.el	747.13MB
indian_raw_training.el	83.01MB

Basic metadata

Open Access	True ✔	
License		
Contact Email		
Publication Date	2015-02-04	
Contributors		
Resource Type	Category	Other
Alternate identifiers	172	
	Type	B2SHARE_V1_ID
	http://hdl.handle.net/11304/9ec5eac8-61b4-4617-ae1c-1f8c8cd3cd74	
	Type	ePIC_PID
Publisher	https://b2share.eudat.eu	
Language	en	

```
[train001@jrl07 indianpines]$ pwd
/homea/hpclab/train001/data/indianpines
[train001@jrl07 indianpines]$ ls -al
total 925344
drwxr-xr-x 2 train001 hpclab       512 Jul  7  2016 .
drwxr-xr-x 5 train001 hpclab       512 Jan 14 13:10 ..
-rw-r--r-- 1 train001 hpclab        36 Jul  7  2016 b2share.txt
-rw-r--r-- 1 train001 hpclab 105594346 Feb  5  2015 indian_processed_test.el
-rw-r--r-- 1 train001 hpclab  11732509 Feb  5  2015 indian_processed_training.el
-rw-r--r-- 1 train001 hpclab 747125597 Feb  5  2015 indian_raw_test.el
-rw-r--r-- 1 train001 hpclab  83014311 Feb  5  2015 indian_raw_training.el
```

Exercises – Gridsearch Cross-Validation Runs – Indian Pines

Exercises – Cross-Validation Runs – Gridsearch Rome

C GAMMA	0.1	1	10	100	1000	10000
1						
2						
4						
8						
16						
32						
64						

Homework:

Please fill the grid with your values
and find out the best parameters

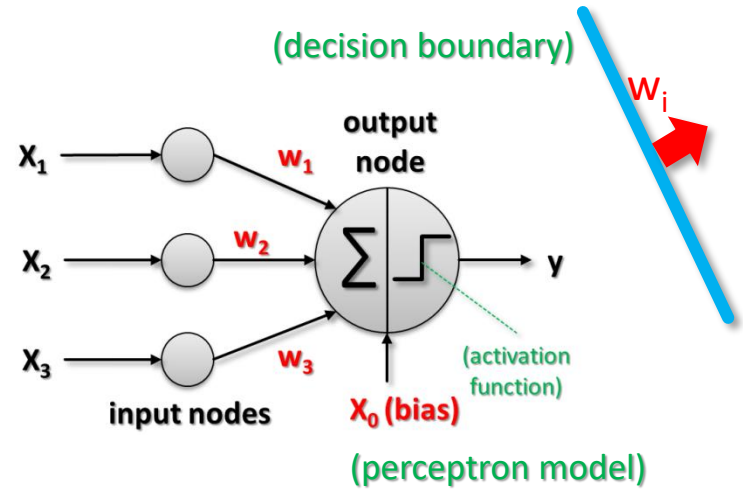
Exercises – Cross-Validation Runs – Gridsearch Indian Pines

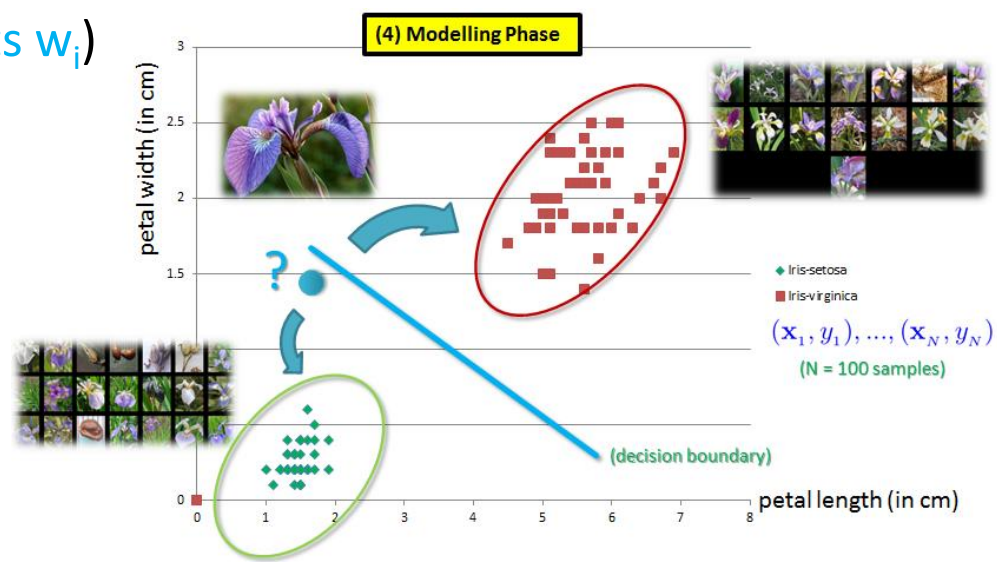
C GAMMA	0.1	1	10	100	1000	10000
1						
2						
4						
8						
16						
32						
64						

Short Course Summary

Lecture 1 – Introduction to Machine Learning Fundamentals

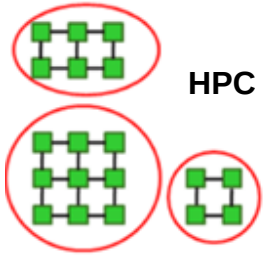
- Machine Learning Prerequisites
  1. Some pattern exists
  2. No exact mathematical formula
  3. Data exists
- Linearly seperable dataset Iris
  - Perceptron Model as hypothesis (simplest linear learning model, linearity in learned weights w_i)
  - Understood Perceptron Learning Algorithm (PLA)
- Learning Approaches
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning



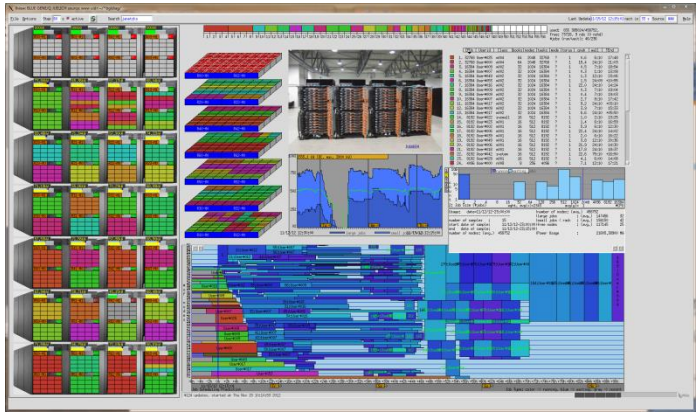


Lecture 2 – PRACE and Parallel Computing Basics

- PRACE
  - High Performance Computing (HPC)
  - Pan-European supercomputing infrastructure
  - Offers HPC resources on leading edge capability systems
  - Peer-review of research proposals
- JURECA (Mon, Wed) & JUROPA3 (Tue)
  - Tutorial HPC systems
  - Accounts train004-train050
- Scheduler & Job scripts
  - Define parallel job properties
  - Reservation in job scripts change daily
  - Job submit with sbatch <jobscript>; check status with squeue –u <userid>



Network interconnection important



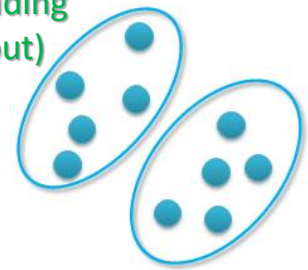
[7] LLView Tool

Lecture 3 – Unsupervised Clustering and Applications

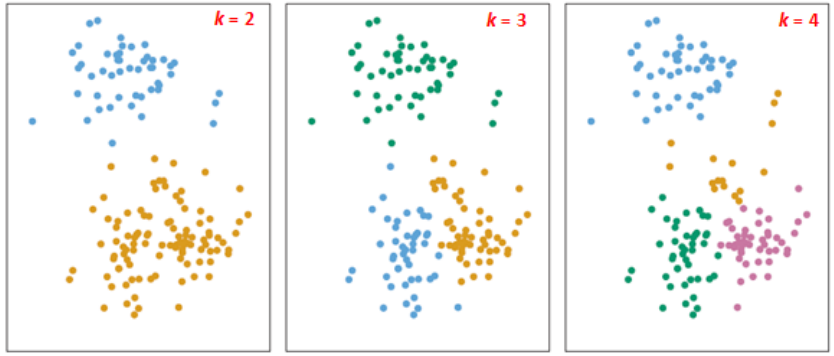
- Unsupervised Clustering
  - Input $\mathbf{x} = x_1, \ldots, x_d$
  - No output (unsupervised)
  - Data $(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$
- K-Means Clustering
  - Problem of assigning K as numbers of clusters
  - Simple, works only sometimes
- DBSCAN Clustering
  - Finds arbitrary shapes and numbers of clusters
  - Still two parameters
  - Parallel version scales for big pointcloud datasets

(serial statistical computing with R tool)

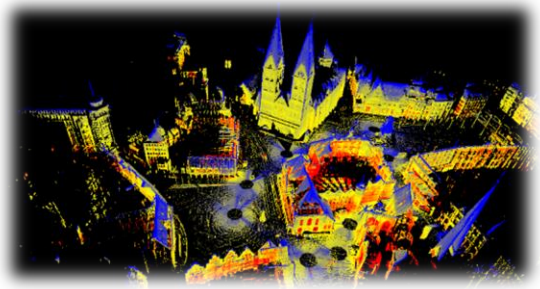
(no guiding output)



(number of cluster can be amiguities sometimes)

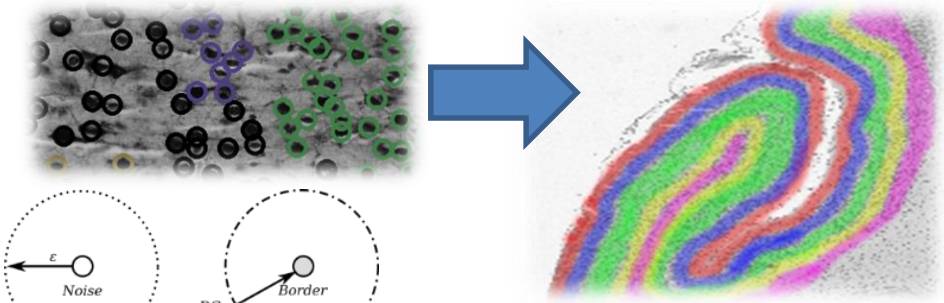


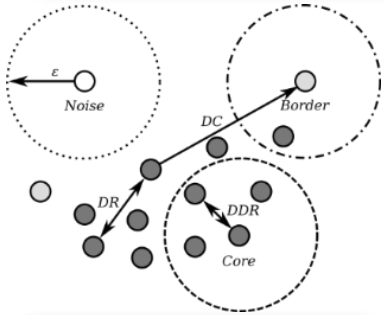
[4] An Introduction to Statistical Learning

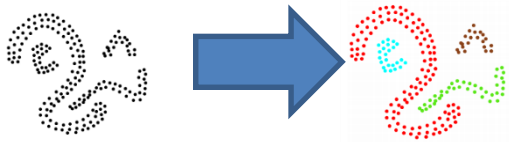


Lecture 4 – Unsupervised Clustering, Challenges & Solutions

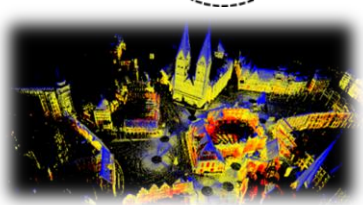
- Unsupervised Clustering
  - K-Means & K-Median
  - DBSCAN very effective
  - Applications in Context
  - Parameter Changes minPoints & Epsilon

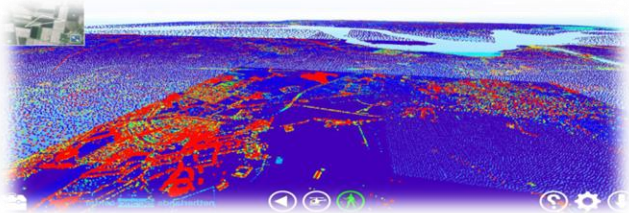


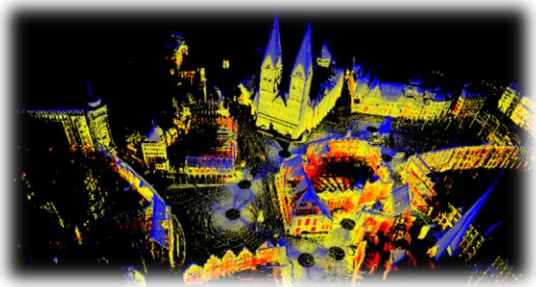




- Point Cloud Datasets
  - 3D/4D laser scans
  - Cities, Buildings, etc.
  - Bremen small & big datasets
  - Big Data: Whole Countries (e.g. Netherlands)

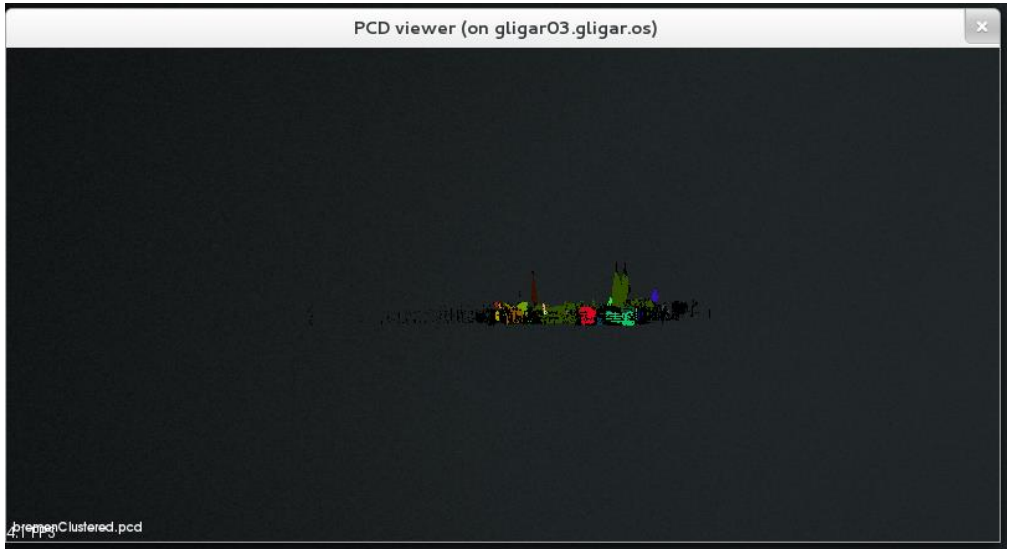






Lecture 4 – Python Script HDF5 → PCL for PCL Viewer

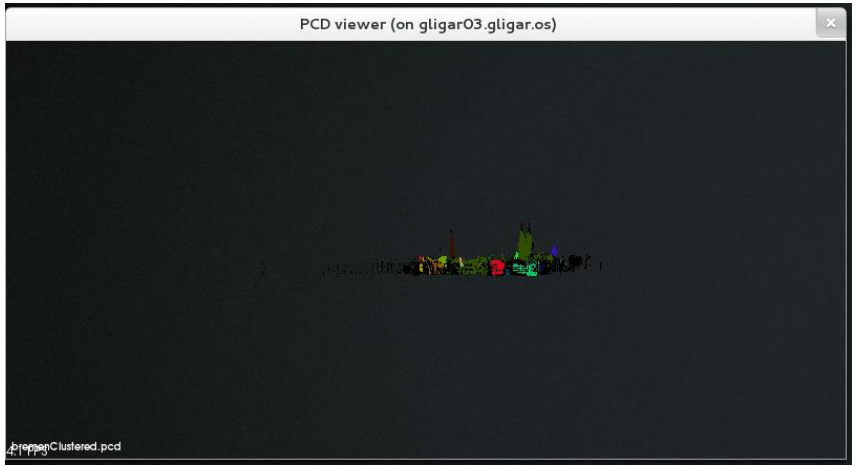
```
import h5py as h5
import numpy as np
import sys

if len(sys.argv) < 2:
    INPUT="bremen.h5"
else:
    INPUT = sys.argv[1]
FILE = "bremenClustered.pcd"

print"loading H5"
bremen = h5.File("bremenSmall.h5")
points = bremen["DBSCAN"]
clusters = bremen["Clusters"]
colors = bremen["COLORS"]

print "Transform to numpy"
points = np.array(points)
clusters = np.array(clusters)
colors = np.array(colors)

#print "Remove Noise"
#points = points[clusters!=0]
#clusters = clusters[clusters!=0]

#data = np.concatenate((points,colors.reshape((-1,1))),axis=1)
data = np.concatenate((points,clusters.reshape((-1,1))),axis=1)

clusters[clusters!=0]=1
data = np.concatenate((data,clusters.reshape((-1,1))),axis=1)

print "Write PCD"
"H5toPCD.py" 49L, 1028C
```



Lecture 4 – Point Cloud Library – Viewer

```
[train001@jrl12 pisvm-1.2.1]$ module spider pcl

-----------------------------------------------------------------------------------------------------------------
  PCL:
-----------------------------------------------------------------------------------------------------------------
    Description:
      The Point Cloud Library (PCL) is a standalone, large scale, open project for 2D/3D image and point cloud processing.

     Versions:
        PCL/1.8.1-Python-2.7.14

-----------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "PCL" module (including how to load the modules) use the module's full name.
  For example:

     $ module spider PCL/1.8.1-Python-2.7.14
-----------------------------------------------------------------------------------------------------------------
```

```
[train001@jrl04 ~]$ module load Intel/2018.0.128-GCC-5.4.0 ParaStationMPI/5.2.0-1
[train001@jrl04 ~]$ module load PCL

[train001@jrl06 ~]$ pcl_viewer
```



Lecture 5 – Supervised Classification & Learning Theory Basics

- 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

Unknown Target Distribution

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

(ideal function)

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

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

(set of candidate formulas)

Final Hypothesis $g \approx f$

'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') $\mathcal{A}$ → 'learn: get error smaller' $E_{in}(g) \approx 0$ 'generalize well for unseen data' $E_{out}(g) \approx E_{in}(g)$

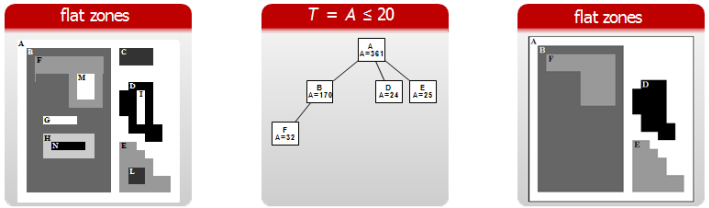
(set of known algorithms)

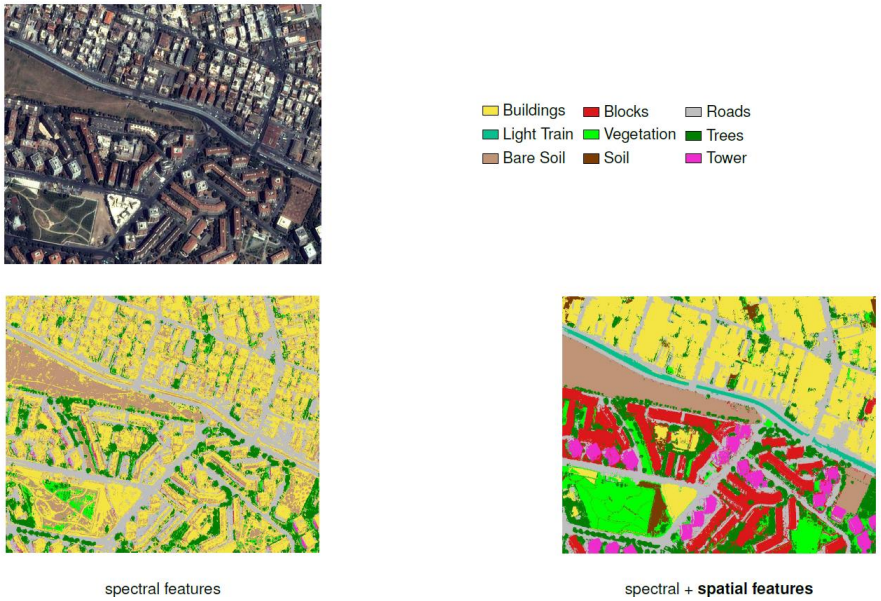
$$\Pr \ [\ | \ E_{in}(g) - E_{out}(g) \ | > \epsilon \ ] \ <= \ 4m_{\mathcal{H}}(2N)e^{-1/8\epsilon^2 N}$$

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

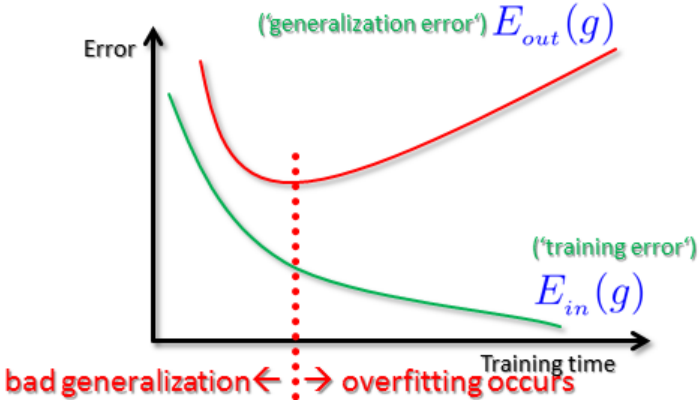
Lecture 6 – Classification Applications, Challenges & Solutions

- Remote Sensing Data
  - Pixel-wise spectral-spatial classifiers
  - Feature enhancements
  - Example of Self-Dual Attribute Profile (SDAP) technique



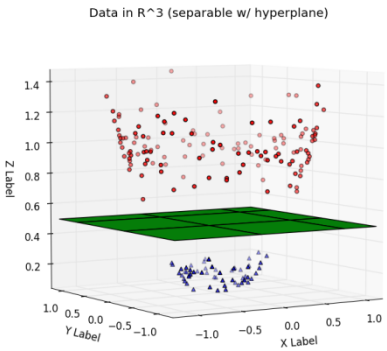


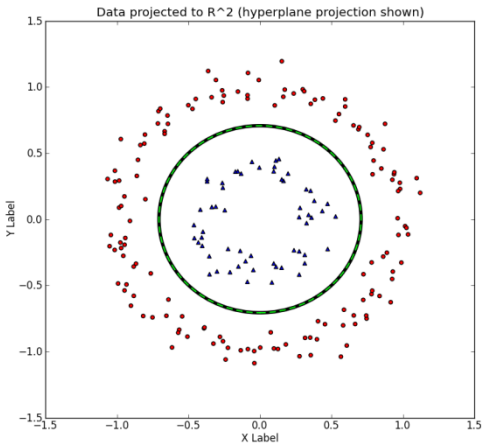
- Classification Challenges
  - Scalability, high dimensionality, etc.
  - Non-linearly seperable data
  - Overfitting as key problem in training machine learning models



Lecture 7 – Support Vector Machines & Kernel Methods

- Non-linear Transformations
  - Use of a mapping function Φ
  - Hyperplane in higher dimensional space possible
  - Mapping back corresponds to non-linear decision boundary in initial input or x space





- Full Support Vector Machine
  - Full = use of non-linear kernel
  - Take advantage of mapping into a higher-level/infinite space
  - Apply 'kernel trick'
  - Kernels quantify similiarity
  - Different trusted kernels available (RBF, polynomial, etc.)

$$\sum \alpha_i y_i \boxed{\mathbf{x}_i \cdot \mathbf{u}_i} + b \geq 0 \quad \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{u}_i)$$

(dual since primal wi and b removed)

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

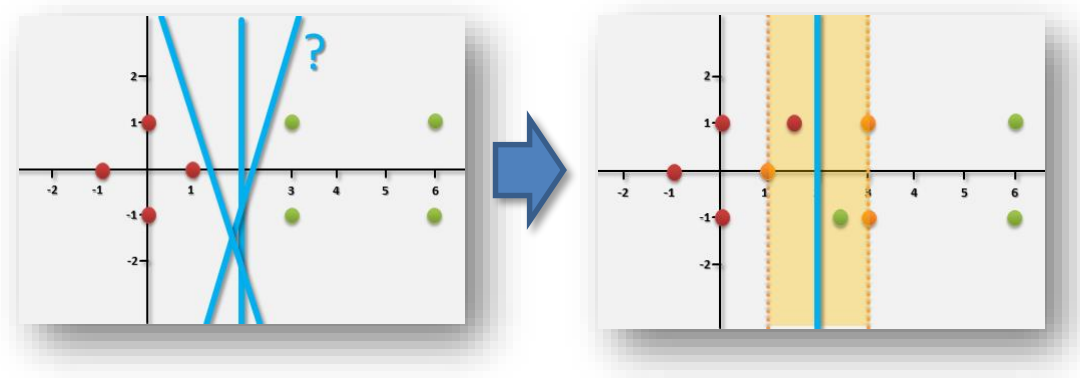
(trusted Kernel avoids to know Phi)

$$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

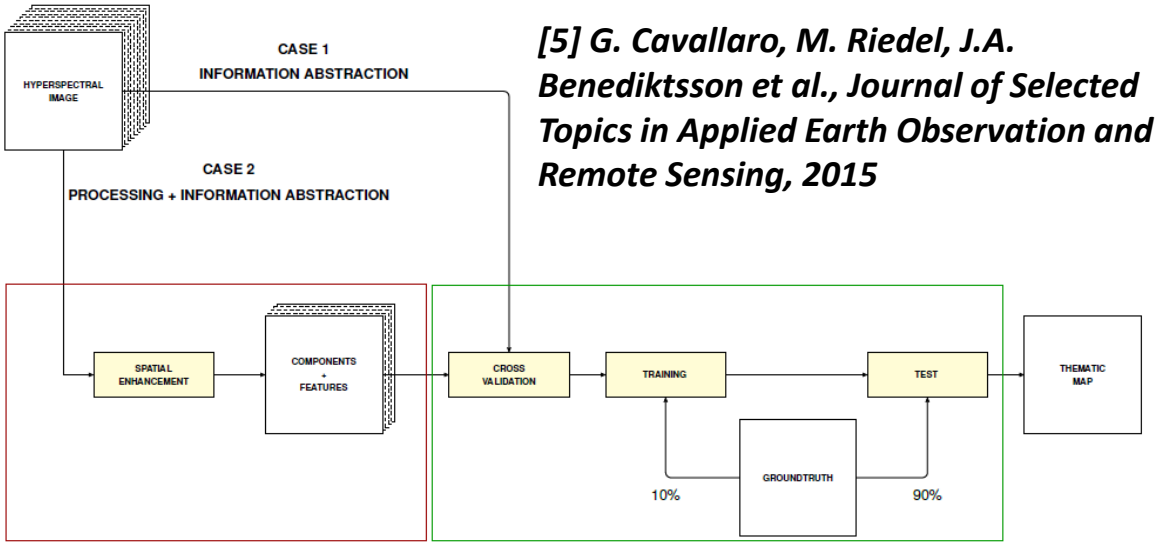
$$\boxed{K(\mathbf{x}_i, \mathbf{x}_j)} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

Lecture 8 – Practicals with SVMs

- Analysing Indian Pines Dataset
  - Remote Sensing Dataset
  - Raw & Processed
  - Using JUROPA3 system with MPI-based piSVM 1.2.1 tool



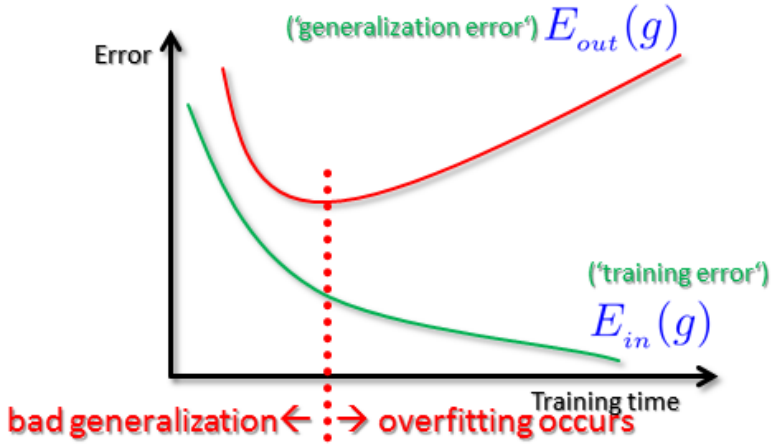
- Feature Engineering in Machine learning
  - Manual feature enhancements increased accuracy from ~44% to ~77%
  - Some work before 'learning from data' is important



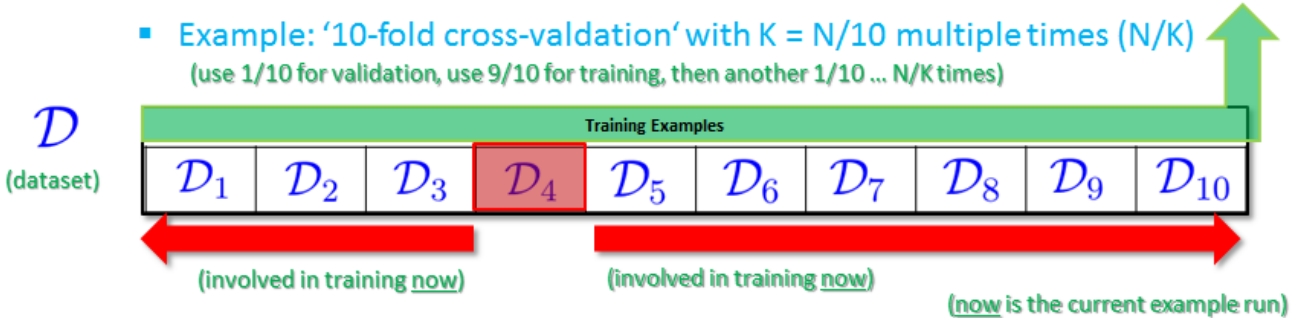
[5] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

Lecture 9 – Validation and Regularization Techniques

- ‘Combat‘ Overfitting in Machine Learning
  - Two key approaches
- Validation
  - 10-fold cross-validation mostly applied in practice
  - Rule of thumb: 1/5 of data
- Regularization
  - Rule of thumb: Simpler model is usually better
  - Regularization approaches often inherent in algorithm / optimization
  - E.g. Parameter C as cost parameter in SVMs to violate the margin



- Example: ‘10-fold cross-valdation‘ with K = N/10 multiple times (N/K)
  (use 1/10 for validation, use 9/10 for training, then another 1/10 ... N/K times)



Lecture 10 – Practicals with Validation and Regularization

- 10-fold Cross-Validation
  - Validation estimates for out of sample performance only on training Dataset is remarkably well
  - Training Dataset is validation Dataset using 10-fold cross-validation
  - Test Dataset kept seperate
- Regularization
  - Parameter C important to steer how much error allowed

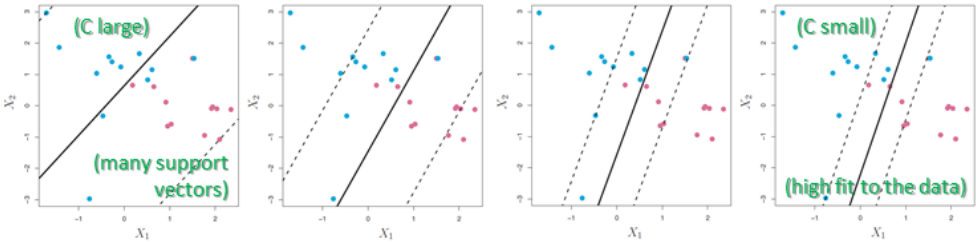
```
#SBATCH--job-name=10x-cv-indianpines-4-96-24

### location executable
PISVM=/homea/hpclab/train001/tools/pisvm-1.2.1/pisvm-train

### location data
TRAINDATA=/homea/hpclab/train001/data/indianpines/indian_processed_training.el

### submit
srun $PISVM -v 10 -D -o 1024 -q 512 -c 100 -g 8 -t 2 -m 1024 -s 0 $TRAINDATA
```

```
[train001@jrl12 pisvm-1.2.1]$ tail mpi-out.4642829
Cross Validation Accuracy = 0.0658762%
I/O time = 0.76
Cross Validation Accuracy = 0.0658762%
I/O time = 0.76
Cross Validation Accuracy = 0.0658762%
I/O time = 0.76
Cross Validation Accuracy = 0.0658762%
I/O time = 0.76
Cross Validation Accuracy = 77.0242%
I/O time = 0.77
```

(cross-validation accuracy on training set)

```
[train001@j3l02 pisvm-1.2.1]$ more mpi-out.15244572
Accuracy = 77.9678% (234336/300555) (classification)
Mean squared error = 153.787 (regression)
Squared correlation coefficient = 0.601418 (regression)
```

(testing on test set – 'simulating out-of-sample')



(fewer support Vectors: just #8)

[1] An Introduction to Statistical Learning

Exercises – Cross-Validation Runs – Evaluate Gridsearch

Parallelization Benefit: Parallel 10-Fold Cross-Validation

- Example: 2 Parameters, 10-fold cross-validation
  - 2 x benefits of parallelization possible in a so-called 'gridsearch'
  - (1) Compute parallel; (2) Do all cross-validation runs in parallel (all cells)
  - Evaluation between Matlab (aka 'serial laptop') & parallel (80 cores)

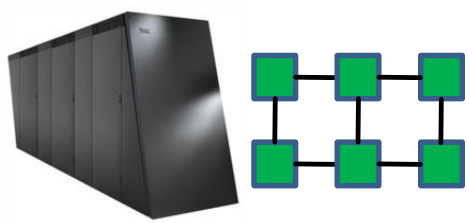
(2) Scenario 'pre-processed data', 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

(1) First Result: best parameter set from 14.41 min to 1.02 min

(2) Second Result: all parameter sets from ~9 hours to ~35 min

[1] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015



'(1) each cell inherent parallel'

'(2) all cells in parallel'

(2) Scenario 'pre-processed data', 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

- **10-fold cross-validation achieves parallelization benefits (1) in each grid cell and (2) across all cells**

Parallelization Summary

- Parallelization benefits are enormous for complex problems
  - Enables feasibility to tackle extremely large datasets & high dimensions
  - Provides functionality for a high number of classes (e.g. #k SVMs)
  - Achieves a massive reduction in time → lower time-to-solution

(1) Scenario 'unprocessed data', 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	27.30 (109.78)	34.59 (124.46)	39.05 (107.85)	37.38 (116.29)	37.20 (121.51)
4	29.24 (98.18)	37.75 (85.31)	38.91 (113.87)	38.36 (119.12)	38.36 (118.98)
8	31.31 (109.95)	**39.68** (118.28)	39.06 (112.99)	39.06 (190.72)	39.06 (872.27)
16	33.37 (126.14)	39.46 (171.11)	39.19 (206.66)	39.19 (181.82)	39.19 (146.98)
32	34.61 (179.04)	38.37 (202.30)	38.37 (231.10)	38.37 (240.36)	38.37 (278.02)

(2) Scenario 'pre-processed data', 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

(1) Scenario 'unprocessed data''10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	27.26 (3.38)	34.49 (3.35)	39.16 (5.35)	37.56 (11.46)	37.57 (13.02)
4	29.12 (3.34)	37.58 (3.38)	38.91 (6.02)	38.43 (7.47)	38.43 (7.47)
8	31.24 (3.38)	**39.77 (4.09)**	39.14 (5.45)	39.14 (5.42)	39.14 (5.43)
16	33.36 (4.09)	39.61 (4.56)	39.25 (5.06)	39.25 (5.27)	39.25 (5.10)
32	34.61 (5.13)	38.37 (5.30)	38.36 (5.43)	38.36 (5.49)	38.36 (5.28)

First Result: best parameter set from 118.28 min to 4.09 min
Second Result: all parameter sets from ~3 days to ~2 hours

(2) Scenario 'pre-processed data', 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

First Result: best parameter set from 14.41 min to 1.02 min
Second Result: all parameter sets from ~9 hours to ~35 min

[1] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

Lecture Bibliography

Lecture Bibliography (1)

- [1] An Introduction to Statistical Learning with Applications in R.
  Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [2] G. Cavallaro, M. Riedel, M. Richerzhagen, J. A. Benediktsson and A. Plaza, "On Understanding Big Data Impacts in Remotely Sensed Image Classification Using Support Vector Machine Methods," *in the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 10, pp. 4634-4646, Oct. 2015.
- [3] SVM and Parameters,
  Online: https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_15.png
- [4] Andreas Töscher and Michael Jahrer, The BigChaos Solution to the Netflix Grand Prize, 2009
- [5] D. Singh and K. Kaur, 'Classification of Abnormalities in Brain MRI Images Using', International Journal of Engineering and Advanced Technology, ISSN: 2249 – 8958, Volume 1, Issue-6, 2012
- [6] Indian Pines Raw and Processed
  Online: http://hdl.handle.net/11304/9ec5eac8-61b4-4617-ae1c-1f8c8cd3cd74
- [7] LLView Tool,
  Online: http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/LLview/_node.html


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