Parallel & Scalable Machine Learning
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 1

Introduction to Machine Learning Fundamentals

March 6th, 2018
JSC, Germany
Outline of the Course

1. Introduction to Machine Learning Fundamentals
   Day One – beginner
2. DEEP – Projects and Parallel Computing Basics
   Day Two – moderate
3. Unsupervised Clustering and Applications
   Day Three – expert
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. Parallelization Benefits
12. Cross-Validation Practicals
Outline

 Machine Learning Basics
   Motivation
   Methods Overview
   Simple Application Example
   Perceptron Learning Model
   Decision Boundary & Linear Separability

 Learning from Data
   Systematic Process to Support Learning
   Predictive and Descriptive Tasks
   Different Learning Approaches
   Terminologies
   Model Evaluation with Testing
Motivation

- Rapid advances in data collection and storage technologies in the last decade
  - Extracting useful information is a challenge considering ever increasing massive datasets
  - Traditional data analysis techniques cannot be used in growing cases (e.g. memory limits)

- Machine learning / Data Mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data
- Machine Learning / Data Mining is the process of automatically discovering useful information in large data repositories ideally following a systematic process

modified from [1] Introduction to Data Mining

- Machine Learning & Statistical Data Mining
  - Traditional statistical approaches are still very useful to consider
  - E.g. in order to reduce large quantities of data to most expressive datasets
Machine Learning Prerequisites

1. Some pattern exists
2. No exact mathematical formula
3. Data exists

- Idea ‘Learning from Data’ shared with a wide variety of other disciplines
  - E.g. signal processing, data mining, etc.
- Challenge: Data is often complex

- Machine learning is a very broad subject and goes from very abstract theory to extreme practice (‘rules of thumb’)
Examples of Real Data Collections

- Data collection of the earth and environmental science domain
  - Different from the known ‘UCI machine learning repository examples’

(see real science datasets examples)

[2] PANGAEA data collection

Methods Overview

- Machine learning 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
Simple Application Example: Classification of a Flower

(1) Problem Understanding Phase

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

(what type of flower is this?)


(flowers of type ‘IRIS Setosa’)
The Learning Problem in the Example

Learning problem: A prediction task
- Determine whether a new Iris flower sample is a “Setosa” or “Virginica”
- Binary (two class) classification problem
- What attributes about the data help?

[Image sources: Species Iris Group of North America Database, www.signa.org]
Feasibility of Machine Learning in this Example

1. Some pattern exists:
   - Believe in a ‘pattern with ‘petal length‘ & ‘petal width‘ somehow influence the type

2. No exact mathematical formula
   - To the best of our knowledge there is no precise formula for this problem

3. Data exists
   - Data collection from UCI Dataset „Iris“
   - 150 labelled samples (aka ‘data points‘)
   - Balanced: 50 samples / class

(2) Data Understanding Phase


(4) Data attributes for each sample in the dataset

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class: Iris Setosa, or Iris Versicolour, or Iris Virginica

Lecture 1 – Introduction to Machine Learning Fundamentals
Demo Exercise – Explore IRIS Dataset
Understanding the Data – Check Metadata

- First: Check metadata if available
  - Example: Downloaded `iris.names` includes metadata about data

1. Title: Iris Plants Database
   Updated Sept 21 by C.Blake - Added discrepancy information

2. Sources:
   - (a) Creator: R.A. Fisher
   - (b) Donor: Michael Marshall (MARSHALL@FLU@io.arc.nasa.gov)
   - (c) Date: July, 1988

3. ... (metadata is not always available in practice)

4. Number of Instances: 150 (50 in each of three classes)

5. Number of Attributes: 4 numeric, predictive attributes and the class

6. Attribute Information:
   1. sepal length in cm
   2. sepal width in cm
   3. petal length in cm
   4. petal width in cm
   5. class:
      -- Iris Setosa
      -- Iris Versicolour
      -- Iris Virginica


Lecture 1 – Introduction to Machine Learning Fundamentals 14 / 59
Understanding the Data – Check Table View

- Second: Check **table view** of the dataset with some samples
  - E.g. Using a GUI like ‘Rattle’ (library of R), or Excel in Windows, etc.
  - E.g. Check the first row if there is **header information** or if is a sample

![Image of Rattle GUI]

- (careful first sample taken as header, resulting in only 149 data samples)
- (four data attributes for each sample in the dataset)
- (one class label for each sample in the dataset)

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class: Iris Setosa, or Iris Versicolour, or Iris Virginica

[7] Rattle Library for R
Preparing the Data – Corrected Header

(3) Data Preparation Phase

(correct header information, resulting in 150 data samples)

(correcting the header is not always necessary, or can be automated, e.g. in Rattle)
Preparing the Data – Remove Third Class Samples

- Data preparation means to prepare our data for our problem
  - In practice the whole dataset is rarely needed to solve one problem
  - E.g. apply several sampling strategies (but be aware of class balance)

- Recall: Our learning problem
  - Determine whether a new Iris flower sample is a “Setosa” or “Virginica”
  - Binary (two class) classification problem: ‘Setosa’ or ‘Virginica’

- (three class problem with N = 150 samples including Iris Versicolour)
- (remove Versicolour class samples from dataset)
- (wo class problem with N = 100 samples excluding Iris Versicolour)
- (export or save dataset to iris-twoclass.data)
Preparing the Data – Feature Selection Process

- Data preparation means to prepare our data for our problem
  - In practice the whole dataset is rarely needed to solve one problem
  - E.g. perform feature selection (aka remove not needed attributes)
- Recall: Our believed pattern in the data
  - A ‘pattern with ‘petal length’ & ‘petal width’ somehow influence the type

(N = 100 samples with 4 attributes and 1 class label)

(Lecture 1 – Introduction to Machine Learning Fundamentals)

(N = 100 samples with 2 attributes and 1 class label)
Demo Exercises – Explore Iris Dataset Features & Classes
Different samples of the original Iris dataset

- Created for **linear separability** and **non-linear separability**
Check Preparation Phase: Plotting the Data

Dataset

(attributes with d=2)

\(X = x_1, \ldots, x_d\)

(x1 is petal length, x2 is petal width)

(Recall: we believed in a ‘pattern’ with ‘petal length’ & ‘petal width’ somehow influence the flower type)

\((N = 100 \text{ samples})\)

(what about the class labels?)

petal length (in cm)

petal width (in cm)
Check Preparation Phase: Class Labels

\[ \mathbf{x} = x_1, \ldots, x_d \]

\[ y_i, \ i = 1, \ldots, n \]

(N = 100 samples)

(still no machine learning so far)
The data is linearly separable (rarely in practice)

A line becomes a decision boundary to determine if a new data point is class red/green.

\[(x_1, y_1), \ldots, (x_N, y_N)\]  
\[(N = 100 \text{ samples})\]

Iris-setosa

Iris-virginica
Separating Line & Mathematical Notation

- **Data exploration results**
  - A line can be crafted between the classes since linearly separable data
  - All the data points representing Iris-setosa will be below the line
  - All the data points representing Iris-virginica will be above the line

- **More formal mathematical notation**
  - Input: \( \mathbf{x} = x_1, \ldots, x_d \) (attributes of flowers)
  - Output: class +1 (Iris-virginica) or class -1 (Iris-setosa)

\[
\text{Iris-virginica if } \sum_{i=1}^{d} w_i x_i > \text{threshold} \\
\text{Iris-setosa if } \sum_{i=1}^{d} w_i x_i < \text{threshold}
\]

\[
\text{(compact notation)} \quad \text{sign} \left( \left( \sum_{i=1}^{d} w_i x_i \right) - \text{threshold} \right)
\]

(w_i and threshold are still unknown to us)
Separating Line & ‘Decision Space’ Example

 modified from [13] An Introduction to Statistical Learning
A Simple Linear Learning Model – The Perceptron

- Human analogy in learning
  - Human brain consists of nerve cells called neurons
  - Human brain learns by changing the strength of neuron connections \( w_i \) upon repeated stimulation by the same impulse (aka a ‘training phase’)
  - Training a perceptron model means adapting the weights \( w_i \)
  - Done until they fit input-output relationships of the given ‘training data’

\[
\text{sign} \left( \sum_{i=1}^{d} w_i x_i - \text{threshold} \right)
\]

(activation function, +1 or -1)

\[ (x_1, y_1), \ldots, (x_N, y_N) \]

(training data)

\[ \sum w_i x_i \]

(threshold)

\( w_1, w_2, w_3 \)

(input nodes)

\( X_0 \) (bias)

(output node)

\( d \)

(dimension of features)

\( y \)

(representing the threshold)
Perceptron – Example of a Boolean Function

\[(x_1, y_1), \ldots, (x_N, y_N)\]
(training data)

(training phase)

\[\text{sign}\left(\sum_{i=1}^{d} w_i x_i - \text{threshold}\right)\]

(trained perceptron model)

**Output node interpretation**

- More than just the weighted sum of the inputs – threshold (aka bias)
- Activation function \(\text{sign}\) (weighted sum): takes sign of the resulting sum

\[y = 1, \text{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0\]
(e.g. consider sample #3, sum is positive (0.2) \(\rightarrow\) +1)

\[y = -1, \text{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0\]
(e.g. consider sample #6, sum is negative (-0.1) \(\rightarrow\) -1)
Summary Perceptron & Hypothesis Set \( h(x) \)

- **When:** Solving a **linear classification problem**
  - Goal: learn a simple value (+1/-1) above/below a certain threshold
  - Class label renamed: Iris-setosa = -1 and Iris-virginica = +1
- **Input:** \( \mathbf{X} = x_1, \ldots, x_d \) (attributes in one dataset)

- **Linear formula** (take attributes and give them different weights – think of ‘impact of the attribute’)
  - All learned formulas are different hypothesis for the given problem

\[
h(x) = \text{sign} \left( \left( \sum_{i=1}^{d} w_i x_i \right) - \text{threshold} \right); h \in \mathcal{H}
\]

(parameters that define one hypothesis vs. another)

(each green space and blue space are regions of the same class label determined by sign function)

(red parameters correspond to the redline in graphics)

(but question remains: how do we actually learn \( w_i \) and threshold?)

[8] F. Rosenblatt, 1957
Perceptron Learning Algorithm – Understanding Vector $W$

- **When:** If we believe there is a linear pattern to be detected
  - **Assumption:** Linearly separable data (lets the algorithm converge)
  - **Decision boundary:** perpendicular vector $w_i$ fixes orientation of the line

\[
\begin{align*}
  w^T x &= 0 \\
  w \cdot x &= 0
\end{align*}
\]
(points on the decision boundary satisfy this equation)

- Possible via simplifications since we also need to learn the threshold:

\[
 h(x) = \text{sign}(w^T x) = \text{sign}(\sum_{i=1}^{d} w_i x_i + w_0) ; w_0 = -\text{threshold}
\]


\[
 h(x) = \text{sign}\left(\sum_{i=1}^{d} w_i x_i\right) ; x_0 = 1
\]

(All notations are equivalent and result is a scalar from which we derive the sign)
Understanding the Dot Product – Example & Interpretation

- **‘Dot product’**
  - Given two vectors
  - Multiplying corresponding components of the vector
  - Then adding the resulting products
  - Simple example: \((2, 3) \cdot (4, 1) = 2 \times 4 + 3 \times 1 = 11\) (a scalar!)
  - Interesting: Dot product of two vectors is a scalar

- **‘Projection capabilities of Dot product’ (simplified)**
  - Orthogonal projection of vector \(\mathbf{v}\) in the direction of vector \(\mathbf{u}\)
  
  \[
  \mathbf{u} \cdot \mathbf{v} = (\|\mathbf{v}\| \cos(\alpha)) \|\mathbf{u}\| = \mathbf{v}_u \|\mathbf{u}\|
  \]
  - Normalize using length of vector
  
  \[
  \frac{\mathbf{u}}{\|\mathbf{u}\|} \|\mathbf{u}\| = \text{length}(\mathbf{u}) = L_2\text{norm} = \sqrt{\mathbf{u} \cdot \mathbf{u}}
  \]
Perceptron Learning Algorithm – Learning Step

- Iterative Method using (labelled) training data \((x_1, y_1), \ldots, (x_N, y_N)\)
  (one point at a time is picked)

1. Pick one misclassified training point where:
   \[ \text{sign}(w^T x_n) \neq y_n \]

2. Update the weight vector:
   \[ w \leftarrow w + y_n x_n \]
   (\(y_n\) is either +1 or -1)

- Terminates when there are no misclassified points
  (converges only with linearly seperable data)
[Video] Perceptron Learning Algorithm
**Systematic Process to Support Learning From Data**

- **Systematic data analysis guided by a ‘standard process’**
  - Cross-Industry Standard Process for Data Mining (CRISP-DM)

- **Lessons Learned from Practice**
  - Go back and forth between the different six phases

---

- A data mining project is guided by these six phases: (1) Problem Understanding; (2) Data Understanding; (3) Data Preparation; (4) Modeling; (5) Evaluation; (6) Deployment


---

**A more detailed description of all six CRISP-DM phases is in the appendix of the slideset**
Machine Learning & Data Mining Tasks in Applications

- **Predictive Tasks**
  - Predicts the value of an attribute based on values of other attributes
  - **Target/dependent variable**: attribute to be predicted
  - **Explanatory/independent variables**: attributed used for making predictions
  - E.g. predicting the species of a flower based on characteristics of a flower

- **Descriptive Tasks**
  - Derive patterns that summarize the underlying relationships in the data
  - Patterns here can refer to **correlations, trends, trajectories, anomalies**
  - Often exploratory in nature and frequently require postprocessing
  - E.g. credit card fraud detection with unusual transactions for owners
Predicting Task: Obtain Class of a new Flower ‘Data Point’

(4) Modelling Phase

\[(x_1, y_1), \ldots, (x_N, y_N)\]

\(N = 100\) samples

What means Learning?

- The basic meaning of learning is ‘to use a set of observations to uncover an underlying process’
- The three different learning approaches are supervised, unsupervised, and reinforcement learning

- **Supervised Learning**
  - Majority of methods follow this approach in this course
  - Example: credit card approval based on previous customer applications

- **Unsupervised Learning**
  - Often applied before other learning → higher level data representation
  - Example: Coin recognition in vending machine based on weight and size

- **Reinforcement Learning**
  - Typical ‘human way’ of learning
  - Example: Toddler tries to touch a hot cup of tea (again and again)
Learning Approaches – Supervised Learning

- Each observation of the predictor measurement(s) has an associated response measurement:
  - Input \( x = x_1, \ldots, x_d \)
  - Output \( y_i, i = 1, \ldots, n \)
  - Data \( (x_1, y_1), \ldots, (x_N, y_N) \)

- Goal: Fit a model that relates the response to the predictors
  - Prediction: Aims of accurately predicting the response for future observations
  - Inference: Aims to better understanding the relationship between the response and the predictors

- Supervised learning approaches fits a model that related the response to the predictors
- Supervised learning approaches are used in classification algorithms such as SVMs
- Supervised learning works with data = [input, correct output]
Learning Approaches – Supervised Learning Example

The labels guide our learning process like a ‘supervisor’ is helping us.

Lecture 5 provides more details on the supervised learning approach using classification methods.
Learning Approaches – Unsupervised Learning

- Each observation of the predictor measurement(s) has no associated response measurement:
  - Input \( x = x_1, \ldots, x_d \)
  - No output
  - Data \( (x_1), \ldots, (x_N) \)
- Goal: Seek to understand relationships between the observations
  - Clustering analysis: check whether the observations fall into distinct groups
- Challenges
  - No response/output that could supervise our data analysis
  - Clustering groups that overlap might be hardly recognized as distinct group

Unsupervised learning approaches seek to understand relationships between the observations
Unsupervised learning approaches are used in clustering algorithms such as k-means, etc.
Unsupervised learning works with data = [input, ---]

[13] An Introduction to Statistical Learning
Learning Approaches – Unsupervised Learning Example

- Practice: The number of clusters can be ambiguities

\[ \text{[13] An Introduction to Statistical Learning} \]

- Lecture 3 offers more details about unsupervised learning using clustering algorithms in practice
Learning Approaches – Reinforcement Learning

- Each observation of the predictor measurement(s) has **some associated response measurement**:
  - Input: \( x = x_1, \ldots, x_d \)
  - Some output & grade of the output
  - Data: \( (x_1), \ldots, (x_N) \)

- Goal: Learn through iterations
  - *Guided by output grade*: check learning and compare with grade

- Challenge:
  - Iterations may require lots of CPU time (e.g. backgammon playing rounds)

- (Rarely tackled in this course, just for the sake of completion)

- Reinforcement learning approaches learn through iterations using the grading output as guide
- Reinforcement learning approaches are used in playing game algorithms (e.g. backgammon)
- Unsupervised learning works with data = [input, some output, grade for this output]
Summary Terminologies & Different Dataset Elements

- **Target Function** $f : X \to Y$
  - Ideal function that ‘explains’ the data we want to learn

- **Labelled Dataset (samples)**
  - ‘in-sample’ data given to us: $(x_1, y_1), \ldots, (x_N, y_N)$

- **Learning vs. Memorizing**
  - The goal is to create a system that works well ‘out of sample’
  - In other words we want to classify ‘future data’ (out of sample) correct

- **Dataset Part One: Training set**
  - Used for training a machine learning algorithms
  - Result after using a training set: a trained system

- **Dataset Part Two: Test set**
  - Used for testing whether the trained system might work well
  - Result after using a test set: accuracy of the trained model
Model Evaluation – Training and Testing Phases

- Different Phases in Learning
  - **Training** phase is a hypothesis search
  - **Testing** phase checks if we are on right track (once the hypothesis clear)

- Work on ‘training examples’
  - Create **two disjoint datasets**
    - One used for training only (aka training set)
    - Another used for testing only (aka test set)
  - Exact separation is **rule of thumb per use case** (e.g. 10% training, 90% test)
  - Practice: If you get a dataset take immediately test data away (‘throw it into the corner and forget about it during modelling’)
  - Reasoning: Once we learned from training data it has an ‘optimistic bias’

(4) Modelling Phase
(5) Evaluation Phase

(e.g. student exam training on examples to get $E_{in}$, then test via exam)
Model Evaluation – Testing Phase & Confusion Matrix

- Model is fixed
  - Model is just used with the testset
  - Parameter $w_i$ are set and we have a linear decision function

- Evaluation of model performance
  - Counts of test records that are incorrectly predicted $sign(w^T x_n) \neq y_n$
  - Counts of test records that are correctly predicted $sign(w^T x_n) = y_n$
  - E.g. create confusion matrix for a two class problem

<table>
<thead>
<tr>
<th>Counting per sample</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Class</td>
<td>Class = 1</td>
</tr>
<tr>
<td>Class = 1</td>
<td>$f_{11}$</td>
</tr>
<tr>
<td>Class = 0</td>
<td>$f_{01}$</td>
</tr>
</tbody>
</table>

(serves as a basis for further performance metrics usually used)
Model Evaluation – Testing Phase & Performance Metrics

### Accuracy (usually in %)

Accuracy = \[
\frac{\text{number of correct predictions}}{\text{total number of predictions}}\]

### Error rate

Error rate = \[
\frac{\text{number of wrong predictions}}{\text{total number of predictions}}\]

### If model evaluation is satisfactory:

(6) Deployment Phase

(5) Evaluation Phase

<table>
<thead>
<tr>
<th>Counting per sample</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class = 1</td>
</tr>
<tr>
<td>Actual Class</td>
<td>Class = 1</td>
</tr>
<tr>
<td></td>
<td>Class = 0</td>
</tr>
</tbody>
</table>

(100% accuracy in learning often points to problems using machine learning methods in practice)
[Video] European Plate Observing System

[14] EPOS Data Community Services, YouTube
Lecture Bibliography (1)

- [2] PANGAEA Data Collection, Data Publisher for Earth & Environmental Science, Online: http://www.pangaea.de/
- [10] PLA Algorithm, YouTube Video, Online:
Lecture Bibliography (2)

- [13] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [14] EPOS - European Plate Observing System -- Community Services, YouTube Video, Online: http://www.youtube.com/watch?v=zh-paxiQhKI
Appendix
Summary: Systematic Process

- Systematic data analysis guided by a ‘standard process’
  - Cross-Industry Standard Process for Data Mining (CRISP-DM)

- A data mining project is guided by these six phases:
  1. Problem Understanding;
  2. Data Understanding;
  3. Data Preparation;
  4. Modeling;
  5. Evaluation;
  6. Deployment

- Lessons Learned from Practice
  - Go back and forth between the different six phases

1 – Problem (Business) Understanding

The Business Understanding phase consists of four distinct tasks: (A) Determine Business Objectives; (B) Situation Assessment; (C) Determine Data Mining Goal; (D) Produce Project Plan

- **Task A – Determine Business Objectives**
  - Background, Business Objectives, Business Success Criteria

- **Task B – Situation Assessment**
  - Inventory of Resources, Requirements, Assumptions, and Constraints
  - Risks and Contingencies, Terminology, Costs & Benefits

- **Task C – Determine Data Mining Goal**
  - Data Mining Goals and Success Criteria

- **Task D – Produce Project Plan**
  - Project Plan
  - Initial Assessment of Tools & Techniques

2 – Data Understanding

- The Data Understanding phase consists of four distinct tasks:
  (A) Collect Initial Data; (B) Describe Data; (C) Explore Data; (D) Verify Data Quality

- Task A – Collect Initial Data
  - Initial Data Collection Report

- Task B – Describe Data
  - Data Description Report

- Task C – Explore Data
  - Data Exploration Report

- Task D – Verify Data Quality
  - Data Quality Report
3 – Data Preparation

- Task A – Data Set
  - Data set description

- Task B – Select Data
  - Rationale for inclusion / exclusion

- Task C – Clean Data
  - Data cleaning report

- Task D – Construct Data
  - Derived attributes, generated records

- Task E – Integrate Data
  - Merged data

- Task F – Format Data
  - Reformatted data

The Data Preparation phase consists of six distinct tasks: (A) Data Set; (B) Select Data; (C) Clean Data; (D) Construct Data; (E) Integrate Data; (F) Format Data

4 – Modeling

- The Data Preparation phase consists of four distinct tasks: (A) Select Modeling Technique; (B) Generate Test Design; (C) Build Model; (D) Assess Model;

- Task A – Select Modeling Technique
  - Modeling assumption, modeling technique

- Task B – Generate Test Design
  - Test design

- Task C – Build Model
  - Parameter settings, models, model description

- Task D – Assess Model
  - Model assessment, revised parameter settings
5 – Evaluation

- The Data Preparation phase consists of three distinct tasks: (A) Evaluate Results; (B) Review Process; (C) Determine Next Steps

- Task A – Evaluate Results
  - Assessment of data mining results w.r.t. business success criteria
  - List approved models

- Task B – Review Process
  - Review of Process

- Task C – Determine Next Steps
  - List of possible actions, decision
6 – Deployment

- Task A – Plan Deployment
  - Establish a deployment plan
- Task B – Plan Monitoring and Maintenance
  - Create a monitoring and maintenance plan
- Task C – Product Final Report
  - Create final report and provide final presentation
- Task D – Review Project
  - Document experience, provide documentation

The Data Preparation phase consists of three distinct tasks: (A) Plan Deployment; (B) Plan Monitoring and Maintenance; (C) Produce Final Report; (D) Review Project

Slides Available at http://www.morrisriedel.de/talks