Machine Learning Methods

Beginner's Workshop Machine Learning

TU Berlin – SoSe 2019

Scope of this lecture

After this lecture you should:

- have an overview of Machine Learning.
- know some of the basic concepts of Machine Learning.
- understand how the typical ML system and design workflow look like.
- know how to assess and select ML models.
- beware of issues arising in production systems.

Part I: Artificial Intelligence and Machine Learning

What is Artificial Intelligence?

Term coined 1956 at Dartmouth workshop:

- "The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

Field of "Machine Intelligence" already existed before:

- Turing test: **test if human and machine are indistinguishable.**
  - "Computing Machinery and Intelligence", Alan Turing, 1950

What is Artificial Intelligence?

Artificial intelligence is typically “measured” by human-like behavior:

- **Thinking/acting humanly or thinking/acting rationally** (e.g., in the sense of logic, or maximizing reward)
  - “Artificial Intelligence - A modern approach”, Russell and Norvig

The actual human perception of AI is shifting: who considers a calculator or Deep Blue still as intelligent?

What is Artificial Intelligence?

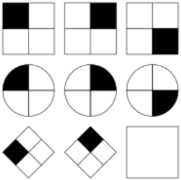
Practically AI-science is engaged with solving problems like:

- **Searching:** for a good algorithm/agent behavior, e.g., evolutionary algorithms.
- **Reasoning:** making statements given assumptions, rules, e.g., inference engines, automated theorem proving.
- **Planning:** how to use resources best, e.g., constraint satisfaction problems.
- **Knowledge representations:** symbolic (e.g., rules, graphs) vs non-/sub-symbolic ones (e.g., tensors).
- **Learning:** rest of the lecture!

Mind: learning is not necessarily part of an AI-system, e.g., rule based reasoning.

What is Machine Learning?

Definition of learning: **“The acquisition of knowledge or skills through study, experience, or being taught.”** (Oxford dict.)
In data-driven contexts often called: “Pattern Recognition”
What does it mean to have learned/recognized a “pattern”?



Credit: https://commons.wikimedia.org/wiki/User:Life_of_Riley

Learning/recognizing is different from memorizing: **need to “generalize” to novel settings!**

ML applications

An incomplete overview.

Traditional:

- **Natural Language Processing (NLP):** speech recognition, translation, speech synthesis.
- **Computer vision:** medical imaging, face recognition.
- **Recommendation:** advertisement, products.
- **Outlier detection:** intrusion detection.

Emerging ones:

- **Data generation:** generating audio, pictures.
- **Data structures:** replacing heuristics, i.e., binary trees.
- **Knowledge discovery:** learning patterns and exploiting it, e.g., Google Go.
- **Optimizing:** meta-learning, optimizing ML, device placement.
- **Planning actions:** navigating.
- **Sciences:** circumventing long equations in physics.

Characterization of Machine Learning

Also called **“Statistical Learning”**.

- use of mathematics and statistics to learn from observations.

Now we characterize the state-of-the-art Machine Learning approach.
The rest of the workshop will treat the content more in depth.

Decision theory and generalization

The problems we want to solve can typically be casted into **making a decision**:

- choosing a class, estimating a value, deciding which action to perform next.
- we want to make the right decision!
- decision theory is about how to make the "best" decisions.

In ML this means after learning the machine should decide well: generalization to novel observations!

Learning settings

Learning itself happens in three major forms:

- **Supervised learning:** teacher provides for each observation the correct decision.
  - E.g., a category label or signal.
- **Unsupervised learning:** no teacher, there are only observations and the algorithm learns something "predefined".
  - E.g., clustering similar observations.
- **Reinforcement learning:** teacher provides for a proposed decision only if it was wrong or right.
  - In supervised the solution is provided.
  - E.g., predicted right class label, agent won a game or not.

Evaluation

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Credit: A Few Useful Things to Know about Machine Learning, Domingos

One of the main tasks in ML is to select good learners!
This is done by a function that measures the generalization performance.

- The criteria depends on the initial problem only! E.g., for classification: accuracy.

How to measure the generalization or future performance?

Representation of knowledge

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Credit: A Few Useful Things to Know about Machine Learning, Domingos

Need a way to represent pre-knowledge, knowledge during and after learning!

Optimization

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Credit: A Few Useful Things to Know about Machine Learning, Domingos

The final step is to optimize the machine knowledge to maximize the evaluation performance!

- **This is the learning part.**
- One can optimize model parameter and hyper-parameter (e.g., for regularization and optimization techniques).

Challenges in Machine Learning

Some of the major challenges in ML are:

- Transfer of knowledge into similar/new settings.
- Extracting semantics out of knowledge, making learned knowledge accessible to humans.
- Compared to humans, machines are very slow at learning (need millions of iterations).
- Understanding complex learning machines like NN theoretically and in action.
- Generating complex and semantic data.

Part II: Machine Learning Theory

Formalizing machine learning

The model of learning from examples can be described using three components:

1) a generator of random vectors x, drawn independently from a fixed but unknown distribution $P(x)$;
2) a supervisor that returns an output vector y for every input vector x, according to a conditional distribution function[1] $P(y|x)$, also fixed but unknown;
3) a learning machine capable of implementing a set of functions $f(x, \alpha), \alpha \in \Lambda$.

Source: Vapnik. An Overview of Statistical Learning Theory (1999)

Formalizing machine learning

The problem of learning is that of choosing from the given set of functions $f(x, \alpha), \alpha \in \Lambda$, the one which predicts the supervisor's response in the best possible way. The selection is based on a training set of ℓ random independent identically distributed (i.i.d.) observations drawn according to $P(x, y) = P(x)P(y|x)$

$$(x_1, y_1), \cdots, (x_\ell, y_\ell). \tag{1}$$

Source: Vapnik. An Overview of Statistical Learning Theory (1999)

Formalizing machine learning

B. Problem of Risk Minimization

In order to choose the best available approximation to the supervisor's response, one measures the *loss* or discrepancy $L(y, f(x, \alpha))$ between the response y of the supervisor to a given input x and the response $f(x, \alpha)$ provided by the learning machine. Consider the expected value of the loss, given by the *risk functional*

$$R(\alpha) = \int L(y, f(x, \alpha))\, dP(x, y). \tag{2}$$

The goal is to find the function $f(x, \alpha_0)$ which minimizes the risk functional $R(\alpha)$ (over the class of functions $f(x, \alpha), \alpha \in \Lambda$) in the situation where the joint probability distribution $P(x, y)$ is unknown and the only available information is contained in the training set (1).

Source: Vapnik. An Overview of Statistical Learning Theory (1999)

Formalizing machine learning

The Problem of Pattern Recognition: Let the supervisor's output y take on only two values $y = \{0, 1\}$ and let $f(x, \alpha), \alpha \in \Lambda$, be a set of *indicator* functions (functions which take on only two values zero and one). Consider the following loss-function:

$$L(y, f(x, \alpha)) = \begin{cases} 0 & \text{if } y = f(x, \alpha) \\ 1 & \text{if } y \neq f(x, \alpha). \end{cases} \tag{3}$$

For this loss function, the functional (2) provides the probability of classification error (i.e., when the answers y given by supervisor and the answers given by indicator function $f(x, \alpha)$ differ). The problem, therefore, is to find the function which minimizes the probability of classification errors when probability measure $P(x, y)$ is unknown, but the data (1) are given.

Source: Vapnik. An Overview of Statistical Learning Theory (1999)

Formalizing machine learning

The General Setting of the Learning Problem: The general setting of the learning problem can be described as follows. Let the probability measure $P(z)$ be defined on the space Z. Consider the set of functions $Q(z, \alpha), \alpha \in \Lambda$. The goal is: to minimize the risk functional

$$R(\alpha) = \int Q(z, \alpha) \, dP(z), \qquad \alpha \in \Lambda \tag{6}$$

if probability measure $P(z)$ is unknown but an i.i.d. sample

$$z_1, \cdots, z_\ell \tag{7}$$

is given.

Source: Vapnik. An Overview of Statistical Learning Theory (1999)

Formalizing machine learning

D. Empirical Risk Minimization Induction Principle

In order to minimize the risk functional (6), for an unknown probability measure $P(z)$ the following induction principle is usually used.

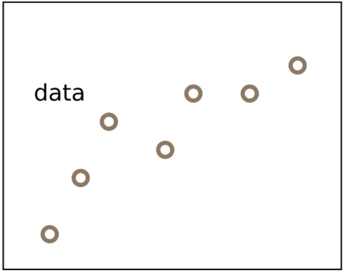
The expected risk functional $R(\alpha)$ is replaced by the *empirical risk* functional

$$R_{\mathrm{emp}}(\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} Q(z, \alpha) \tag{8}$$

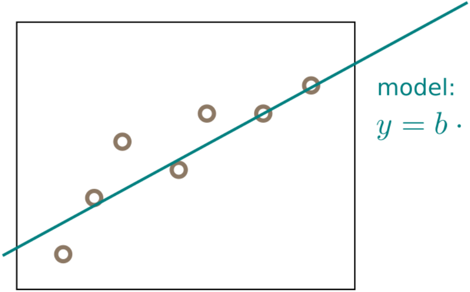
constructed on the basis of the training set (7).

Source: Vapnik. An Overview of Statistical Learning Theory (1999)

Example: Predicting the Future



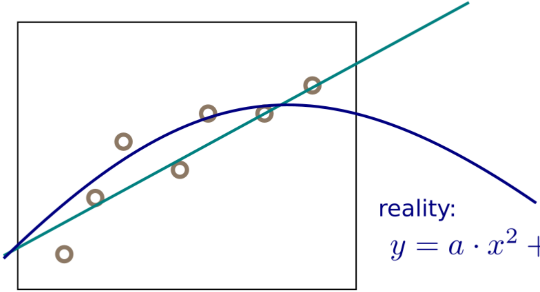
Example: Predicting the Future



model:

$$y = b \cdot x + c$$

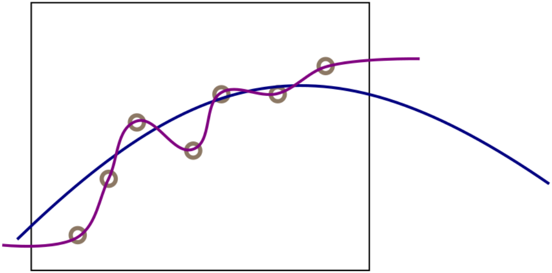
Example: Predicting the Future



reality:

$$y = a \cdot x^2 + b \cdot x + c$$

Example: Predicting the Future



more complex model:

$$y = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0$$

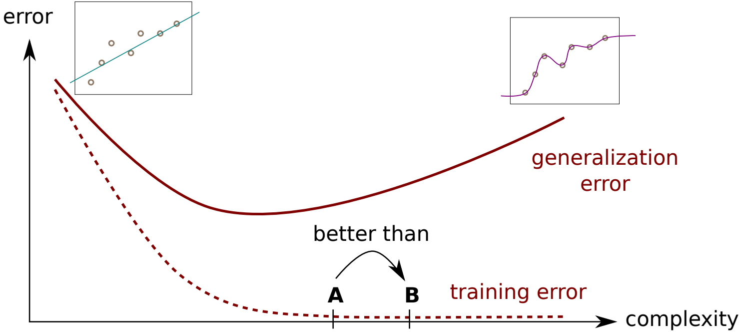
Occam's Razor

"Among competing hypotheses, the one with the fewest assumptions should be selected."

Domingos (1998): "Occam's two Razors: The Sharp and the Blunt" lists **two common interpretations** of it in a ML setting:

- **1st Razor:** *"Given two models with the same generalization error, the simpler one should be preferred because simplicity is desirable in itself."*
- **2nd Razor:** *"Given two models with the same training-set error the simpler one should be preferred because it is likely to have lower generalization error."*

Occam's (2nd) Razor in ML



- A too simple model causes "underfitting".
- A too complex model causes "overfitting".
- **Question:** How to define model complexity?

Possible Measures of Complexity

Measure 1: Number of parameters of the model

- $f(\boldsymbol{x}) = c$ has 1 parameters.
- $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + c$ has $(d+1)$ parameters.
- $f(\boldsymbol{x}) = \boldsymbol{x}^\top A \boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{x} + c$ has (d^2+d+1) parameters.

Measure 2: Number of variables the model receives as input

- Feature selection.
- PCA dimensionality reduction.

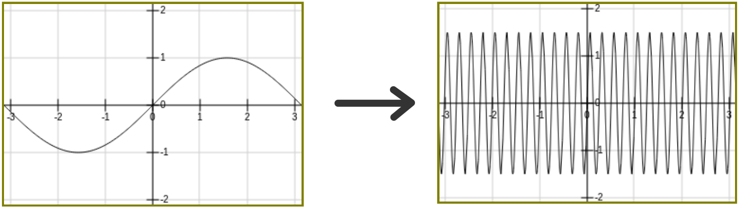
Measure 3: Number of variations in function

- Function's continuity and slope.

A Second Look at Occam's Razor

> *"Given two models with the same training-set error the simpler one should be preferred because it is likely to have lower generalization error."*

Counter-example for "simplicity = few parameters": The two-parameters model $f(x) = a\sin(\omega x)$ can fit almost *any* dataset in $\mathbb{R}$.



I.e. $(a = 1, \omega = 21833.5)$ is "simple" (only 2 numbers), but does not lead to low generalization error.

Approach 1: Bound on Generalization Error

Generalization bound [Vapnik]:

Let h denote the VC-dimension of $\mathcal{F}$. The true risk $R[f]$ (with $f \in \mathcal{F}$) is upper-bounded as:

$$R[f] \leq R_{\text{emp}}[f] + \sqrt{\frac{h\left(\log \frac{2N}{h} + 1\right) - \log(\eta/4)}{N}}$$

with probability $1 - \eta$.

Interpretation:

- Error increases with the VC-dimension h (for h small enough).
- Error decreases with the number of samples N.

Remark:

- If the VC-dimension is infinite, the empirical error does not converge to $R[f]$ even for $N \to \infty$ (see $\sin(\alpha x)$ example).

Approach 1: Bound on Generalization Error

VC-dimension: Intuitive definition

The VC-dimension is the *maximum* number of data points that the function class can *always* shatter (i.e. classify in *any* possible ways).

Examples:

- VC-dimension of $f : \mathbb{R} \to \{-1, 1\},\ f(x) = \mathrm{sign}(\sin(\alpha x))$?
  Answer: ∞

- VC-dimension of $f : \mathbb{R}^d \to \{-1, 1\},\ f(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$?
  Answer: $d + 1$ (also related to the number of parameters).

- VC-dimension of $f : \mathbb{R}^d \to \{-1, 1\},\ f(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$, where all data points are in a minimum enclosing sphere of radius R, and classified with some margin M ?
  Answer: $\min \left\{ d + 1\, ,\, 4\frac{R^2}{M^2} + 1 \right\}$.
  $\Rightarrow$ Does not only depend on input dimension.
  $\Rightarrow$ Large margin lowers model complexity.

Approach 2: From Occam's Razor to Popper's

> *"Given two models with the same training-set error the simpler one should be preferred because it is likely to have lower generalization error."*

Problem: what does "simple" or "intuitive" mean?

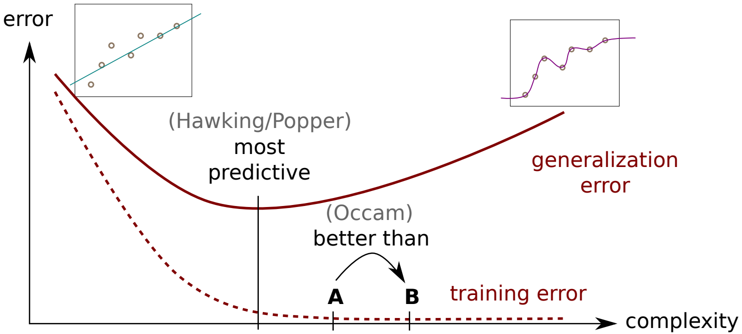
Falsifiability/prediction strength (S. Hawking, after K. Popper)

> *"[a good model] must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements,* **and** *it must make definite predictions about the results of future observations."*

($\Rightarrow$ The model with lowest generalization error is preferable.)

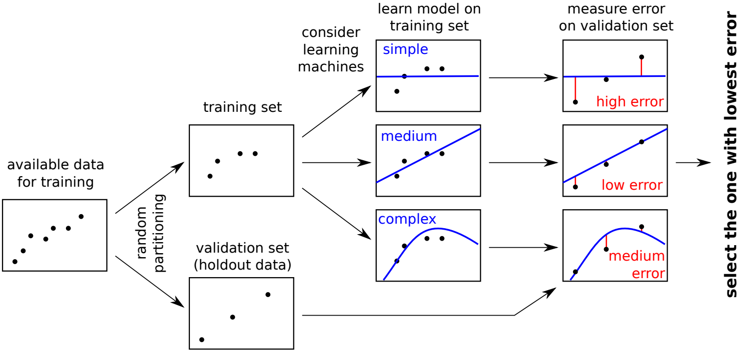
Approach 2: From Occam's Razor to Popper's

> *"[a good model] must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements,* ***and*** *it must make definite predictions about the results of future observations."*



The Holdout Selection Procedure

Idea: Hold out some of the available data for model selection.



Remarks:

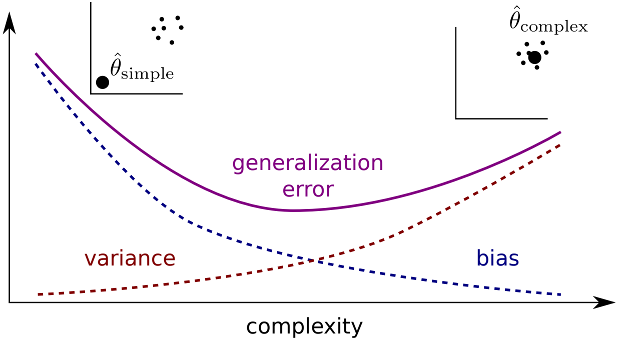
- Holding out data can lower the quality of the trained model.
- There is a trade-off between the amount of data used for training and the amount of data used for validation.

Bias-Variance Decomposition of Error

Let $\mathcal{D}$ and $\hat{\theta}$ be *random variables*. The error of a machine learning model can be decomposed as:

$$\text{Error}(\hat{\theta}) = \text{Bias}(\hat{\theta}) + \text{Variance}(\hat{\theta})$$

where the <u>bias</u> is the systematic deviation from the truth θ, and the <u>variance</u> is the noise caused by sampling $\mathcal{D}$. The higher the complexity, the closer $\hat{\theta}$ follows $\mathcal{D}$ and the more noisy it becomes.



Wrap-up

- **(1st) Occam's Razor:** Given two models with the same generalization error, the simpler one should be preferred, because simplicity is desirable in itself.
- **Model Selection/Validation:** How to make sure that a model predicts well? By testing it on out-of-sample data. The k-fold procedure can be used for parameter selection and error estimation.
- **Bias-Variance Decomposition:** The performance of a predictive model can be decomposed into bias and variance. Biased estimators are sometimes preferable.

Never use the test set to train the model or select the parameters.

Part III: Machine Learning In Practice

The typical ML system

Input: an event or a fact.

- E.g., scene, sound, movement, click.

Sensing: a (technical) system making an observation.

- E.g., human, camera, micro-phone, SW-system.

The typical ML system

Segmentation is the isolation of the interesting parts of the data.

- E.g., cropping image around an object.
- Segmentation is one of the hardest ML problems!
- How to know what is interesting for a given task? **Chicken - egg problem.**

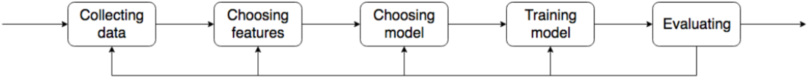
The typical ML system

Extracting "useful" features and estimating/extracting target "signal":

- The distinction between feature extraction and decision/estimation motivated by practical reasons.
- E.g., the best possible feature to extract would be the class label.
- Feature extraction is typically based on heuristics and not "automatically" learned.

Post-processing: pooling of decisions/estimations, evaluation routines.

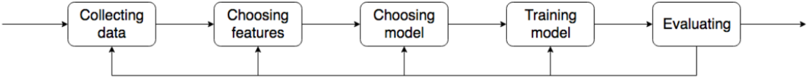
The typical design workflow



Iterative cycle:

- Re-examine steps if the evaluation is not good enough.
- Each step is influences the next steps!

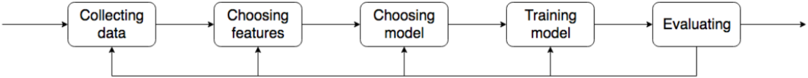
Data collection



Crucial first step! It is very important to have a **representative data**. Can be, e.g.,:

- (Synthetic) data generation
- Harvesting (and labeling) of data

Choosing features



Again: distinction between features selection and model feature learning of practical nature.

- **"Choosing features" typically entails pre-processing and preparing data using expert knowledge.**
- This data is then presented to the learning machine.
- In contrast, some learning machines can learn to extract useful features from the input data.

An example in text-processing, e.g., many systems remove stop-words from the input.

Choosing model: representation & optimization

In this step one chooses a **knowledge representation and a process to optimize it:**

- Knowledge representation.
- Optimization.

Typically one chooses a model family and in the next step an actual model instance is chosen.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Credit: A Few Useful Things to Know about Machine Learning, Domingos

Evaluation

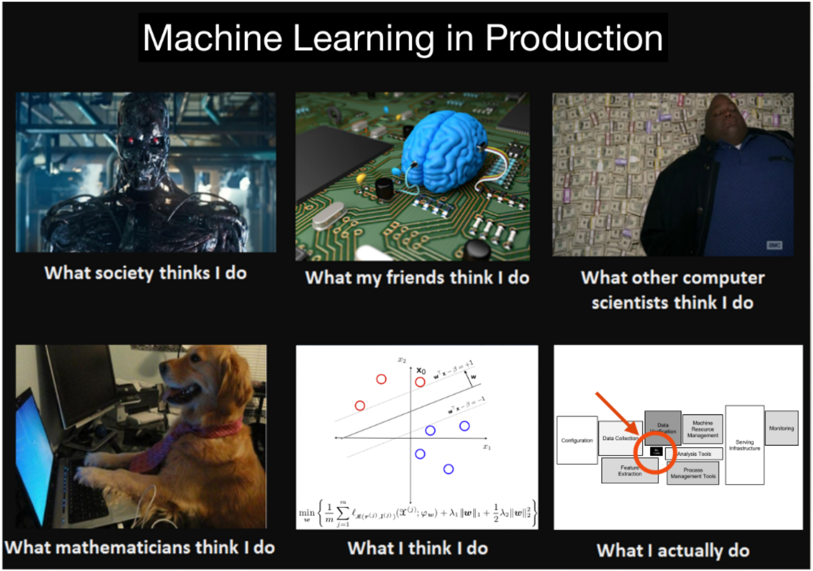
Given a set of models one needs to choose the best candidate:

- Best means, a model performs best on future data with respect to the chosen evaluation metric.

Common metrics are:

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Credit: A Few Useful Things to Know about Machine Learning, Domingos

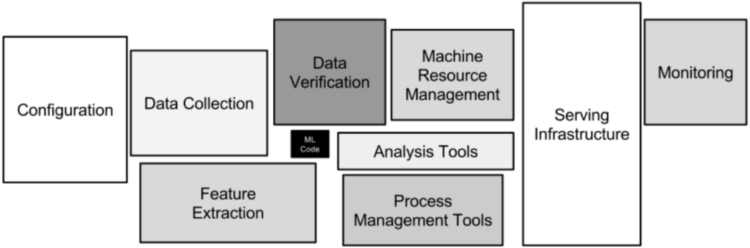


Credit: Felix Biessmann

Production aspects

Model engineering is typically only small fraction of total production effort:

- ML-software: **software complexity plus data complexity**.



From research to production

Research code often ends up in a creative chaos!

- Pipeline jungles: preprocessing evolved over time and is distributed over many modules.
- Glue code: getting data in and out of learning systems.
- Dead experimental code paths.
- Complicated experiment configurations.

This can make it a challenge to transfer results of a prototype into production system!

In production

A few points to consider:

- Input distributions are: unstable or shift over time?
- **Re-training a model: If, when and how often?**
  - Automatic training safety check: new model needs to perform better than old model!
- How to deal with outliers?

Interaction of ML systems

Data processing systems are sensitive to data characteristics/distributions:

- It's their **"interface"**.
- Data distribution is influenced by all of "up-stream" systems. If one changes, the data semantic potentially changes.
- Even: Improving a signal producing system might hurt downstream data systems. Effects can cascade!

Feedback loops:

- The output of a system can influence it's input distribution.
  - E.g., recommender system influences taste of user, which influences the recommender system's data.
- (hidden) Output of system A influences input of system B, output of system B influences input of system A.

Summary

Summary - Part 1 (AI an ML)

- Definitions of AI and ML.
- ML as a subdomain of AI, which is mainly focused on learning models that generalize to new observations.
- ML is applicable to various fields (vision, NLP, science, etc.).
- Three principal categories of ML problems (supervised, unsupervised, reinforcement).

Summary - Part 2 (ML theory)

- Useful formalizations exist for the ML problem.
- Makes it easier to view seemingly different knowledge representations and evaluation techniques in a single unified manner.
- Learning from limited data and Occam's razor.
- Measuring model complexity and relating it to generalization.
- Looking at proxy of test error (holdout validation set)

Summary - Part 3 (ML in practice)

- Deploying a practical ML system involves a complex pipeline.
- Actual ML code is often a small fraction of the whole pipeline.
- Various additional factors to be considered in practice, e.g. feature selection, nonstationarity, feedback effects, data outliers.