A field guide to the machine learning zoo

Theodore Vasiloudis SICS/KTH

From idea to objective function

Formulating an ML problem

Formulating an ML problem

- Common aspects

Source: Xing (2015)

Formulating an ML problem

- Common aspects
  - Model (θ)

Source: Xing (2015)

Formulating an ML problem

- Common aspects
  - Model (θ)
  - Data (D)

Source: Xing (2015)

Formulating an ML problem

- Common aspects
    - Model (θ)
    - Data (D)
- Objective function: $L(\theta, D)$

Source: Xing (2015)

Formulating an ML problem

- Common aspects
  - Model (θ)
  - Data (D)
- Objective function: L(θ, D)
- Prior knowledge: r(θ)

Source: Xing (2015)

Formulating an ML problem

- Common aspects
  - Model (θ)
  - Data (D)
- Objective function: $L(\theta, D)$
- Prior knowledge: $r(\theta)$
- **ML program: $f(\theta, D) = L(\theta, D) + r(\theta)$**

Source: Xing (2015)

Formulating an ML problem

- Common aspects
  - Model (θ)
  - Data (D)
- Objective function: $L(\theta, D)$
- Prior knowledge: $r(\theta)$
- **ML program: $f(\theta, D) = L(\theta, D) + r(\theta)$**
- ML Algorithm: How to optimize $f(\theta, D)$

Source: Xing (2015)

Example: Improve retention at Twitter

- Goal: Reduce the churn of users on Twitter
- Assumption: Users churn because they don't engage with the platform
- Idea: Increase the retweets, by promoting tweets more likely to be retweeted

Example: Improve retention at Twitter

- Goal: Reduce the churn of users on Twitter
- Assumption: Users churn because they don't engage with the platform
- Idea: Increase the retweets, by promoting tweets more likely to be retweeted

- Data (D):
- Model (θ):
- Objective function - L(**D**, **θ**):
- Prior knowledge (Regularization):
- Algorithm:

Example: Improve retention at Twitter

- Goal: Reduce the churn of users on Twitter
- Assumption: Users churn because they don't engage with the platform
- Idea: Increase the retweets, by promoting tweets more likely to be retweeted

- Data (D): Features and labels, $\mathbf{x_i}$, y_i
- Model (θ):
- Objective function - L(**D**, **θ**):
- Prior knowledge (Regularization):
- Algorithm:

Example: Improve retention at Twitter

- Goal: Reduce the churn of users on Twitter
- Assumption: Users churn because they don't engage with the platform
- Idea: Increase the retweets, by promoting tweets more likely to be retweeted

- Data (D): Features and labels, $\mathbf{x}_i$, y_i
- Model (θ): Logistic regression, parameters $\mathbf{w}$
  - $p(y|\mathbf{x}, \mathbf{w}) = \text{Bernouli}(y \mid \text{sigm}(\mathbf{w}^T\mathbf{x}))$
- Objective function - L(**D**, **θ**):
- Prior knowledge (Regularization):
- Algorithm:

Example: Improve retention at Twitter

- Goal: Reduce the churn of users on Twitter
- Assumption: Users churn because they don't engage with the platform
- Idea: Increase the retweets, by promoting tweets more likely to be retweeted

- Data (D): Features and labels, $\mathbf{x}_i$, y_i
- Model (θ): Logistic regression, parameters **w**
  - $p(y|\mathbf{x}, \mathbf{w}) = \text{Bernouli}(y \mid \text{sigm}(\mathbf{w}^T\mathbf{x}))$
- Objective function - $L(\mathbf{D}, \boldsymbol{\theta})$: $NLL(\mathbf{w}) = \Sigma \log(1 + \exp(-y\, \mathbf{w}^T\mathbf{x}_i))$
- Prior knowledge (Regularization): $r(\mathbf{w}) = \lambda * \mathbf{w}^T\mathbf{w}$
- Algorithm:

Warning:
Notation abuse

Example: Improve retention at Twitter

- Goal: Reduce the churn of users on Twitter
- Assumption: Users churn because they don't engage with the platform
- Idea: Increase the retweets, by promoting tweets more likely to be retweeted

- Data (D): Features and labels, $\mathbf{x}_i$, y_i
- Model (θ): Logistic regression, parameters **w**
  - $p(y|\mathbf{x}, \mathbf{w}) = \text{Bernouli}(y \mid \text{sigm}(\mathbf{w}^T\mathbf{x}))$
- Objective function - L(**D**, **θ**): $\text{NLL}(\mathbf{w}) = \Sigma \log(1 + \exp(-y\, \mathbf{w}^T\mathbf{x}_i))$
- Prior knowledge (Regularization): $r(\mathbf{w}) = \lambda * \mathbf{w}^T\mathbf{w}$
- Algorithm: Gradient Descent

Data problems

Data problems

- GIGO: Garbage In - Garbage Out

Data readiness

Source: Lawrence (2017)

Data readiness

- Problem: “Data” as a concept is hard to reason about.
- Goal: Make the stakeholders aware of the state of the data at all stages

Source: Lawrence (2017)

Data readiness

Source: Lawrence (2017)

Data readiness

- Band C
  - Accessibility

Source: Lawrence (2017)

Data readiness

- Band C
  - Accessibility
- Band B
  - Representation and faithfulness

Source: Lawrence (2017)

Data readiness

- Band C
  - Accessibility
- Band B
  - Representation and faithfulness
- Band A
  - Data in context

Source: Lawrence (2017)

Data readiness

- Band C
  - *"How long will it take to bring our user data to C1 level?"*
- Band B
  - *"Until we know the collection process we can't move the data to B1."*
- Band A
  - *"We realized that we would need location data in order to have an A1 dataset."*

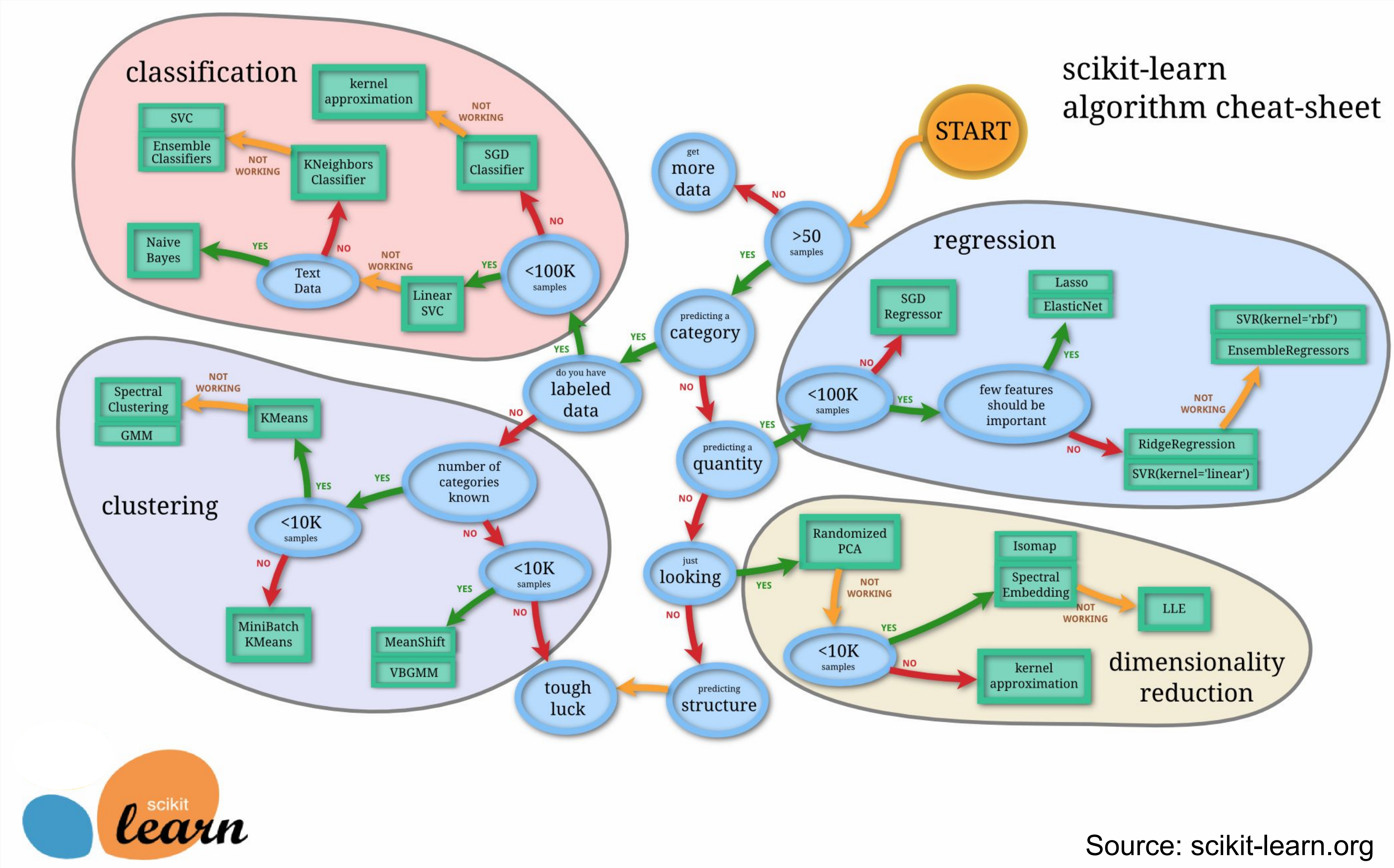
Source: Lawrence (2017)

Data readiness

- Band C
  - *"How long will it take to bring our user data to C1 level?"*
- Band B
  - *"Until we know the collection process we can't move the data to B1."*
- Band A
  - *"We realized that we would need location data in order to have an A1 dataset."*

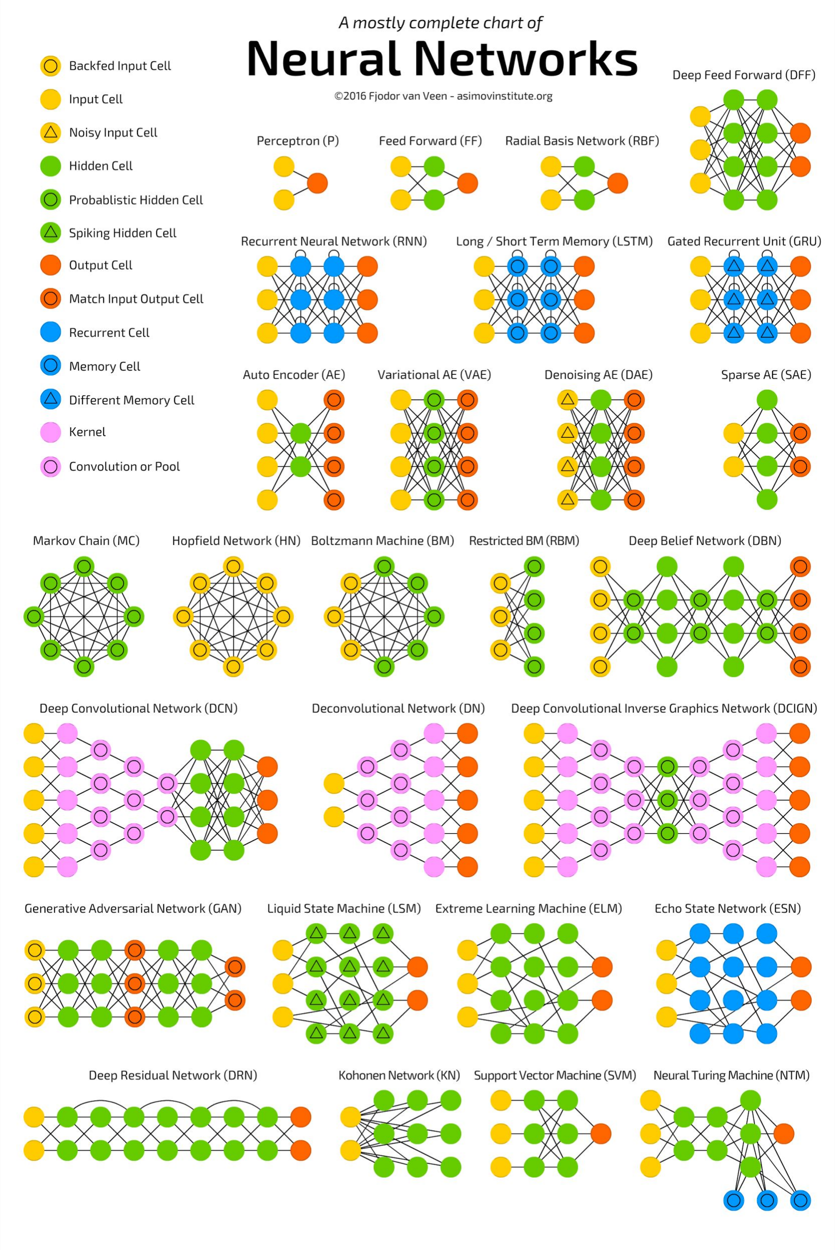
Selecting algorithm & software: “Easy” choices

Selecting algorithms



Source: scikit-learn.org

An ML algorithm “farm”



Source: Asimov Institute (2016)

The neural network zoo

Selecting algorithms

- Always go for the simplest model you can afford

Selecting algorithms

- Always go for the simplest model you can afford
    - Your first model is more about getting the infrastructure right

Source: Zinkevich (2017)

Selecting algorithms

- Always go for the simplest model you can afford
  - Your first model is more about getting the infrastructure right
  - Simple models are usually interpretable. Interpretable models are easier to debug.

Source: Zinkevich (2017)

Selecting algorithms

- Always go for the simplest model you can afford
  - Your first model is more about getting the infrastructure right
  - Simple models are usually interpretable. Interpretable models are easier to debug.
  - Complex model erode boundaries

Source: Sculley et al. (2015)

Selecting algorithms

- Always go for the simplest model you can afford
  - Your first model is more about getting the infrastructure right
  - Simple models are usually interpretable. Interpretable models are easier to debug.
  - Complex model erode boundaries
    - CACE principle: Changing Anything Changes Everything

Source: Sculley et al. (2015)

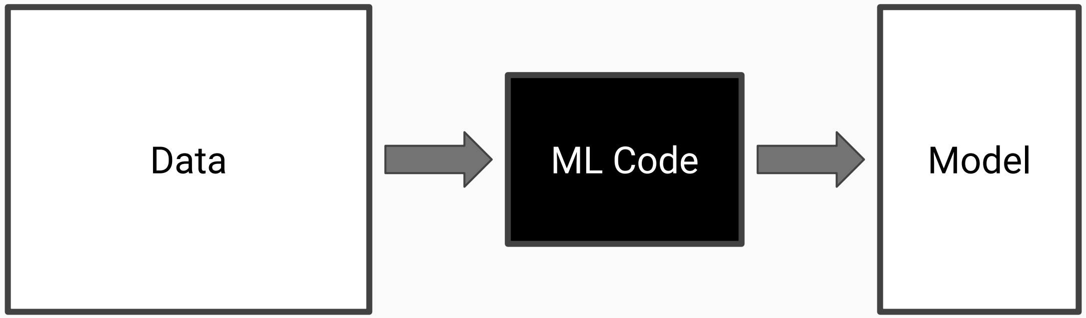
Selecting software

PaddlePaddle
mxnet
scikit learn
TensorFlow
PYTORCH
theano
torch
Microsoft CNTK
DMTK
DL4J DEEPLEARNING4J
Caffe
BID DATA
H2O.ai
Chainer
将軍
K
Dlib C++ Library
mahout
Spark MLlib
Leaf
Edward
PHP-ML
Jubatus
WEKA The University of Waikato
VW
moa
PETUUM
PyMC3

The ML software zoo

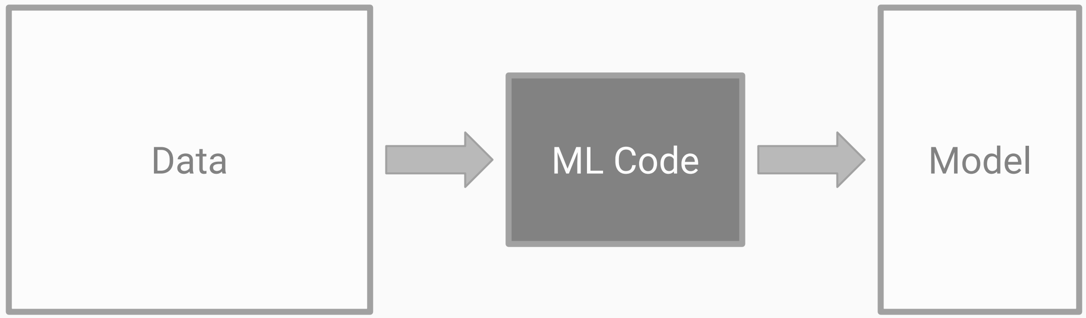
Your model vs. the world

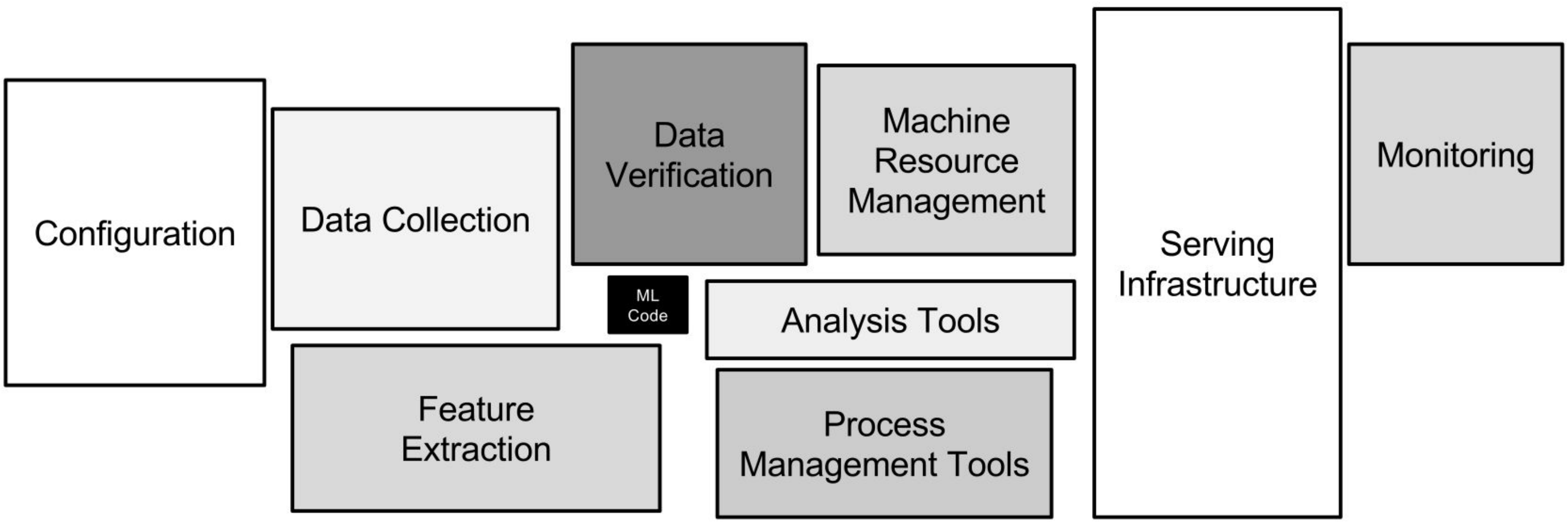
What are the problems with ML systems?



Expectation

What are the problems with ML systems?





Sculley et al. (2015)

Reality

Things to watch out for

Things to watch out for

- Data dependencies

Sculley et al. (2015)
& Zinkevich (2017)

Things to watch out for

- Data dependencies
    - Unstable dependencies

Sculley et al. (2015)
& Zinkevich (2017)

Things to watch out for

- Data dependencies
  - Unstable dependencies
- Feedback loops

Sculley et al. (2015)
& Zinkevich (2017)

Things to watch out for

- Data dependencies
  - Unstable dependencies
- Feedback loops
  - Direct

Sculley et al. (2015)
& Zinkevich (2017)

Things to watch out for

- Data dependencies
  - Unstable dependencies
- Feedback loops
  - Direct
  - Indirect

Sculley et al. (2015)
& Zinkevich (2017)

Bringing it all together

Bringing it all together

- Define your problem as optimizing your objective function using data
- Determine (and monitor) the readiness of your data
- Don't spend too much time at first choosing an ML framework/algorithm
- Worry much more about what happens when your model meets the world.

Thank you.

@thvasilo

tvas@sics.se

Sources

- Google auto-replies: Shared photos, and text
- Silver et al. (2016): Mastering the game of Go
- Xing (2015): A new look at the system, algorithm and theory foundations of Distributed ML
- Lawrence (2017): Data readiness levels
- Asimov Institute (2016): The Neural Network Zoo
- Zinkevich (2017): Rules of Machine Learning - Best Practices for ML Engineering
- Sculley et al. (2015): Hidden Technical Debt in Machine Learning Systems