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 ($\theta$)
  ○ Data ($D$)
Formulating an ML problem

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

Source: Xing (2015)
Formulating an ML problem

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

Source: Xing (2015)
Formulating an ML problem

- **Common aspects**
  - Model \( \theta \)
  - 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 ($\theta$)
  - 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, \( x_i, y_i \)
- Model (\( \theta \)):
- Objective function - \( L(D, \theta) \):
- 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, $x_i, y_i$
- Model ($\theta$): Logistic regression, parameters $w$
  - $p(y|x, w) = \text{Bernouli}(y \mid \text{sigm}(w^T x))$
- Objective function - $L(D, \theta)$:
- 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, \( x_i, y_i \)
- Model (\( \theta \)): Logistic regression, parameters \( w \)
  - \( p(y|x, w) = \text{Bernoulli}(y | \text{sigm}(w^T x)) \)
- Objective function - \( L(D, \theta) \): \( \text{NLL}(w) = \sum \log(1 + \exp(-y w^T x_i)) \)
- Prior knowledge (Regularization): \( r(w) = \lambda w^T w \)
- 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, $x_i, y_i$
- Model ($\theta$): Logistic regression, parameters $w$
  - $p(y|x, w) = \text{Bernoulli}(y | \text{sigm}(w^T x))$
- Objective function - $L(D, \theta)$: $\text{NLL}(w) = \sum \log(1 + \exp(-y w^T x_i))$
- Prior knowledge (Regularization): $r(w) = \lambda * w^T w$
- 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
An ML algorithm “farm”
The neural network zoo

Source: Asimov Institute (2016)
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.
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 models erode boundaries
    ■ CACE principle: Changing Anything Changes Everything
Selecting software
Your model vs. the world
What are the problems with ML systems?
What are the problems with ML systems?

Sculley et al. (2015)
Things to watch out for
Things to watch out for

- Data dependencies
Things to watch out for

- Data dependencies
  - Unstable dependencies

Things to watch out for

- Data dependencies
  - Unstable dependencies
- Feedback loops

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

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.
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](#)