Introduction to (practical) machine learning

Balázs Kégl

Linear Accelerator Laboratory and Computer Science Laboratory
CNRS & University of Paris Sud
Paris-Saclay Center for Data Science

DSSP, June 12, 2015

What is machine learning (a.k.a predictive analytics)?

- Where does it fit into the pipeline?
- What is supervised learning, why is it important?
- What are the main principles when building/tuning analytics solutions?
- What are the main techniques, and how do we choose from them?
- What do *you* want to know in the domain?

Machine learning a.k.a predictive analytics

- Induce functions (programs) from data
- More and more data, (different kinds of) problems, algorithms/tools, computational power $\rightarrow$ ML is ubiquitious
- Theoretical guarantees are nice, but rarely provide practical guidelines
- Today's objective: pitfalls to avoid, issues to focus on, some answers to common questions

Outline

It is generalization that counts

- We don't have access to the function we'd like to optimize
- $\rightarrow$ evaluation – cross-validation
- $\rightarrow$ regularization – often intertwined with the original loss
  - optimizing surrogate losses may actually improve the generalization performance on the original loss
- data is not enough
  - $\rightarrow$ representation, model
  - $\rightarrow$ regularization

Outline

Define a quantitative objective

- It should capture your prediction loss/gain
- Something off-the shelf?
- Does it look similar to something that has been solved?
    - convert it into a known problem (preprocessing), refine the result (postprocessing)
- Computability, differentiability, convexity

Some examples

- Standard
  - counting: accuracy (error rate), precision/recall, f-measure, AUC
  - regression: squared error, squared log error (relative error)
  - probabilistic: likelihood, KL divergence, posterior probability, perplexity
- Exotic
  - ranking: (N)DCG, ERR
  - discovery: AMS
- Real-time: computational cost
- Cost-sensitivity

Outline

Know your data

- Sampling bias
  - for the classical (cross validation) setup to work, the data you learn on should come from the same distribution as the real data
- Data leakage
- Data forensics
  - No access to the person who collected the data / knows the sensors

Know your data

- Features
  - numerical, nominal
  - transformations, normalization, flattening, whitening, 1-hot, binarization, binning
  - weighting
  - domain knowledge, invariances, symmetries $\rightarrow$ feature engineering
  - iterate with building the predictor and evaluating
  - noise and curse of dimensionality, training complexity $\rightarrow$ careful with adding features

Know your data

- Missing values
  - completely at random, at random, not at random
  - listwise deletion, mean substitution, (stochastic) regression substitution
  - expectation maximization

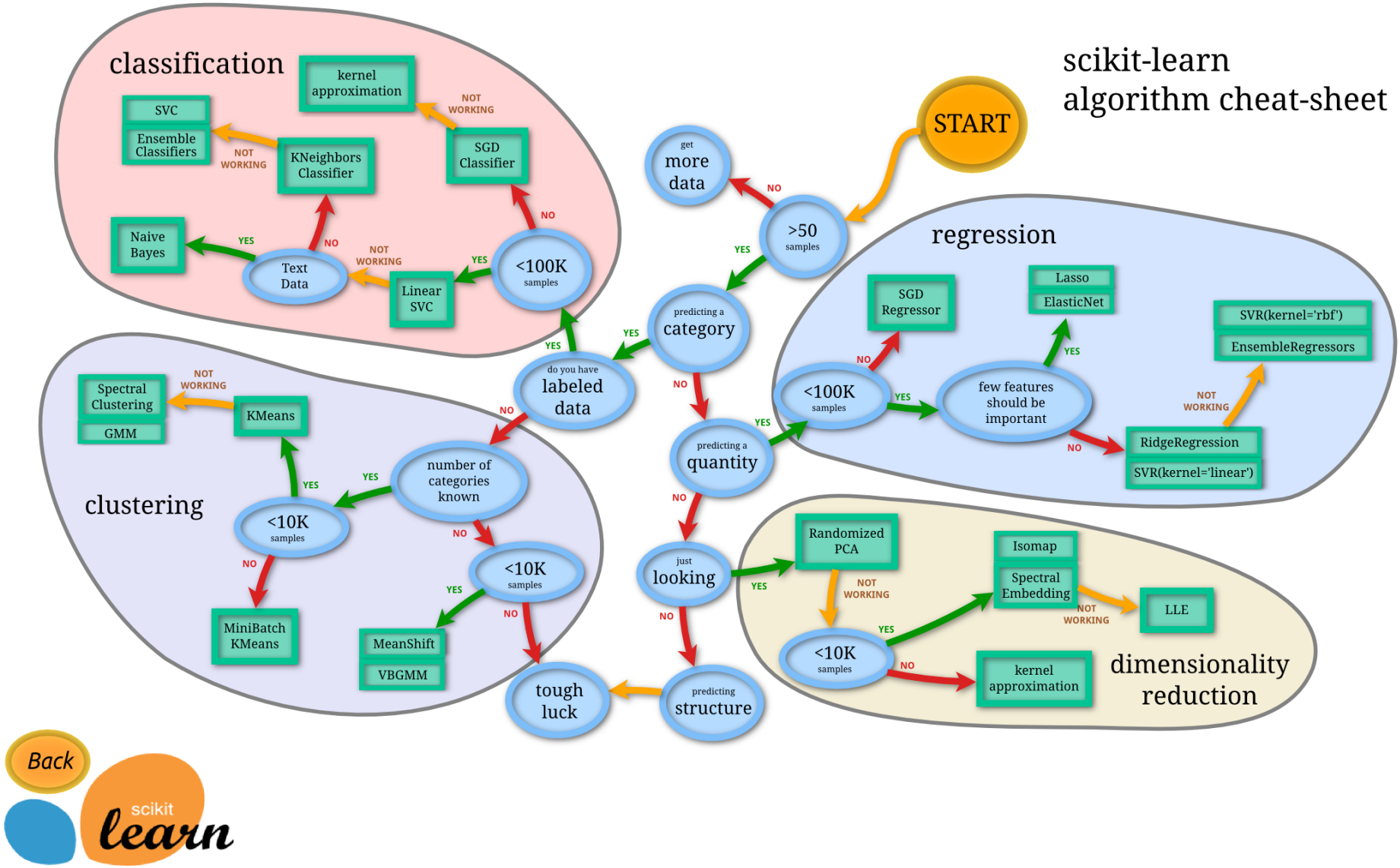
Outline

Know your model

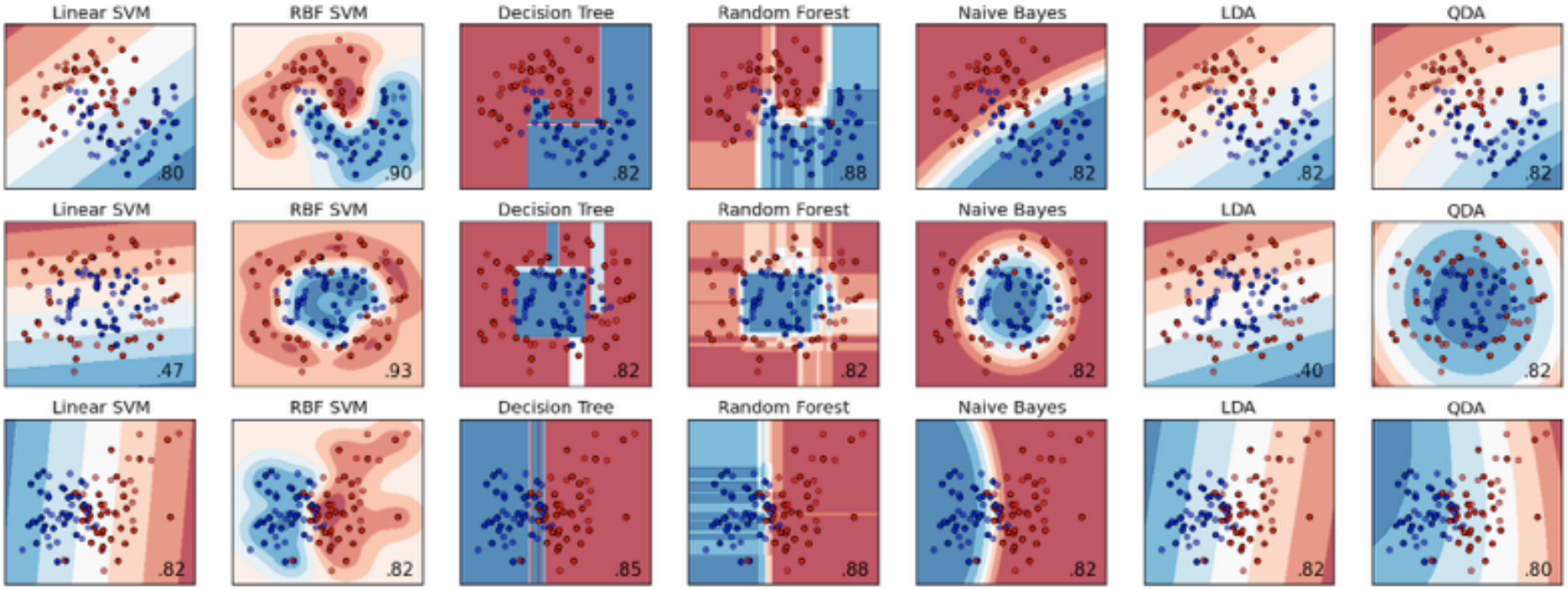
- No free lunch: if you don't have a model, you can't predict anything
- Explicit: parametric statistics (likelihood, prior, posterior)
- Implicit
  - smoothness, sparsity $\rightarrow$ regularization
  - features or similarity? $\rightarrow$ algorithm
  - class of models, representation $\rightarrow$ algorithm

Outline

Know your algorithms



Know your algorithms



Representation

- linear
- generalized linear model: $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
- kernel methods: sophisticated pattern matching
- feature families and sparsity
- interpretability

Optimization

- gradient descent
- linear programming
- quadratic programming
- convex programming
- greedy
- anytime
- global/local optimum
- batch – minibatch – stochastic/online

Regularization

- model selection
- prior (parametric/non-parametric)
- penalizing complexity ($\not\propto$ number of parameters)
- smoothness $\rightarrow L_2 = \sum_t \alpha_t^2$ – weight decay, early stopping, ridge regression
- sparsity $\rightarrow L_1 = \sum_t \alpha_t$ – forward selection LASSO, LARS
- ensemble averaging
- dropout

Over and underfitting

- Bias and variance
- What is in play
  - noise
  - dimensionality
  - data size
  - predictor complexity
  - optimization
- Diagnostics?
  - cross-validation
  - learning curves
  - error/ablative analysis

The curse of dimensionality

- low-dimensional intuitions break down
- high-dimensional spaces are empty
- mass is concentrated in shells and corners
- similarity-based approaches break down: everybody is equally far
- random directions almost always add noise, bring towards low-density regions
- low-dimensional (filament-like) manifolds
- in a million-dimensional space even a linear classifier is very complex
- strong false assumptions sometimes better than weak true ones

Outline

Evaluation

- regularization can help fighting overfitting, but at the end of the day, algorithms will have to be selected, parameters will have to be tuned, the final performance will have to be assessed through

cross-validation

Tuning

- optimizing an expensive function (train + test)
- grid search
- Bayesian surrogate optimization
- stochastic optimization

Post-processing

- classification threshold
- calibration
- model averaging: state-of-the-art performance is rarely achieved by one single predictor