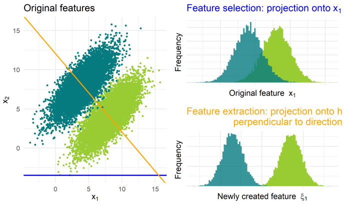
Introduction to Machine Learning

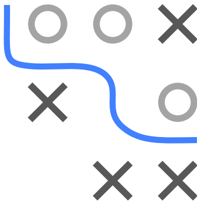
Feature Selection
Feature Selection: Introduction



Learning goals

- Too many features can be harmful in prediction
- Selection vs. extraction
- Types of selection methods

INTRODUCTION



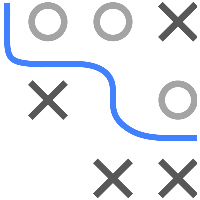
Feature selection:
Finding a well-performing, hopefully small set of features for a task.

Feature selection is critical for

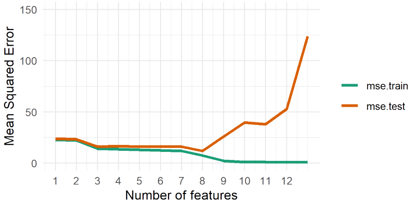
- reducing noise and overfitting
- improving performance/generalization
- enhancing interpretability by identifying most informative features

Features can be selected based on domain knowledge, or data-driven algorithmic approaches. We focus on the latter here.

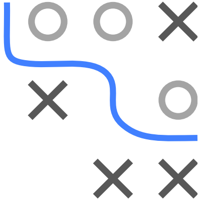
MOTIVATION



- Naive view:
  - More features $\rightarrow$ more information $\rightarrow$ discriminant power $\uparrow$
  - Model is not harmed by irrelevant features since their parameters can simply be estimated as 0.
- In practice, irrelevant and redundant features can "confuse" learners (see **curse of dimensionality**) and worsen performance.
- Example: In linear regression, R^2 is monotonically increasing in p, but adding irrelevant features leads to overfitting (capturing noise).



MOTIVATION / 2



- In high-dimensional data sets, we often have prior information that many features are either irrelevant or of low quality
- Having redundant features can cost something during prediction (money or time)
- Many models require $n > p$ data. Thus, we either need to
  - adapt models to high-dimensional data (e.g., regularization)
  - design entirely new procedures for $p > n$ data
  - use filter preprocessing methods from this lecture

SIZE OF DATASETS

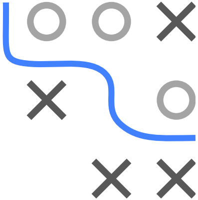
Many new forms of technical measurements and connected data leads to availability of extremely high-dimensional data sets.

- **Classical setting**: Up to around 10^2 features, feature selection might be relevant, but benefits often negligible.

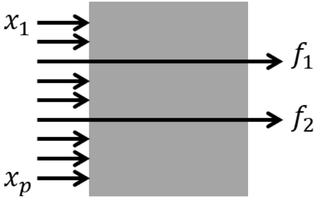
- **Datasets of medium to high dimensionality**: At around 10^2 to 10^3 features, classical approaches can still work well, while principled feature selection helps in many cases.

- **High-dimensional data**: 10^3 to 10^9 or more features. Examples: micro-array / gene expression data and text categorization (bag-of-words features). If we also have few observations, scenario is called $p \gg n$.

FEATURE SELECTION VS. EXTRACTION

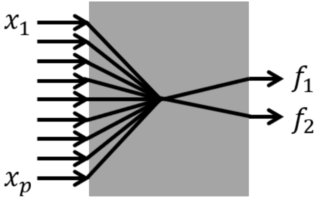


Feature selection



- Creates a subset of original features **x** by selecting $\tilde{p} < p$ features $\boldsymbol{f}$.
- Retains information on selected individual features.

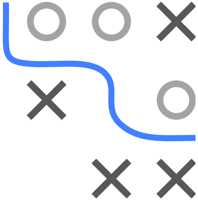
Feature extraction

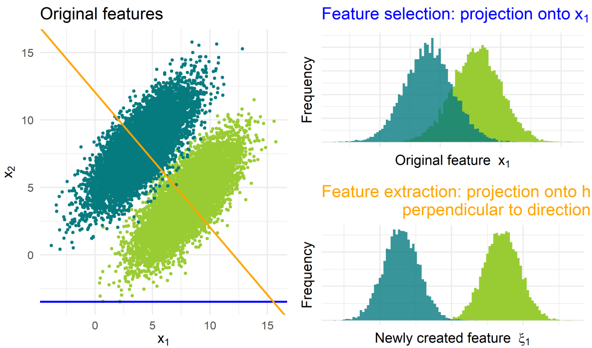


- Maps p features in **x** to $\tilde{p}$ extracted features $\boldsymbol{f}$.
- Info on individual features can be lost through (non-)linear combination.

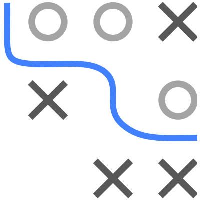
FEATURE SELECTION VS. EXTRACTION / 2

- Both FS and FE contribute to 1) dimensionality reduction and 2) simplicity of models
- FE can be unsupervised (PCA, multidim scaling, manifold learning) or supervised (supervised PCA, partial least squares)
- FE can produce lower dim projections which can work better than FS; whether FE+model is interpretable depends on how interpretable extracted features are





TYPES OF FEATURE SELECTION METHODS



In rest of the chapter, we introduce different types of methods for FS:

- Filters: evaluate relevance of features using statistical properties such as correlation with target variable
- Wrappers: use a model to evaluate subsets of features
- Embedded methods: integrate FS directly into specific model - we look at them in their dedicated chapters (e.g., CART, L_0, L_1)

Example: embedded method (Lasso) regularizing model params with $L1$ penalty enables “automatic" feature selection:

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} |\theta_j|$$