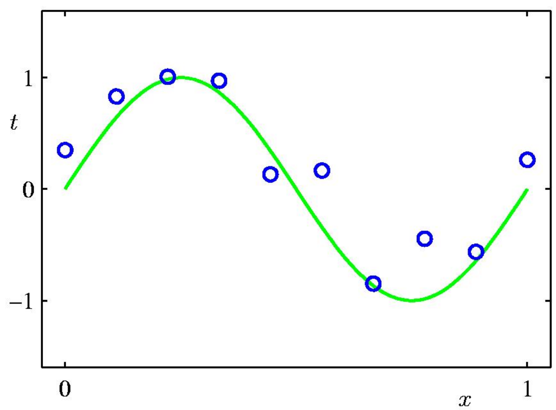
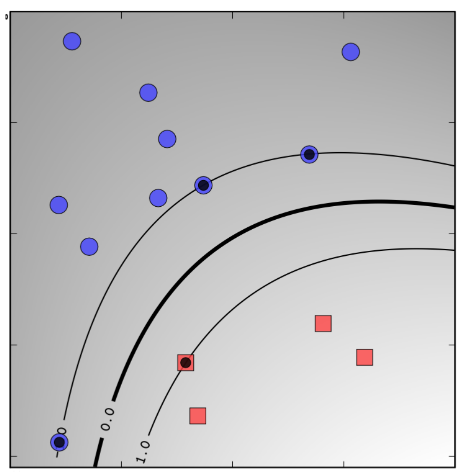
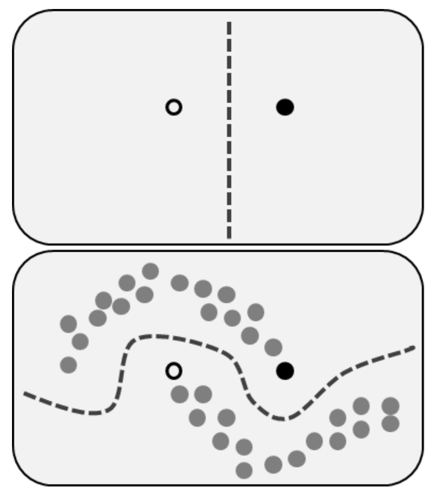
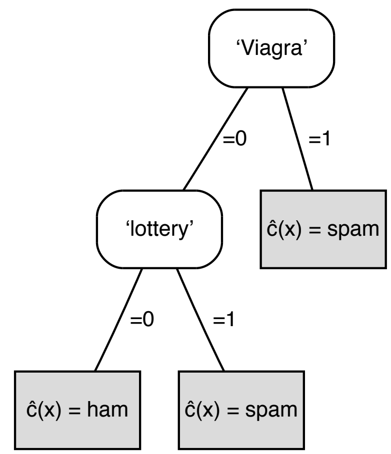
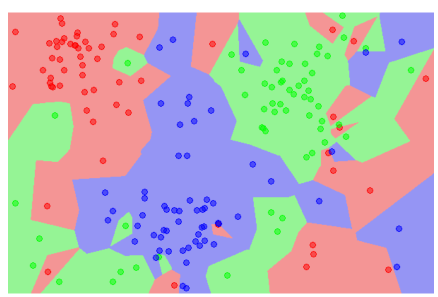
Epilogue: what have you learned this semester?

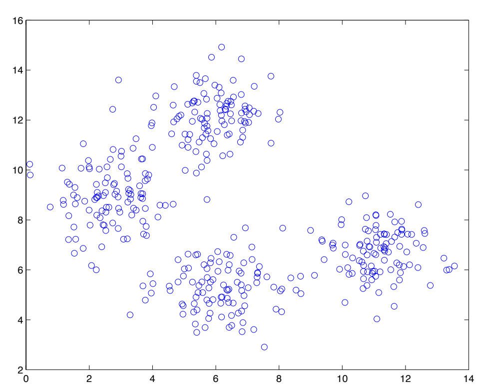












What did you get out of this course?

What skills have you learned in this course that you feel would be useful?

What are the most important insights you gained this semester?

What advice would you give future students?

What was your biggest challenge in this course?

What would you like me to do differently?

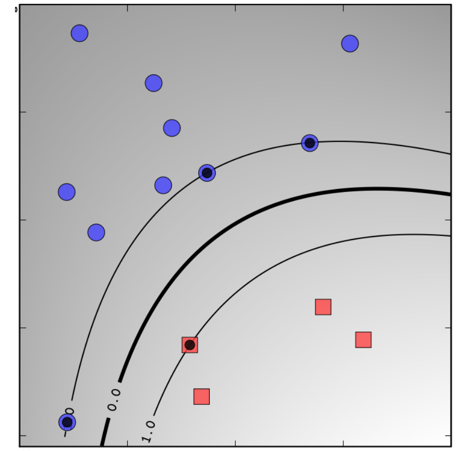
What I hope you got out of this course

The machine learning toolbox

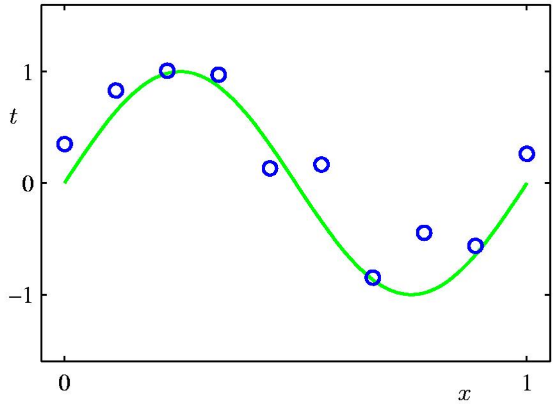
- Formulating a problem as an ML problem
- Understanding a variety of ML algorithms
- Running and interpreting ML experiments
- Understanding what makes ML work – theory and practice

Learning scenarios we covered

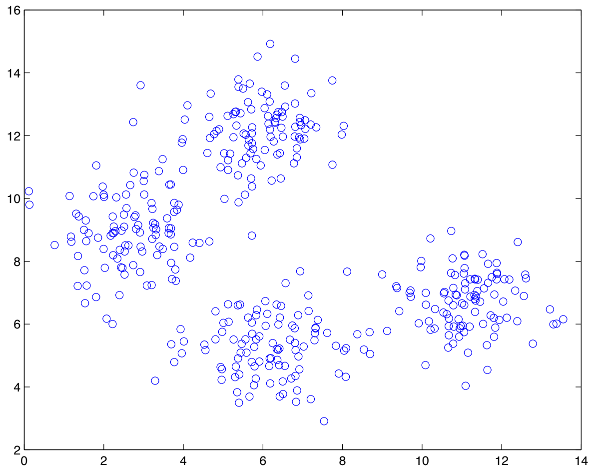
Classification: discrete/categorical labels



Regression: continuous labels

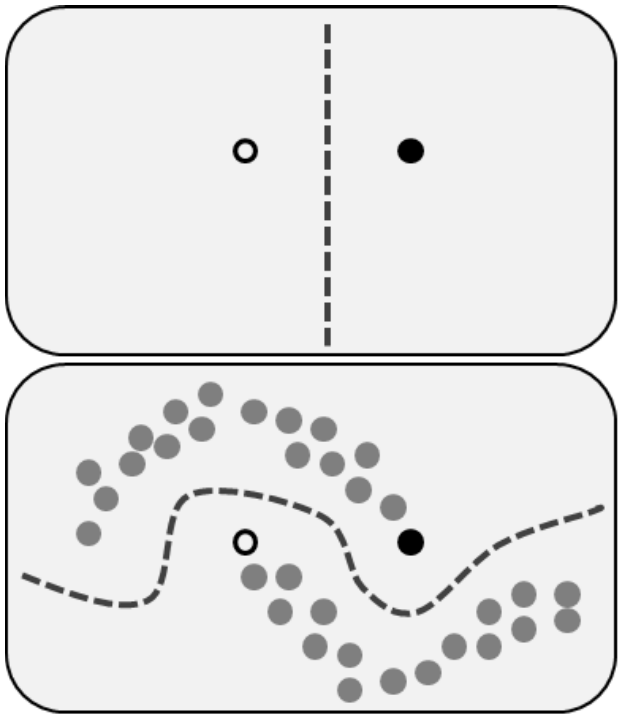


Clustering: no labels

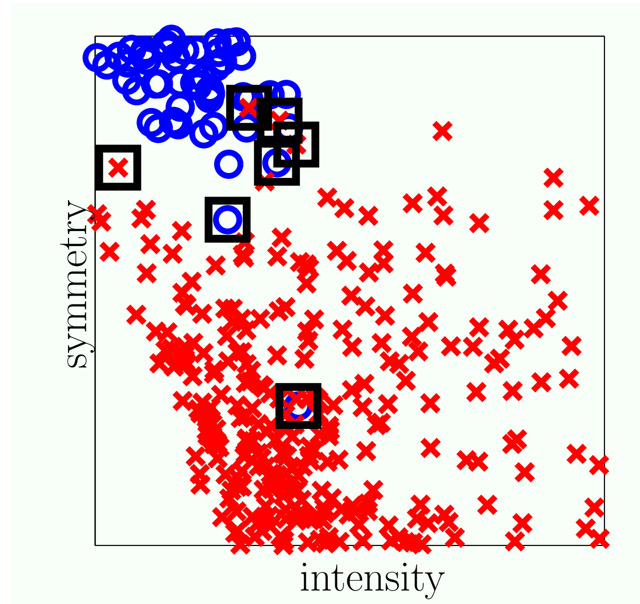


A variety of learning tasks

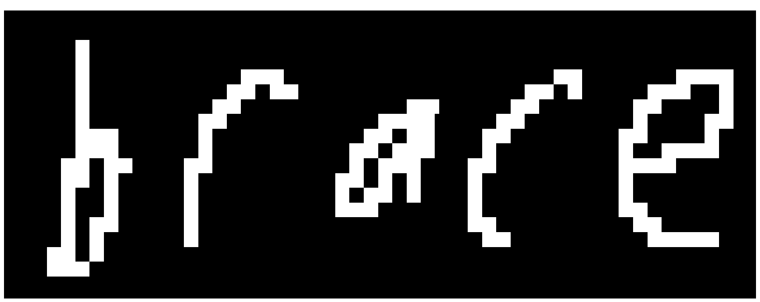
- Supervised learning
- Unsupervised learning
- Semi-supervised learning
  - Access to a lot of unlabeled data
- Multi-label classification
  - Each example can belong to multiple classes
- Multi-task classification
  - Solving multiple related tasks



A variety of learning tasks



- Outlier/novelty detection
  - Novelty: anything that is not part of the normal behavior of a system.
- Reinforcement learning
  - Learn action to maximize payoff
- Structured output learning

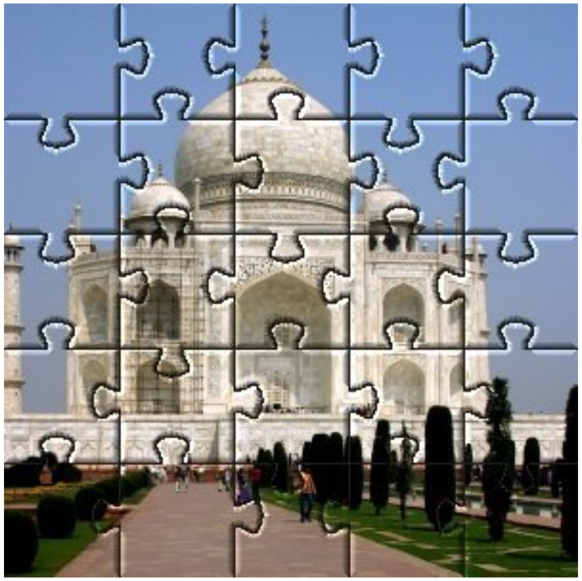


brace

Learning in structured output spaces

Handle prediction problems with complex output spaces

- Structured outputs: multivariate, correlated, constrained

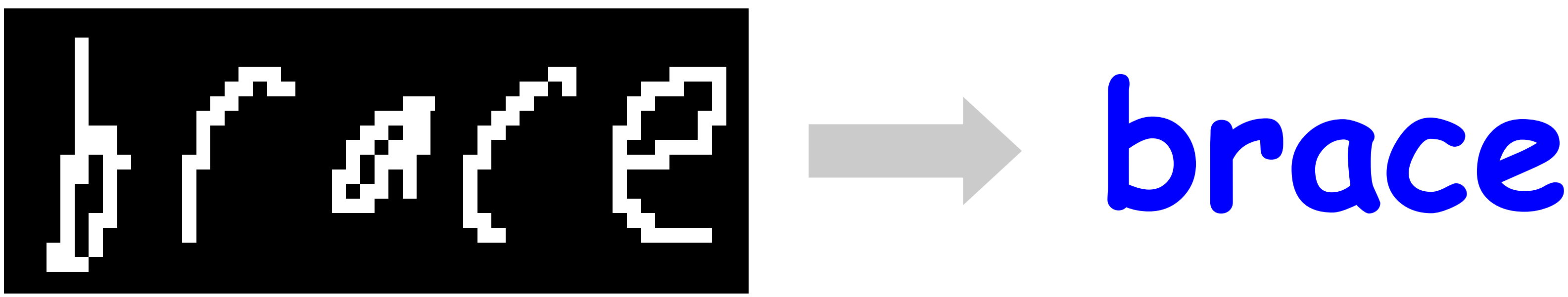


General way to solve many learning problems

Examples taken from Ben Taskar's 07 NIPS tutorial

Local vs. Global

Global classification takes advantage of correlations and satisfies the constraints in the problem



Other techniques

- Graphical models (conditional random fields, Bayesian networks)
- Bayesian model averaging

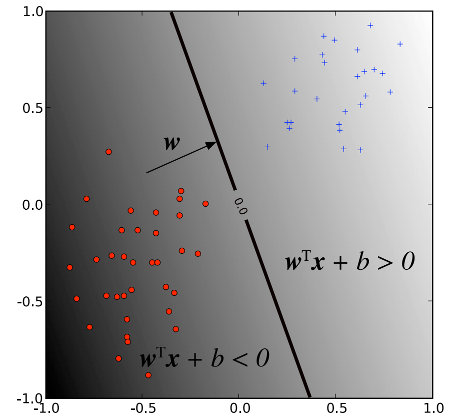
The importance of features and their representation

Choosing the right features is one of the most important aspects of applying ML.

What you can do with features:

- Normalization
- Selection
- Construction
- Fill in missing values

Types of models

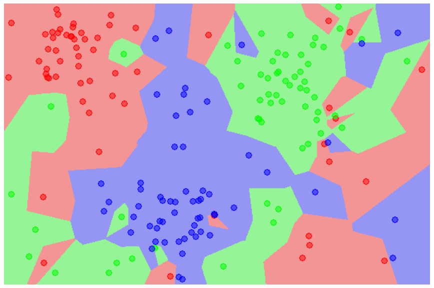


Geometric

- Ridge-regression, SVM, perceptron
- Neural networks

Distance-based

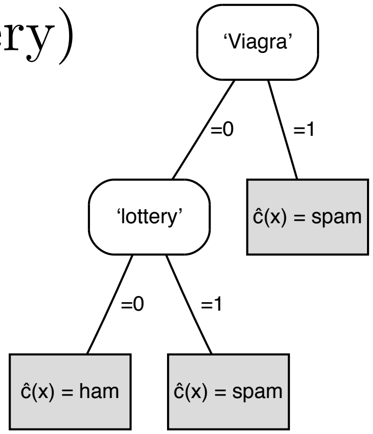
- K-nearest-neighbors



Probabilistic

- Naïve-bayes

$$P(Y = \text{spam} | \text{Viagara}, \text{lottery})$$



Logical models: Tree/Rule based

- Decision trees

Ensembles

Loss + regularization

Many of the models we studied are based on a cost function of the form:

loss + regularization

Example:
Ridge regression

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\intercal \mathbf{x}_i)^2 + \lambda \mathbf{w}^\intercal \mathbf{w}$$

Loss + regularization for classification

SVM $$\frac{C}{N}\sum_{i=1}^{N}\max\left[1 - y_i h_{\mathbf{w}}(\mathbf{x}_i), 0\right] + \frac{1}{2}\mathbf{w}^{\top}\mathbf{w}$$

Hinge loss L$_2$ regularizer

The hinge loss is a margin maximizing loss function

Can use other regularizers:

$||\mathbf{w}||_1$ (L$_1$ norm)

Leads to very sparse solutions and is non-differentiable.

Elastic Net regularizer: $\alpha||\mathbf{w}||_1 + (1-\alpha)||\mathbf{w}||_2^2$

Loss + regularization for classification

SVM

$$\frac{C}{N}\sum_{i=1}^{N}\max\left[1-y_i h_{\mathbf{w}}(\mathbf{x}_i), 0\right]+\frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

Hinge loss L$_2$ regularizer

Logistic regression

$$\frac{1}{N}\sum_{i=1}^{N}\log(1+\exp(y_i h_{\mathbf{w}}(\mathbf{x}_i)))+\frac{\lambda}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

Log loss L$_2$ regularizer

AdaBoost can be shown to optimize the exponential loss

$$\frac{1}{N}\sum_{i=1}^{N}\exp(-y_i h_{\mathbf{w}}(\mathbf{x}_i))$$

Loss + regularization for regression

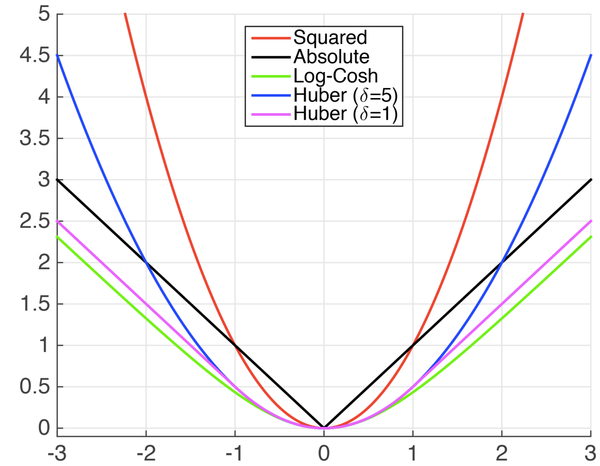
Ridge regression $$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\intercal\mathbf{x})^2 + \lambda\mathbf{w}^\intercal\mathbf{w}$$

Closed form solution; sensitivity to outliers

Lasso $$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\intercal\mathbf{x})^2 + \lambda||\mathbf{w}||_1$$

Sparse solutions; non-differentiable

Can use alternative loss functions



Comparison of learning methods

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

Characteristic	Neural Nets	SVM	Trees	random forests	k-NN,
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▲	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▲	▼
Computational scalability (large N)	▼	▲	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▼	▼
Predictive power	▲	▲	▼	▲	◆

Table 10.1 from "Elements of statistical learning"

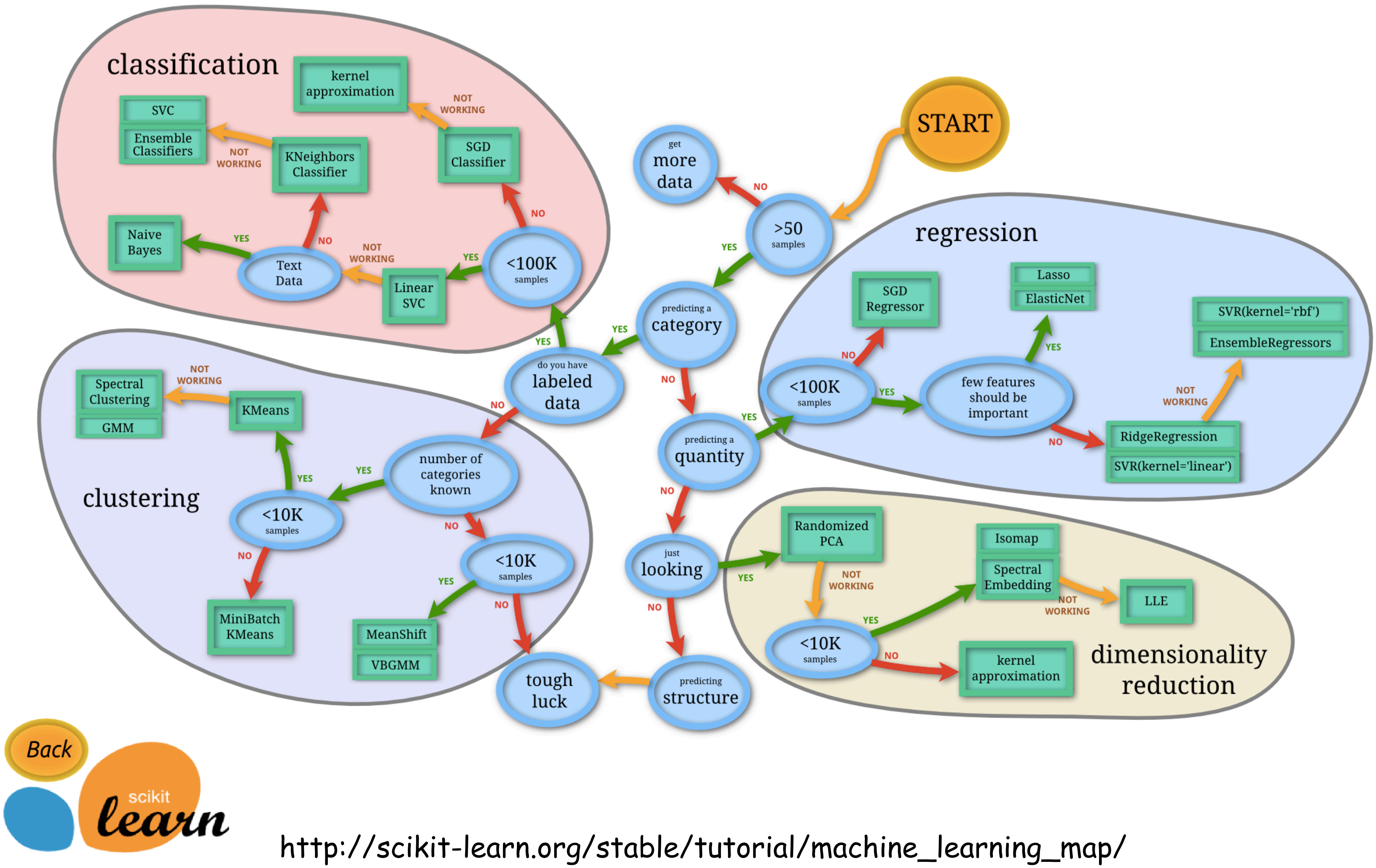
Comparison of learning methods

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

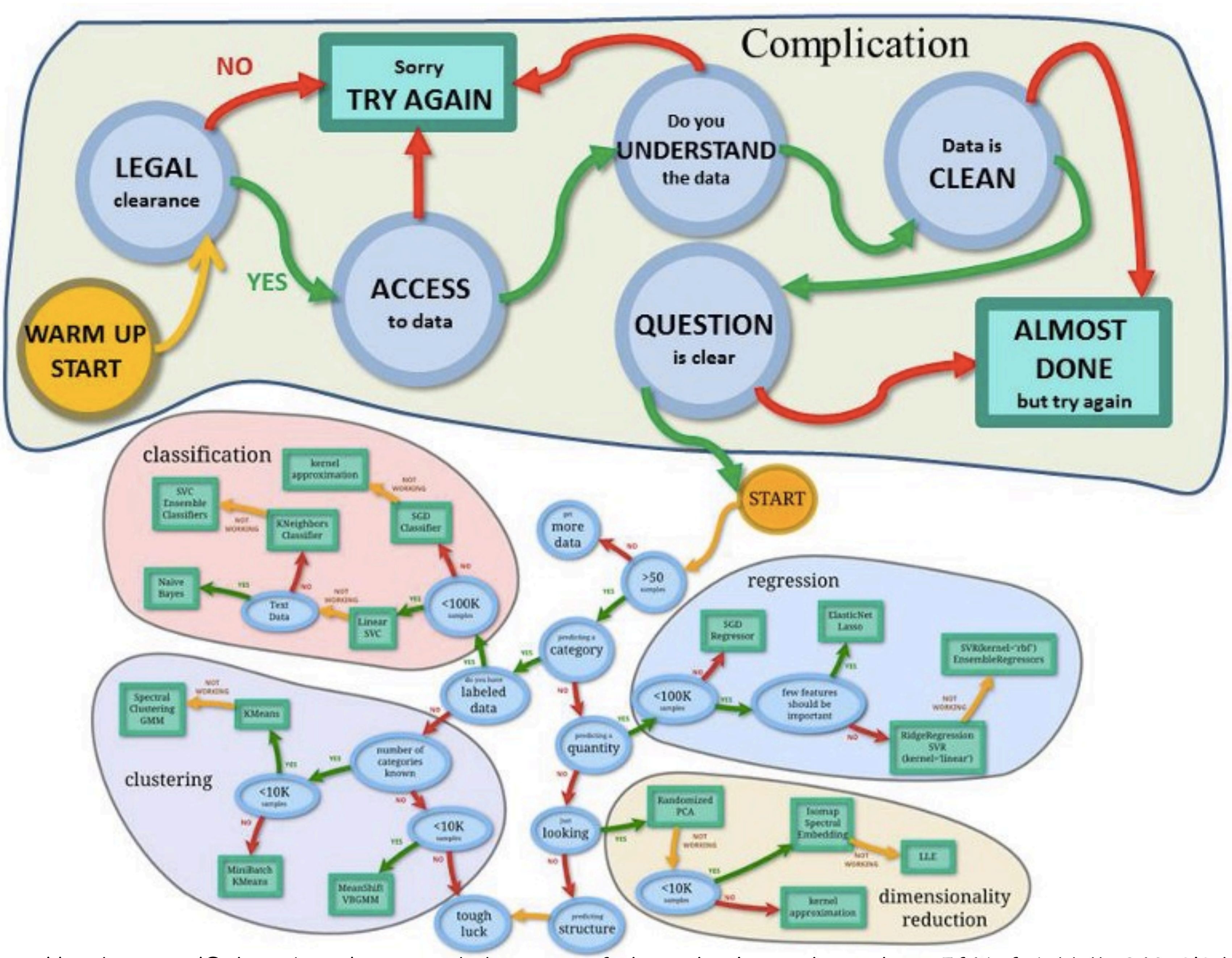
Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

Table 10.1 from "Elements of statistical learning"

The scikit-learn algorithm cheat sheet



http://scikit-learn.org/stable/tutorial/machine_learning_map/



https://medium.com/@chris_bour/an-extended-version-of-the-scikit-learn-cheat-sheet-5f46efc6cbb#.g942x8l3d

Applying machine learning

Always try multiple models

- What would you start with?

If accuracy is not high enough

- Design new features
- Collect more data