Brief Introduction to Machine Learning

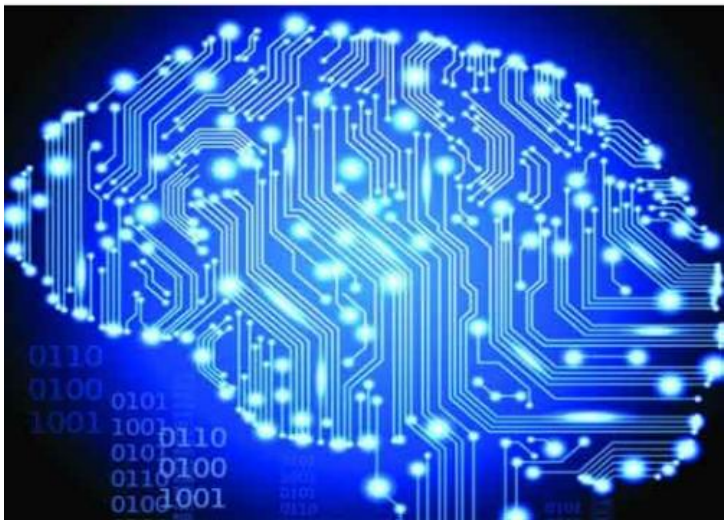
Hong Chang

Institute of Computing Technology,
Chinese Academy of Sciences

Pattern Recognition and Machine Learning (Fall 2022)

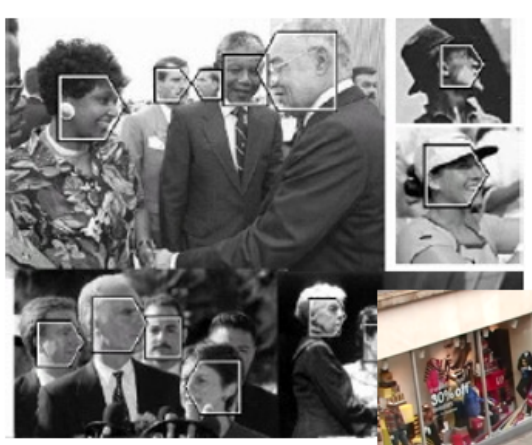
Outline I

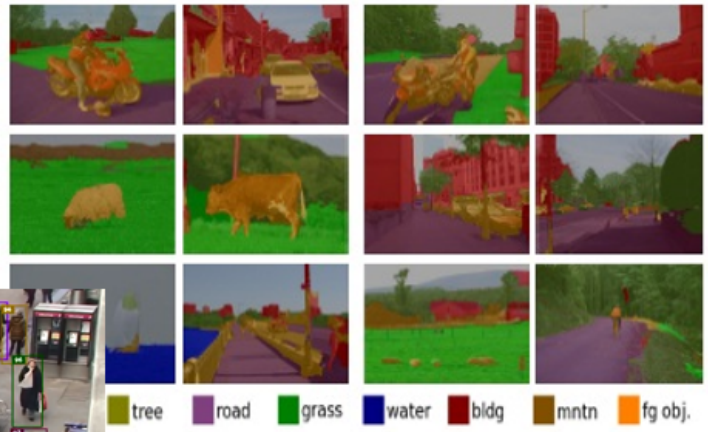
Machine Learning - A Ubiquitous Science

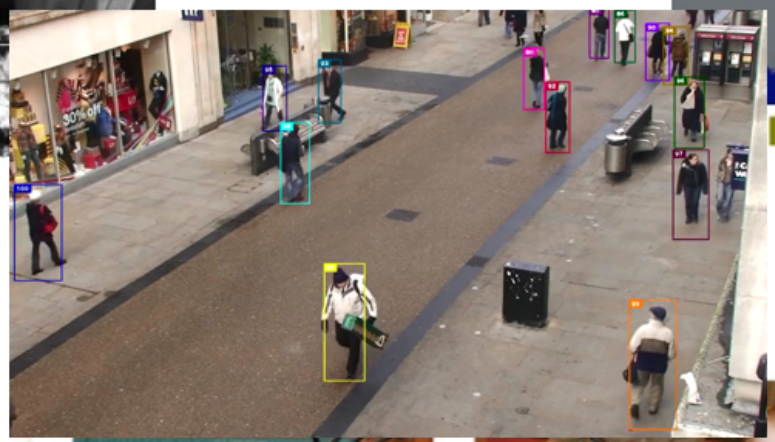


- Grand challenges in computer science and technology
  - understanding the brain, i.e. reasoning, cognition, creativity
  - creating useful intelligent machines
  - arguably AI poses the most interesting challenges and questions in computer science today
- Many research areas in computer science rely on machine learning methods

Computer Vision







"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

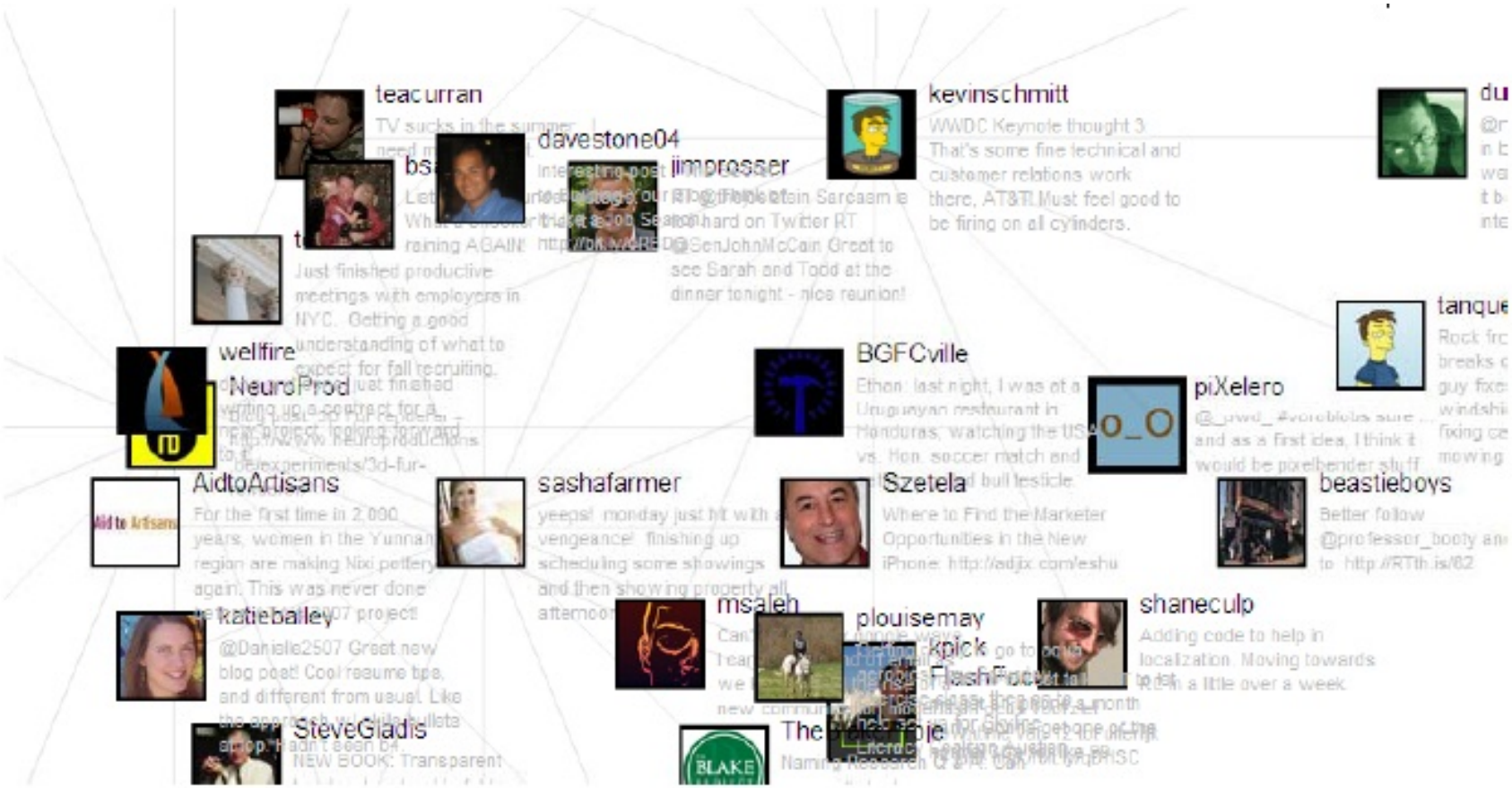
"two young girls are playing with legos toy."

"boy is doing backflip on wakeboard."

Speech Recognition

- Now most pocket Speech Recognizers or Translators are running on some sort of learning device — the more you play/use them, the smarter they become.
- Deep networks advance state of art in speech.

Social Computing

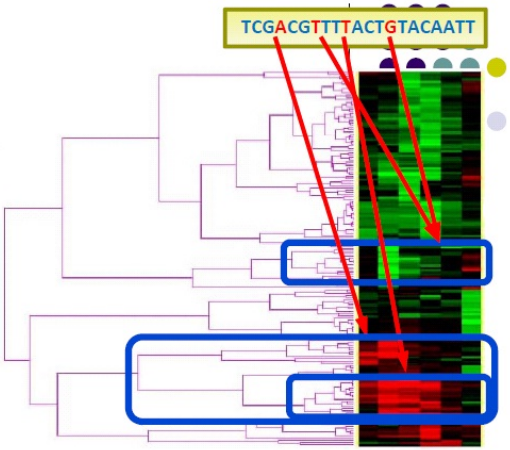


Web Search and Recommendation

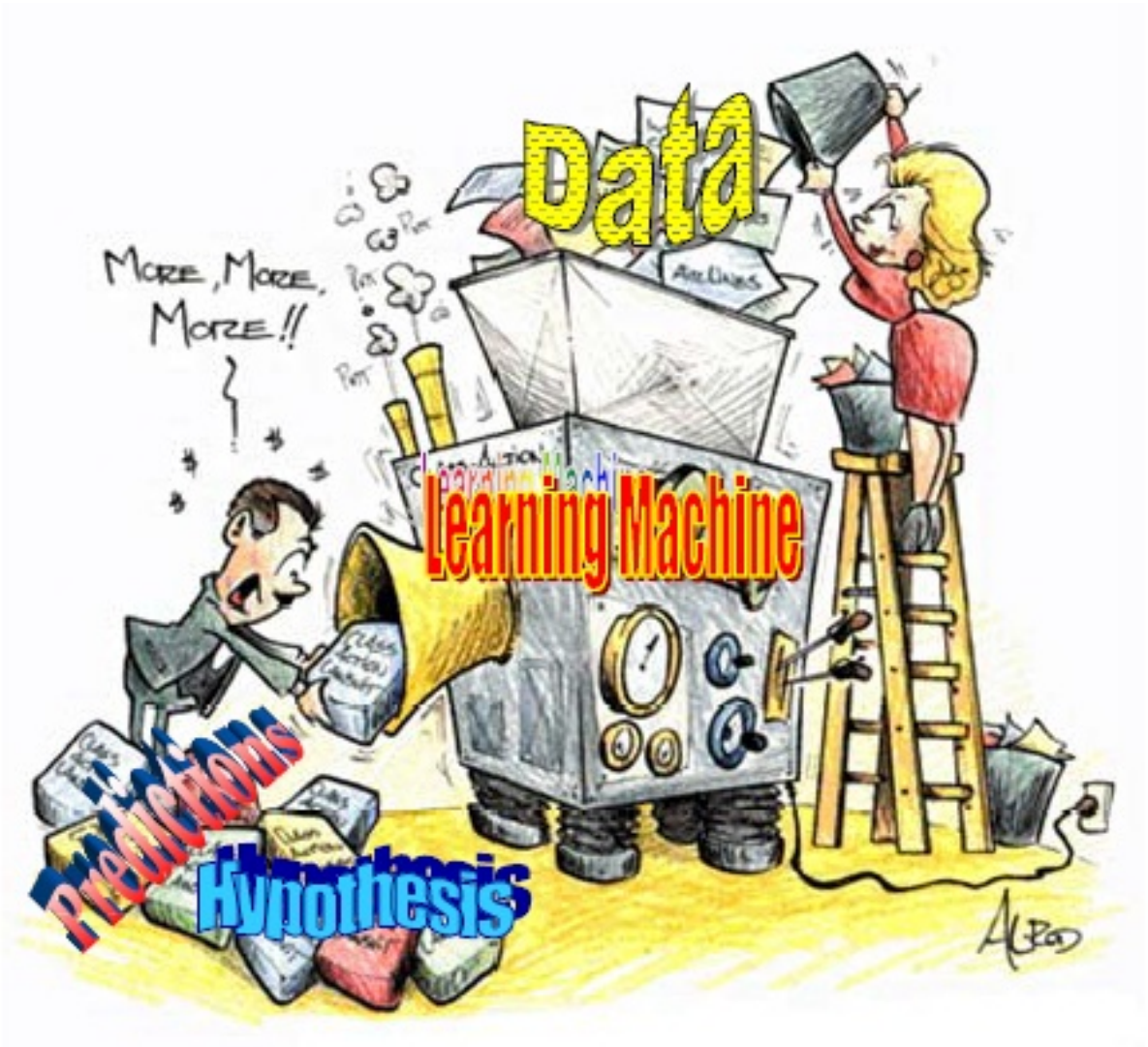
- Search engine
  - query classification, ranking, spam detection
- Computational advertising
  - estimate click-through rate, optimal ads placement
- Recommendation

Far from Exhaustion...

- Robotics
- Bioinformatics
- Question answer
- Microprocessor design
- ...



A Learning Comic



Definitions on Machine Learning

- *Wiki*: "The design and development of algorithms that take as input empirical data and yield patterns or predictions that generated the data."
- *Arthur Samuel*: "Field of study that gives computers the ability to learn without being explicitly programmed".
- *Tom M. Mitchell*: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

More specifically

- Machine Learning seeks to develop theories, methods and computer systems for
  - representing
  - classifying, clustering and recognizing
  - reasoning under uncertainty
  - predicting
  - and reacting to
  - ...

  complex, real world data, based on the system's own experience with data, and (hopefully) under a unified model or mathematical framework, that
  - can be formally characterized and analyzed
  - can take into account human prior knowledge
  - can generalize and adapt across data and domains
  - can operate automatically and autonomously
  - and can be interpreted and perceived by human.

Machine Learning and Applications

- **Machine Learning** is the science of
  making computer artifacts improve their performance
  with respect to a certain performance criterion
  using example data or past experience,
  without requiring humans to program their behavior explicitly.
- **Data Mining** (a.k.a. **knowledge discovery in databases (KDD)**)
  is the application of machine learning methods to large databases.
- **Computer Vision**
- ...

When Machine Learning is Necessary

- Human expertise is too expensive
  (e.g., intrusion detection)
- Human expertise does not exist
  (e.g., navigating on Mars)
- Humans cannot explain their expertise
  (e.g., speech recognition)
- Problem (and hence solution) changes over time
  (e.g., network routing)
- Solution needs to be adapted to particular cases
  (e.g., user biometrics for intelligent/adaptive user interface)

The Characteristics of Machine Learning

- Data is cheap and abundant; knowledge is expensive and scarce.
- Details of the data generation process may be unknown, but the process is not completely random.
- Learning models from data by exploiting certain patterns or regularities in the data: inverting the data generation path.
- A model is often not an exact replica of the complete process, but is a good and useful approximation. (George Box: “All models are wrong, but some are useful.”)
- A model may be descriptive to gain knowledge from data, or predictive to make predictions in the future, or both.
- Almost all of science is concerned with fitting models to data: induction.

Elements of Learning

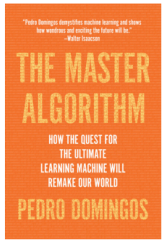
- Here are some important elements to consider before you start:
  - Task:
    - Embedding? Classification? Clustering? Topic extraction?
  - Data and other info:
    - Input and output (e.g., continuous, binary, counts, ...)
    - Supervised or unsupervised, of a blend of everything?
    - Prior knowledge? Bias?
  - Models and paradigms:
    - BN? MRF? Regression? SVM?
    - Bayesian/Frequents ? Parametric/Nonparametric?
  - Objective/Loss function:
    - MLE? MCLE? Max margin?
    - Log loss, hinge loss, square loss?
  - Tractability and exactness trade off:
    - Exact inference? MCMC? Variational? Gradient? Greedy search?
    - Online? Batch? Distributed?
  - Evaluation:
    - Visualization? Human interpretability? Predictive accuracy?
- It is better to consider one element at a time!

General Notations

$\mathbb{R}^D$	D-dimensional Euclidean space
$\mathbf{x}, \mathbf{A}$	boldface is used for vectors and matrices
$\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$	a set of N samples
$\mathbf{x} = (x_1, \ldots, x_D)^T$	x_j is j^{th} variable of $\mathbf{x}$
$\lvert\mathcal{S}\rvert$	cardinality of set $\mathcal{S}$
$(\mathbf{x}, y)$	labeled data point $\mathbf{x}$ with label y
$\mathbf{A} \in \mathbb{R}^{s \times t}$	$\mathbf{A}$ is a $s \times t$ matrix
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$\mathbf{A}^+$	pseudoinverse of matrix $\mathbf{A}$
$\mathrm{Tr}(\mathbf{A})$	trace of matrix $\mathbf{A}$
$\lVert\mathbf{A}\rVert_F$	Frobenius norm of matrix $\mathbf{A}$
$\langle\mathbf{A}, \mathbf{B}\rangle$	Frobenius product between matrix $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A} \succeq 0$	$\mathbf{A}$ is positive semi-definite

Knowledge Acquisition

- Knowledge acquisition:
  - Evolution
  - Experience
  - Culture
  - Computer
- "Most of the knowledge in the world in the future is going to be extracted by machines and will reside in machines." by Yann LeCun
- Five basic methods of computer knowledge acquisition:
  - Fill in gaps in existing knowledge
  - Emulate the human brain
  - simulate the evolutionary process
  - Systematically reduce uncertainties
  - Notice similarities between old and new information



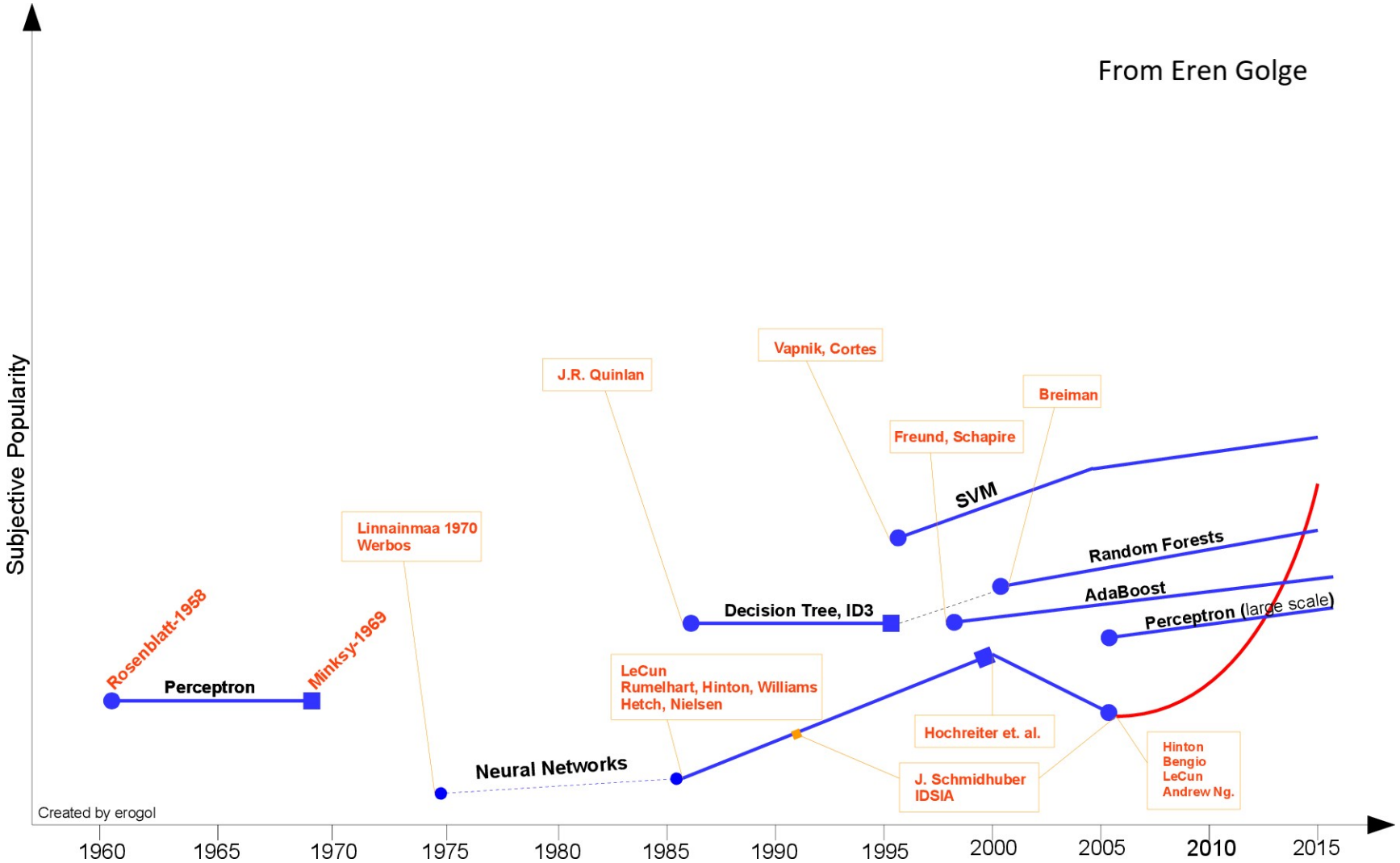
* Main Contents from Prof. Pedro Domingos.

Five Tribes of Machine Learning

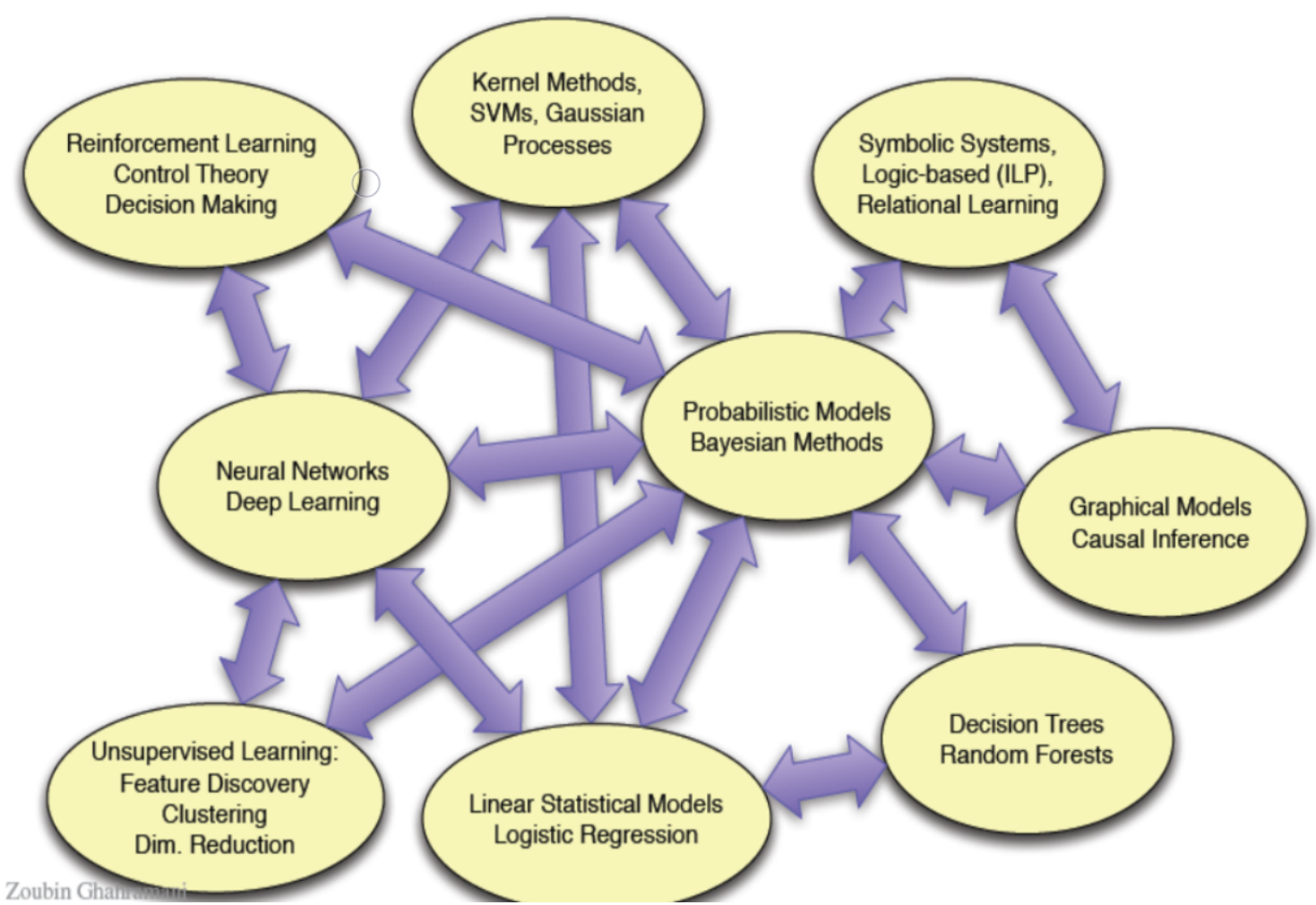
Tribes	People	Origins	Master algorithms
Symbolists	Tom Mitchell Steve Muggleton Ross Quinlan	philosophy, psychology logic	Inverse deduction
Connectionists	Yann LeCun Geoff Hinton Yoshua Bengio	neuroscience, physics	Backpropagation
Evolutionaries	John Koza John Holland Hod Lipson	genetics, evolutionary biology	Genetic programming
Bayesians	David Heckerman Judea Pearl Michael Jordan	statistics	Probabilistic inference
Analogizers	Peter Hart Vladimir Vapnik Douglas Hofstadter	psychology, mathematical optimization	Kernel machines

* from Prof. Pedro Domingos.

Machine Learning History



Machine Learning Methods



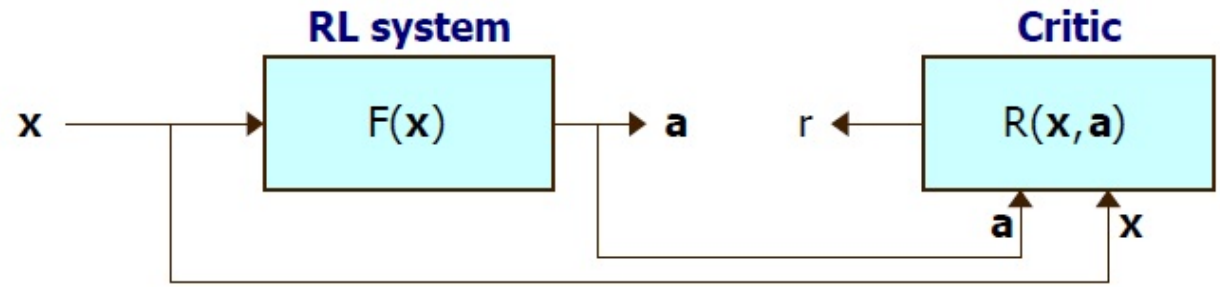
Supervised Learning

- Given $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$, learn $\hat{y} = f(\mathbf{x}; \mathbf{w})$
- Prediction of future cases:
  predict output for future input using the rule learned
  - classification: y is categorical
  - regression: y is continuous
  - ranking: y is ordinal
- Knowledge extraction:
  rule is easier to understand
- Compression:
  rule is simpler than data it explains
- Outlier detection:
  exceptions not covered by rule, e.g., fraud

Unsupervised Learning

- Given $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$, learn $\hat{y} = f(\mathbf{x}; \mathbf{w})$
- Learning "what normally happens"
- Density estimation: y is density
- Dimensionality reduction/visualization: y is lower-dimensional representation of $\mathbf{x}$
- Clustering (grouping similar instances): y is clusters
- Examples of applications:
  - Clustering: image segmentation; gene grouping
  - Image compression: color quantization
  - cocktail party problem

Reinforcement Learning



- Learning a policy (a sequence of actions) via a trial-and-error process (exploration vs. exploitation)
- No supervised output but delayed reward
- Examples of applications:
  - Game playing
  - Robot navigation in search of goal location
- Some challenging issues:
  - Multiple agents
  - Partial observability of states

Predictive/Self-Supervised Learning

- Prediction is the essence of intelligence.
- Predictive learning and self-supervised learning[1]
  - Predict any part of the input from the past
  - Predict the future from the past
  - Predict the future from the recent past
  - Predict the past from the present
  - Predict the top from the bottom
  - Predict the occluded from the visible
  - Pretend there is a part of the input you don't know and predict that

[1] Contents from Prof. Yann LeCun.

Comparison of Three Types of Learning

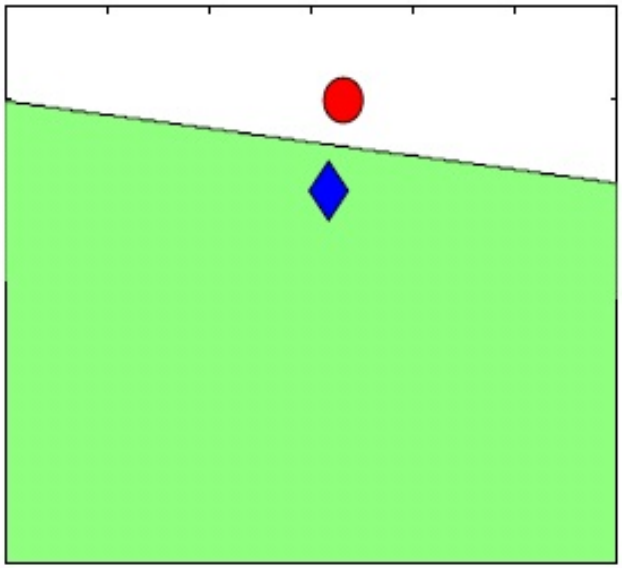
Learning Types	Machine Prediction	Feedback Information
Reinforcement Learning	a scalar reward given once in a while	very low
Supervised Learning	a category or a few numbers for each input	medium
Self-supervised Learning	any parts of its input for any observed part	high, but stochastic

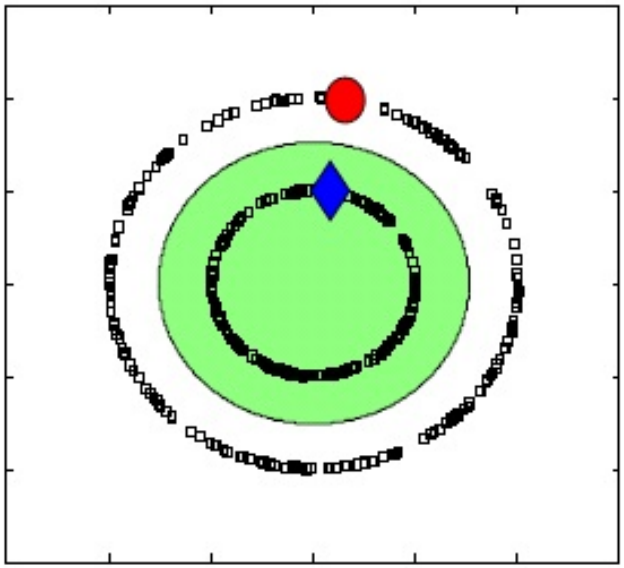
The revolution will not be supervised nor purely reinforced!

Semi-Supervised Learning

- Semi-supervised learning (SSL) is halfway between supervised and unsupervised learning.
- Supervision information is available for some, but not all, training instances.
- SSL may be regarded as:
  - Supervised learning augmented with unlabeled data,
    e.g., semi-supervised classification, semi-supervised regression
  - Unsupervised learning augmented with labeled data or constraints between data points,
    e.g., semi-supervised clustering

Semi-Supervised Learning - A Simple Example

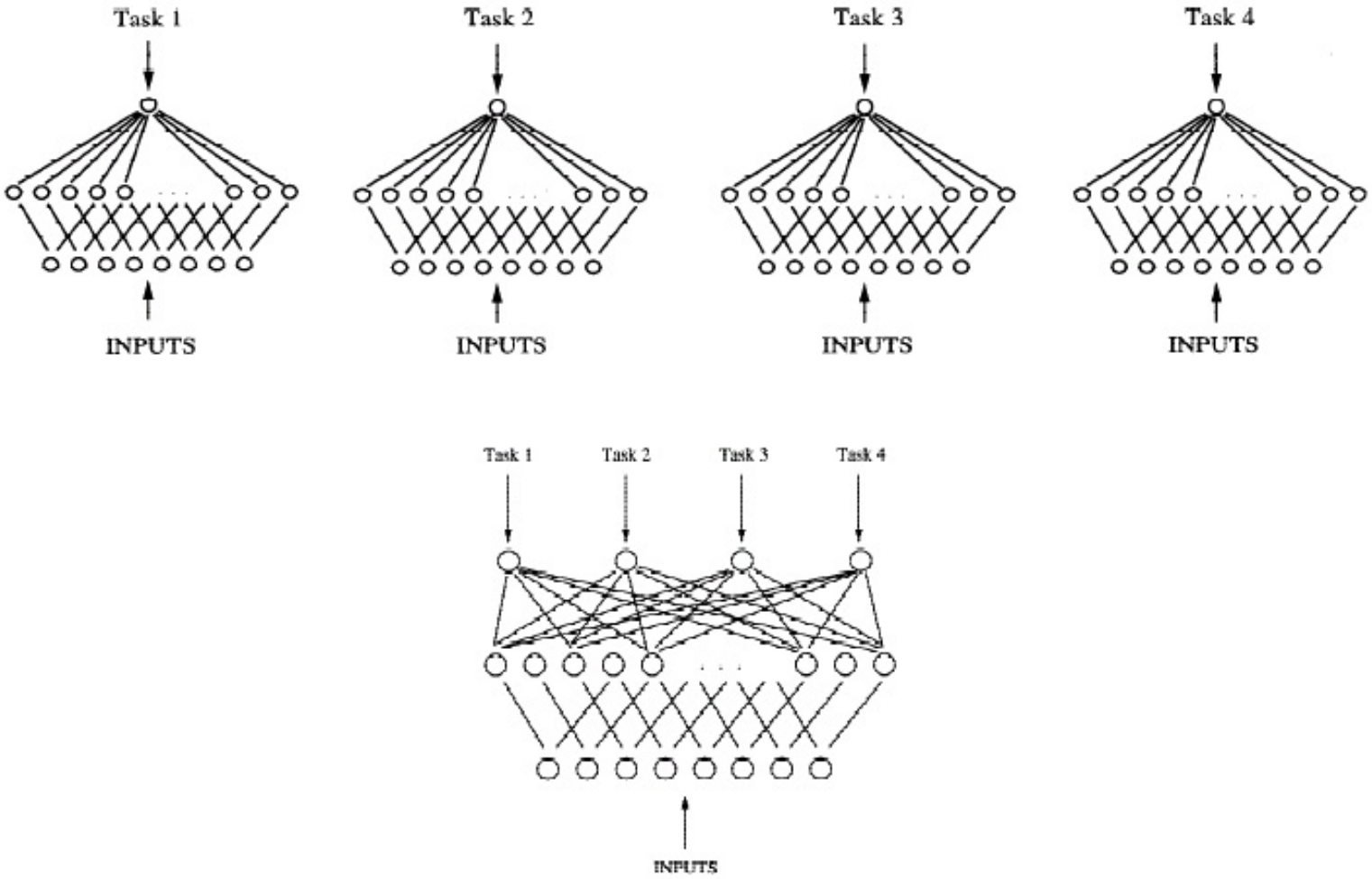




Transfer and Multi-Task Learning

- Transfer of knowledge in human learning: The experience gained through learning a task can help in learning similar tasks later.
- In many applications (e.g., spam detection), one learning task has only very few labeled training instances but there are many similar learning tasks.
- Transfer learning:
  Source tasks learned previously can help the learning of target task(s); asymmetric transfer.
- Multi-task learning:
  Multiple tasks are typically learned simultaneously to leverage the labeled data from similar tasks; symmetric transfer.

Early Multi-Task Learning - Multilayer Perceptron

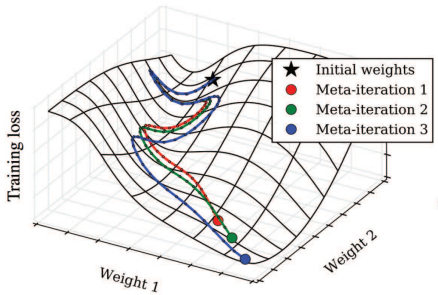


Multi-Task Learning - Sharing Information

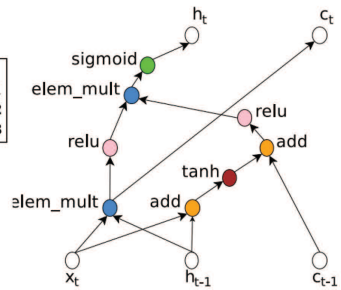
- Different tasks may share:
  - The same representation, e.g., hidden layer representation in a multilayer perceptron.
  - The same model parameters, e.g., weights in a regression function.
  - The same distribution for the model parameters, e.g., mean and covariance matrix of the multivariate normal distribution of the parameters.
  - The same cluster in some clustering structure.
- The similarity between tasks can be used to define a regularizer for the objective function.
- Negative transfer: Transfer of knowledge from a dissimilar task can degrade performance.

Meta Learning

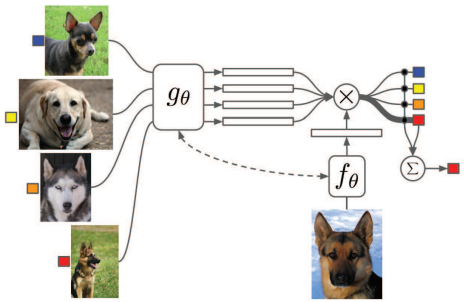
- Meta learning or Learning to learn, Inductive transfer aims to act and adapt intelligently to a wide variety of new, unseen situations.
- Typical approaches:
  - recurrent models
  - metric learning
  - learning optimizers
  - etc.



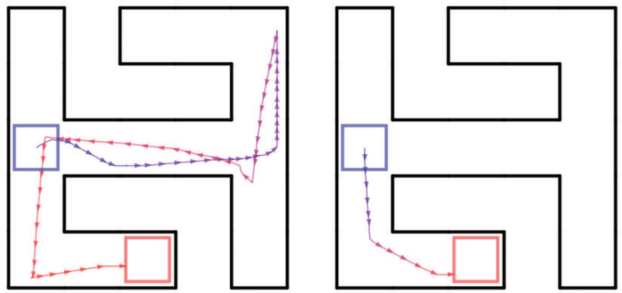
hyperparameter optimization
Maclaurin et al. '15



learned recurrent cell
Zoph & Le '17



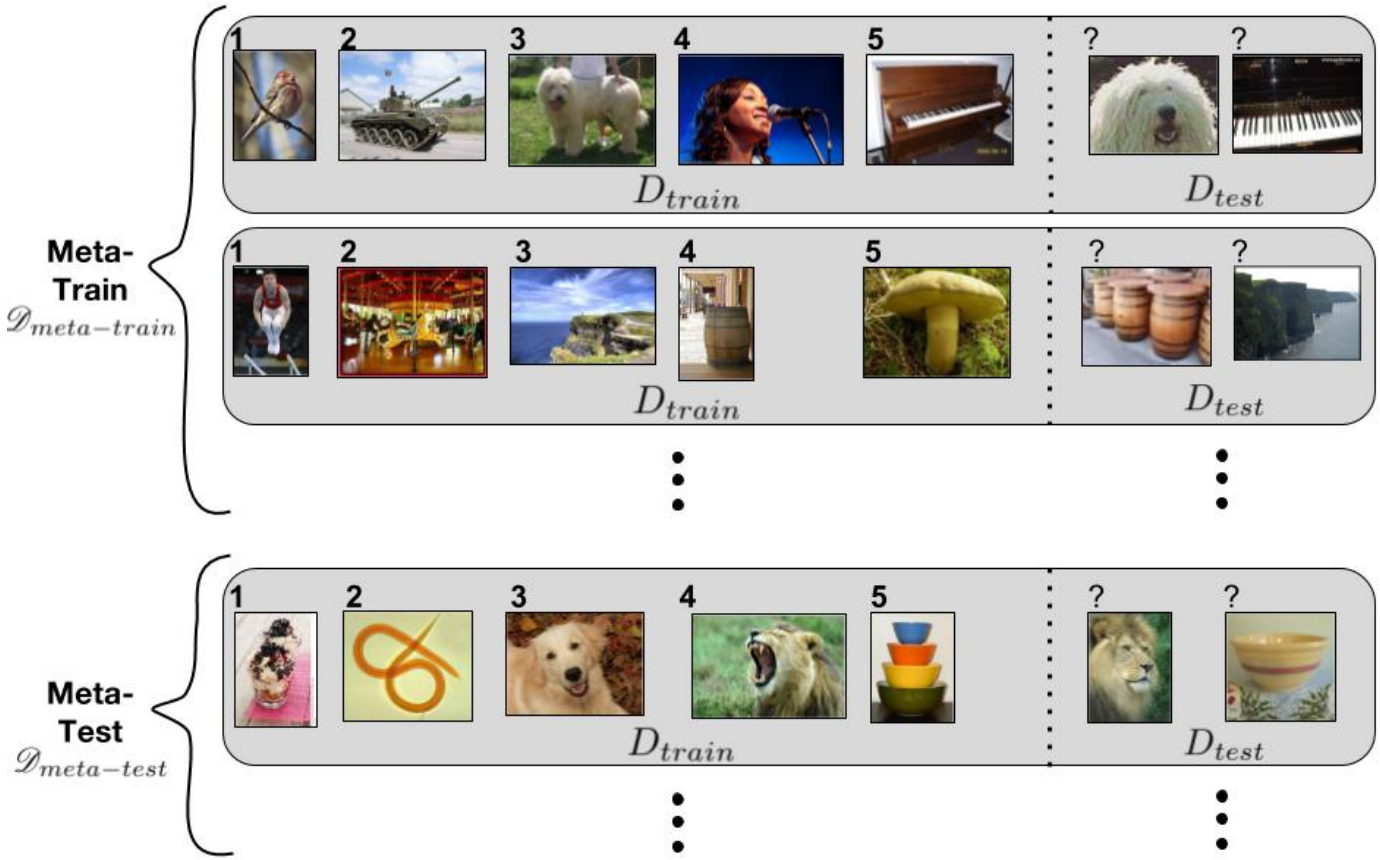
few-shot image classifier
Vinyals et al. '16



learning to quickly navigate new mazes
Duan et al. '16

Meta Learning

- Meta learning set-up for few-shot image classification[2]



[2] S. Ravi and H. Larochelle. Optimization as a Model for Few-Shot Learning. ICLR 2017.

Transfer/Multi-task/Meta Learning

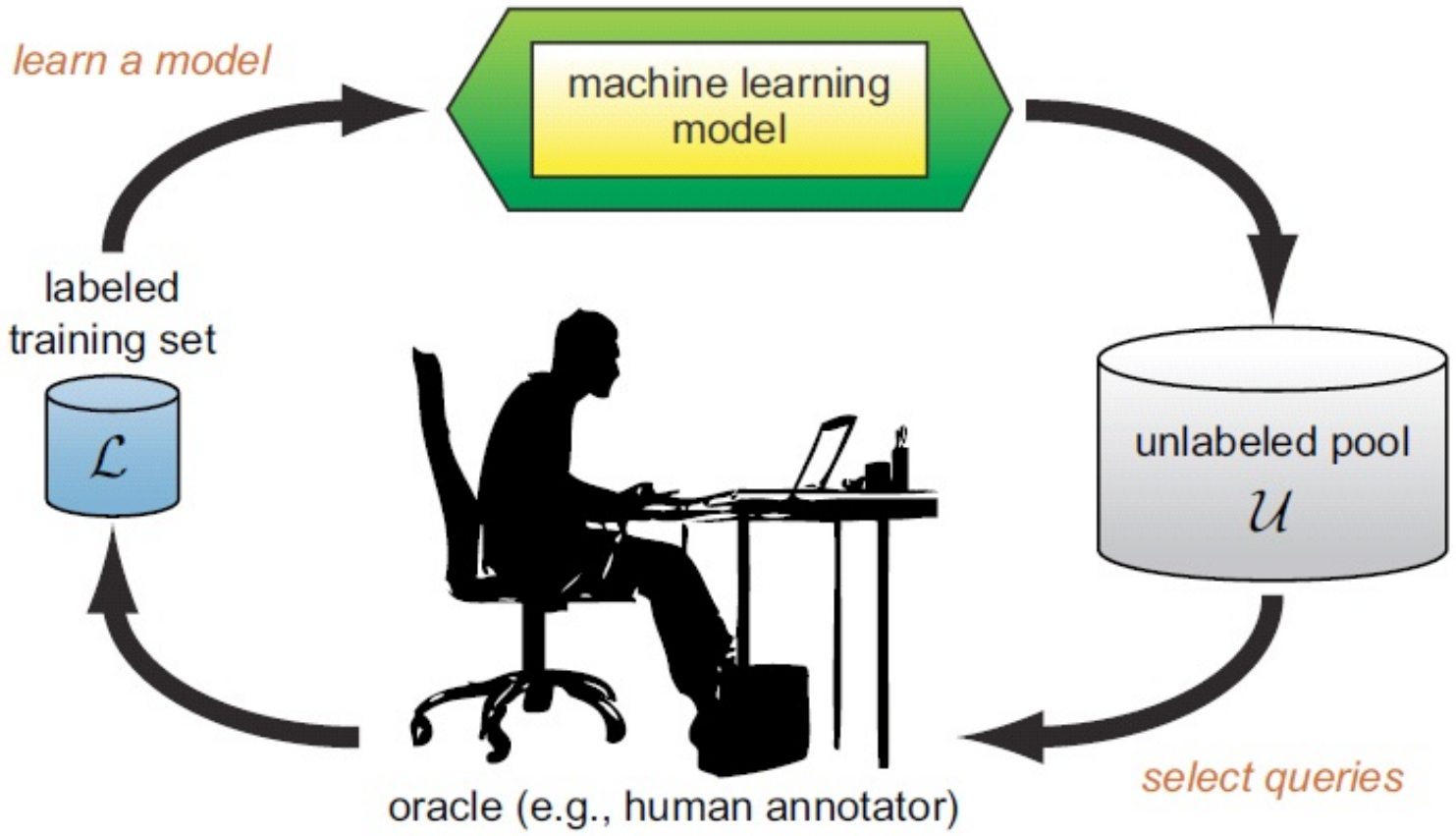
- Comparison of different learning settings:
  - Transfer learning: studies transfer from scratch.
  - Multi-task learning: assumes training and testing examples follow the same distribution
  - Lifelong learning: online meta learning

Methods	Training	Test
Multi-task learning	$T_1, T_2, \ldots, T_N$	$T_1, T_2, \ldots, T_N$
Transfer learning	T_1	T_2
Meta learning	$T_1, T_2, \ldots, T_N$	T_{N+1}
Lifelong learning	$\underrightarrow{T_1, T_2, \ldots, T_N}$	T_{N+1}
Continual learning	$\underrightarrow{T_1, T_2, \ldots, T_N}$	$(T_1, T_2, \ldots, T_N), T_{N+1}$

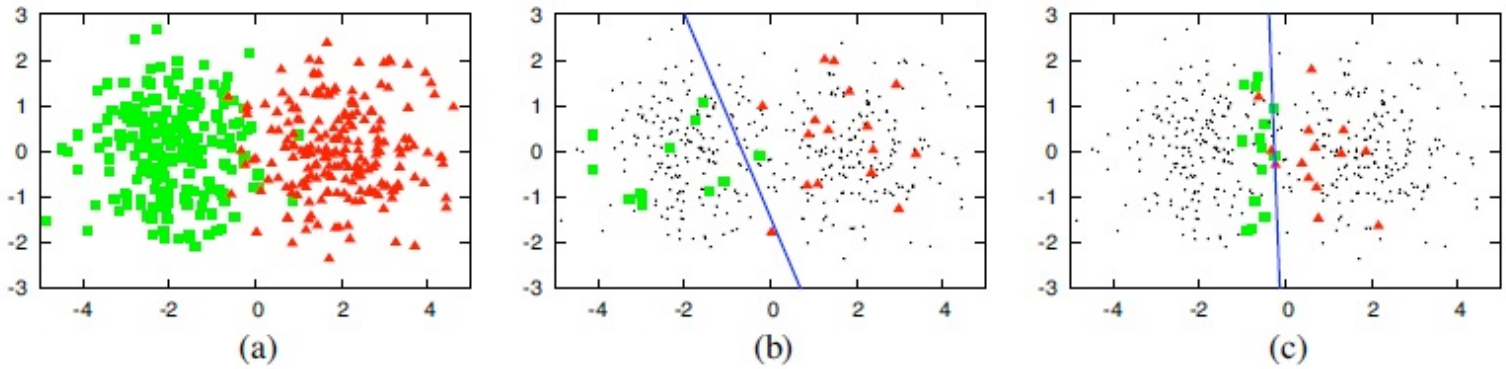
Active Learning

- Active learning is also known as query learning or optimal experimental design (in statistics).
- Active learning is well motivated in many applications in which unlabeled data may be abundant but labels are difficult, time-consuming or expensive to obtain.
- Key hypothesis:
  If the learning algorithm is allowed to actively choose the data from which it learns (instead of receiving the training data passively), it can achieve higher accuracy with less training data.
- An active learner may ask queries in the form of unlabeled instances to be labeled by an oracle (e.g., a human annotator).

Active Learning Cycle



Active Learning - A Simple Example



- (a) A toy data set of 400 instances, evenly sampled from two Gaussian classes.
- (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem domain, with accuracy 0.7.
- (c) A logistic regression model trained with 30 actively queried instances using uncertainty sampling, with accuracy 0.9.

Generative vs. Discriminative Learning

- Goal: wish to learn $f : x \to y$, e.g., $P(y|x)$.
- Generative learning (e.g., Naive Bayes)
  - Assume some functional form for $P(x|y), P(x)$. This is a "generative" model of the data.
  - Estimate parameters of $P(x|y)$, $P(x)$ directly from training data.
  - Use Bayes rule to calculate $P(y|x)$.
- Discriminative learning
  - Directly assume some function form for $P(y|x)$.This is a "discriminative" model of the data.
  - Estimate parameters of $P(y|x)$ directly from training data.
- Instance-based learning
  - A special case of nonparametric learning

Parametric vs. Nonparametric

- Parametric:
  - Data distribution follows a parametric model, e.g., Gaussian.
  - The number of parameters is independent of the sample size $|\mathcal{X}|$.
- Semiparametric:
  - Data distribution is represented by a small number of local parametric models.
- Nonparametric:
  - The data speaks for itself.
  - A nonparametric model is associated with an infinite set of parameters.
  - The number of stuff we need to keep to represent the hypothesis grows linearly with the size of the training set.
- The model flexibility increases (and hence the model bias decreases) from parametric to semiparametric to nonparametric methods.

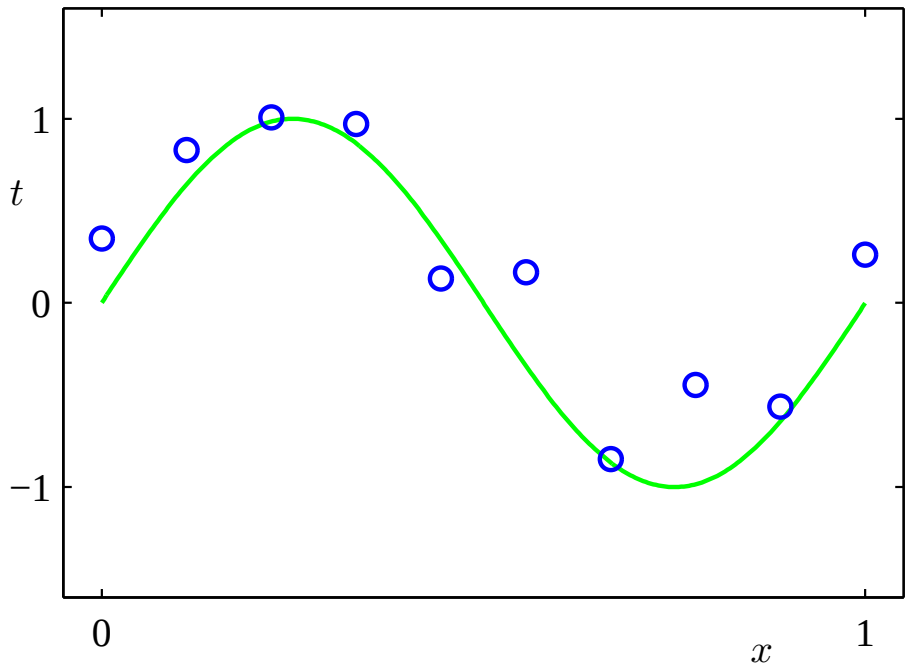
Nonparametric Bayesian Methods

- Probabilistic graphical models are parametric Bayesian models.
- Bayesian methods are most powerful when the prior distribution adequately captures our belief.
- Inflexible models (e.g., mixture of 3 Gaussian components, polynomial of order 3) sometimes yield unreasonable inferences.
- Nonparametric Bayesian models provide a way of getting very flexible models.
- Many nonparametric Bayesian models can be derived by starting with a finite parametric model and taking the limit as the number of parameters goes to ∞.
- Common Nonparametric Bayesian Methods: Gaussian Process, Dirichlet Process

Polynomial Curve Fitting

- Regression problem:
  - Input variable x
  - Target variable t
- Data generation:

$$t = \sin(2\pi x) + noise$$



Polynomial Function as Linear Model

- Polynomial function for fitting data:

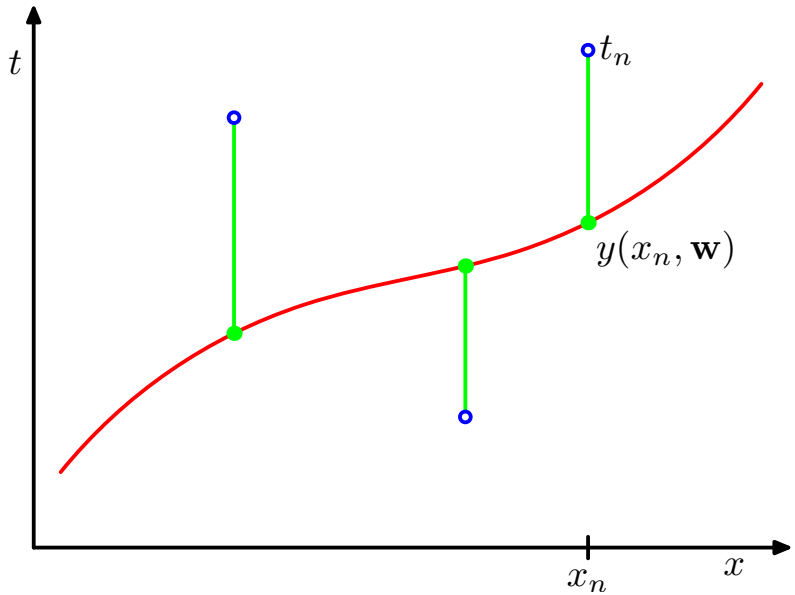
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

  where $\mathbf{w} = (w_0, \ldots, w_M)^T$ and M is the order of the polynomial.
- Linear model: The function $y(x, \mathbf{w})$ is nonlinear in x (if $M > 1$) but linear in $\mathbf{w}$.

Curve Fitting via Error Minimization

- Error function:

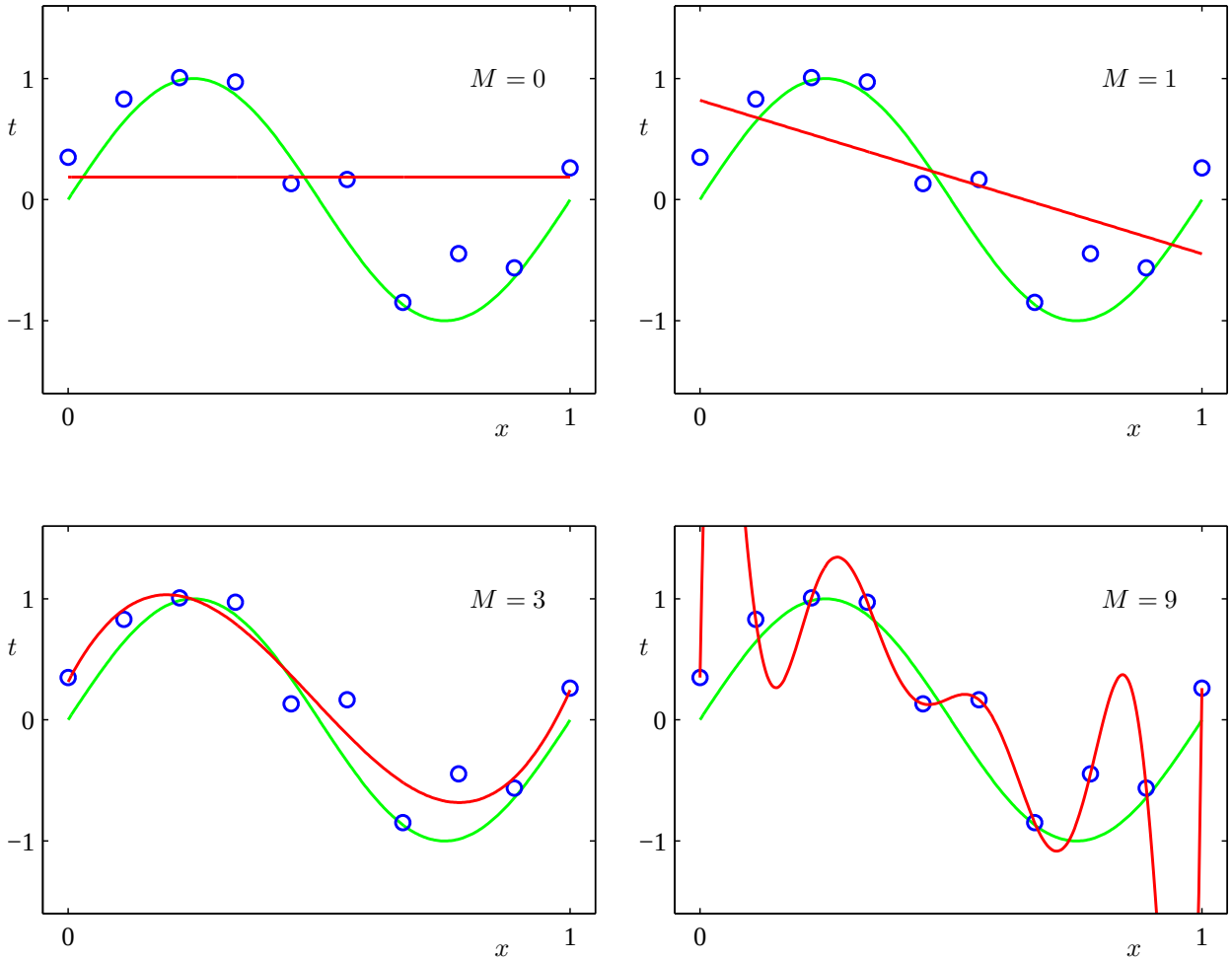
$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y(x_i, \mathbf{w}) - t_i)^2$$



- Optimal solution $\mathbf{w}^*$ that minimizes $E(\mathbf{w})$ can be found in closed form.

Order of Polynomial

- Choosing the order is an example of model selection.

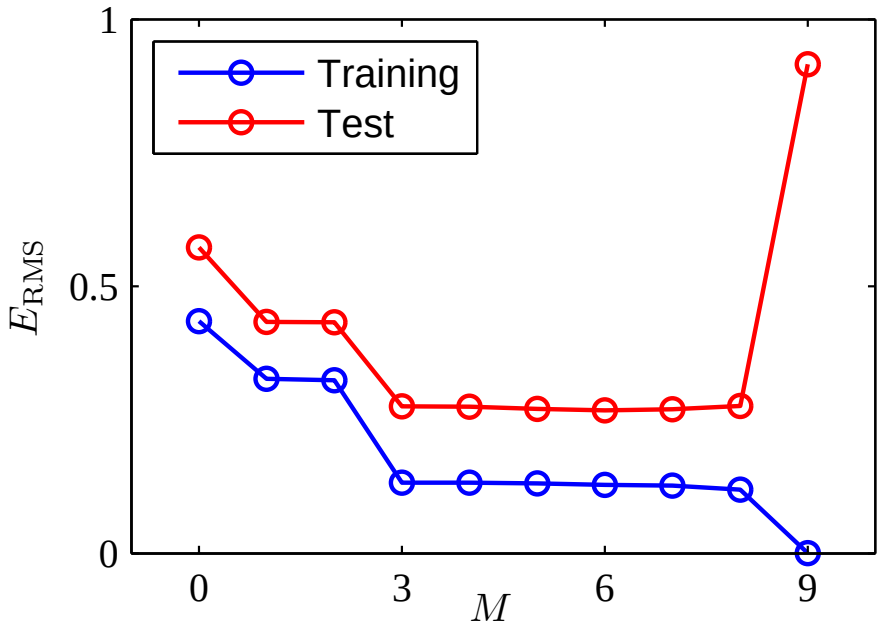


Overfitting

- Root-mean-square (RMS) error:

$$E_{RMS} = \sqrt{2E(\mathbf{w}^*)/N}$$

- Overfitting occurs at $M = 9$:

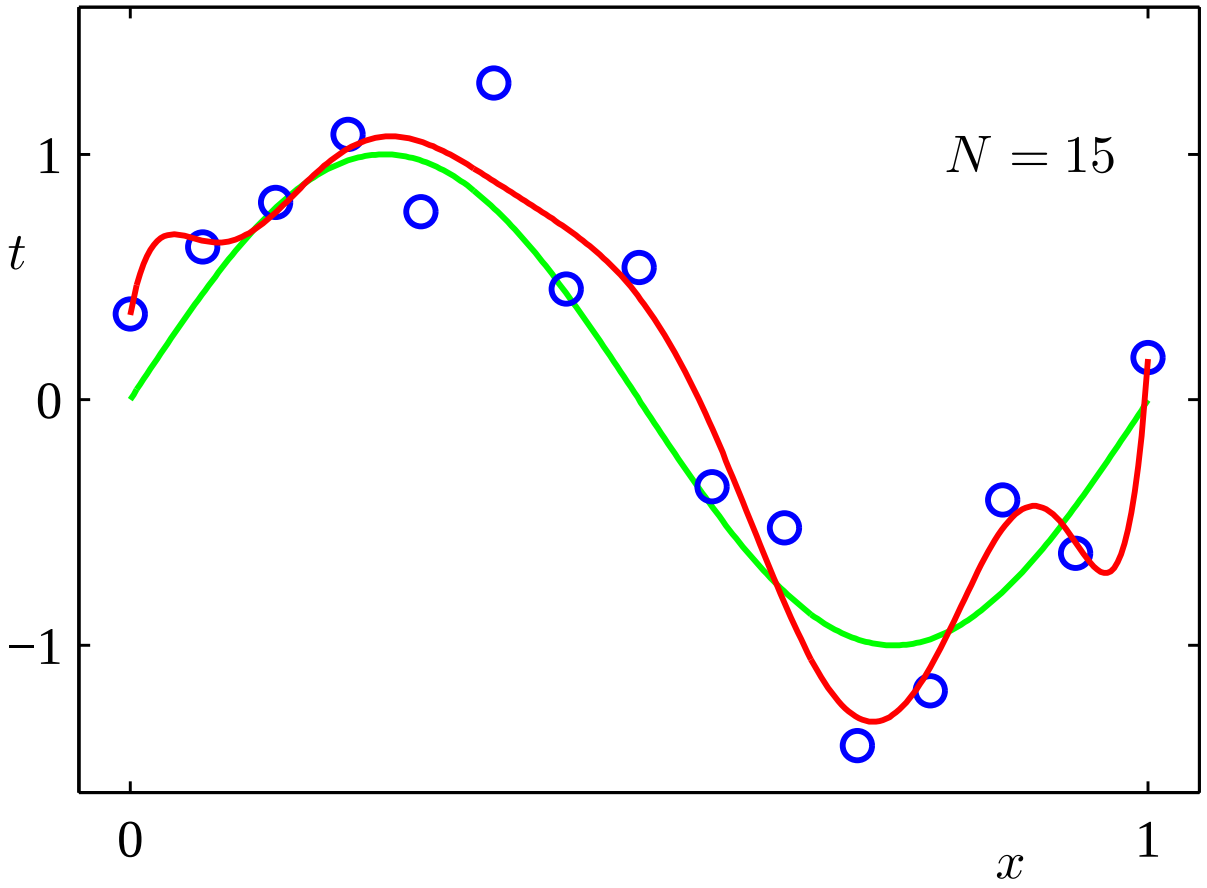


Polynomial Coefficients

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

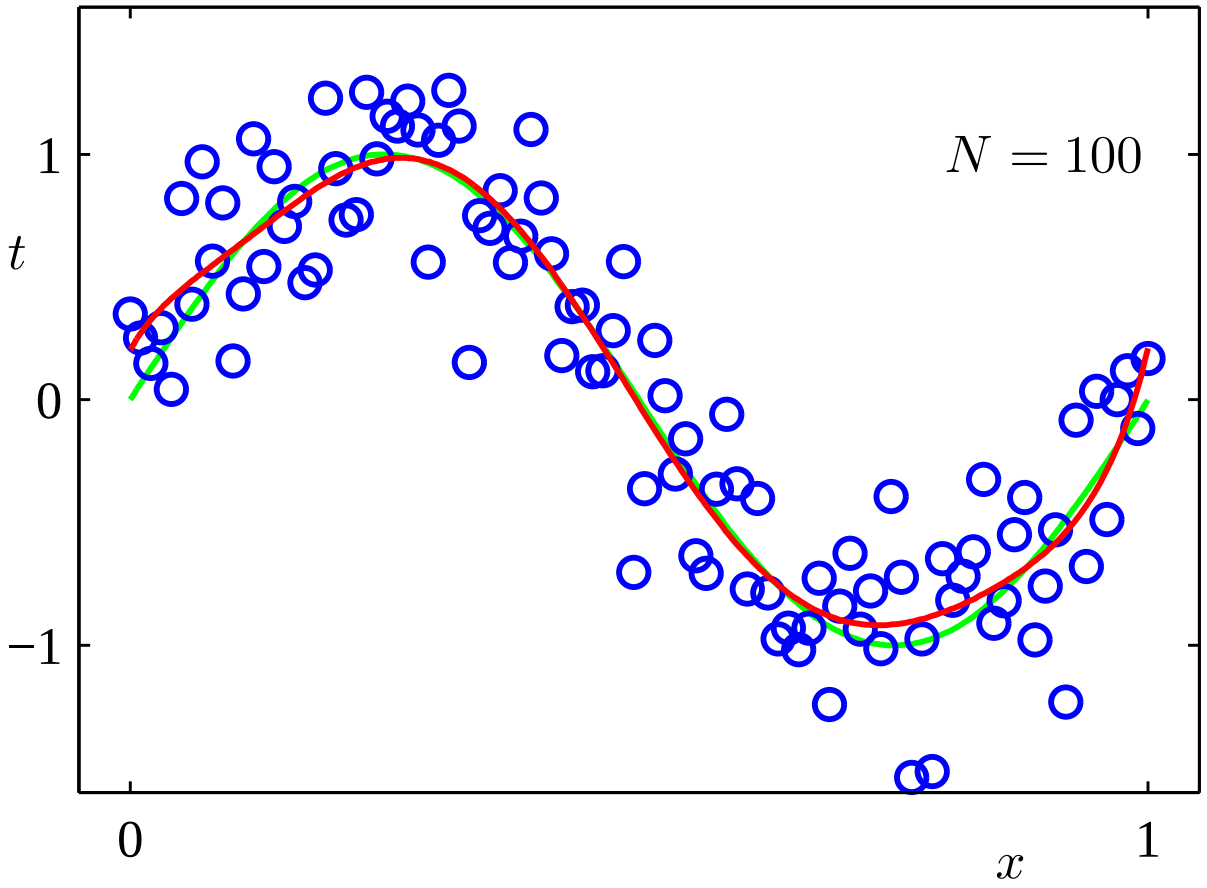
Sample Size $N = 15$

- Increasing the size of the data set reduces the over-fitting problem.



Sample Size $N = 100$

- Increasing the size of the data set reduces the over-fitting problem.



Main Reference Books

- Christopher M. Bishop (2006). Pattern Recognition and Machine Learning. Springer.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman (2001). The Elements of Statistical Learning. Springer.
- Richard O. Duda, Peter E. Hart, and David G. Stork (2001). Pattern Classification. Wiley.
- Zhi-Hua Zhou. Machine Learning (in Chinese). 2015.