E2 236
Lecture 1: Machine Learning Landscape

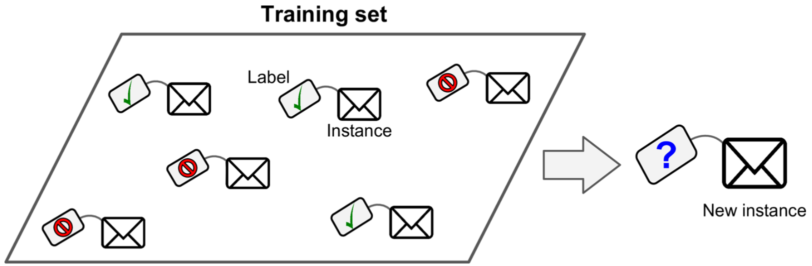
Sundeep Prabhakar Chepuri
Email: spchepuri@iisc.ac.in

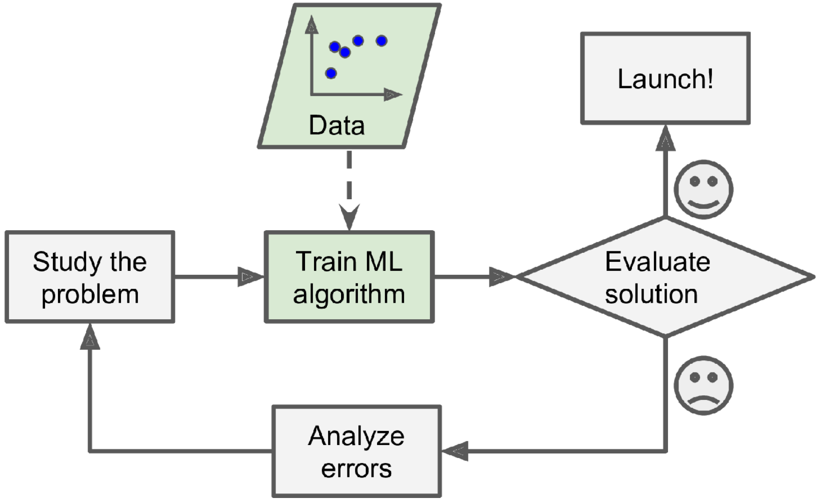
Reading: MLPP Ch. 1, UML Ch. 1, HML Ch. 1

Indian Institute of Science
भारतीय विज्ञान संस्थान

What is machine learning?

- Machine learning is the science (and **art**) of programming computers so they can learn from data.
- Your *spam filter* is a machine learning program that, given examples of spam emails (e.g., flagged by users) and examples of regular (nonspam, also called "ham") emails, can learn to flag spam.





A formal learning model

Learner has access to

- **Domain set:** An arbitrary input set or the instance space $\mathcal{X}$.
- **Features, attributes, or covariates:** Points from the domain set are instances or examples.

  E.g., represented by vectors $\mathbf{x}_i \in \mathbb{R}^D$.

  N such vectors or examples are used by the learner.

  Features are collected in the **design matrix** $\mathbf{X} \in \mathbb{R}^{N \times D}$

  Generally, features can be images, text, time-series, graph etc.

- **Label set**: Each instance may have an associated label y_i from the label space $\mathcal{Y}$

Machine learning scenarios: supervised learning

Predictive or supervised learning

The goal is to learn a predictor, a hypothesis, or a classifier

$$h : \mathcal{X} \to \mathcal{Y}$$

given a labeled set of input-output pairs or the **training set** $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$

- **Classification or pattern recognition:** The response variable y_i is categorical from a finite set, such as $\{0, 1\}$ or $\{1, \cdots, C\}$
  - Binary classification if $C = 2$ and multiclass classification if $C > 2$
  - Multi-label classification: if class labels are not mutually exclusive (email: ham/spam, urgent/not urgent)
- **Regression:** The response variable y_i is real-valued

Function approximation

- Assume some correct labeling function $f : \mathcal{X} \rightarrow \mathcal{Y}$ so that, for all i

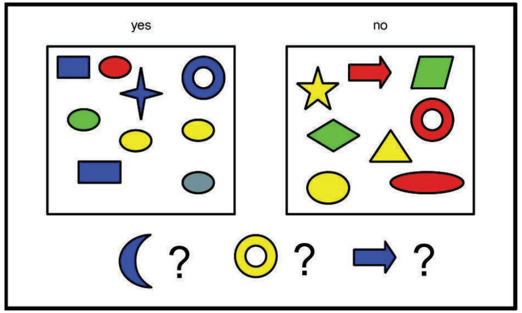
$$y_i = f(\mathbf{x}_i)$$

- This is unknown to the learner, and is precisely what the learning is trying to figure out to make predictions as

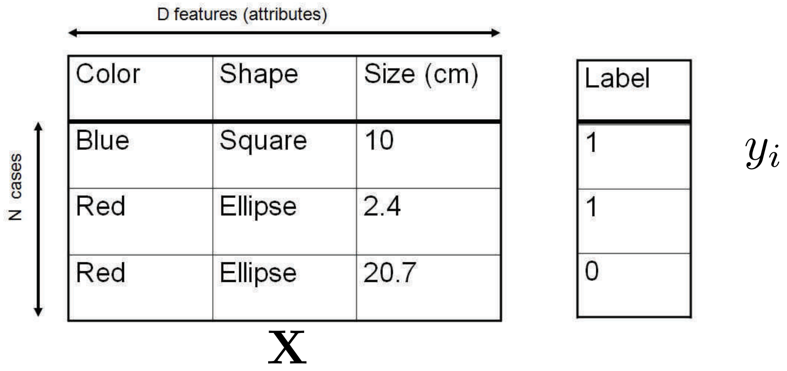
$$\hat{y}_i = \hat{f}(\mathbf{x}_i) = h(\mathbf{x}_i)$$

- Of course, the goal is to make predictions on novel inputs (unseen instances), aka **generalization**

Classification example



Cartoon labeled training examples of colored shapes, along with 3 unlabeled test cases. Image from MLPP.


D features (attributes)
N cases
$\mathbf{X}$
y_i

Color	Shape	Size (cm)	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0

As a design matrix and labels. Image from MLPP.

Predicting test cases requires **generalization** beyond training set

Blue crescent may be “yes”, but the other two are hard to guess. So, it is desirable to return a probability.

Probabilistic prediction

- Denote the probability distribution over possible labels, given the input vector $\mathbf{x}$ and training data $\mathcal{D}$

$$p(y|\mathbf{x}, \mathcal{D})$$

- This is generally a vector of length C, but suffices to return a number for binary classification

- We can compute our “best guess” as to the true label as the most probable class label

$$\hat{y} = h(\mathbf{x}) = \hat{f}(\mathbf{x}) = \mathrm{argmax}_{c \in \{1,\ldots,C\}}\ p(y|\mathbf{x}, \mathcal{D})$$

This is called the **MAP (maximum a posteriori) estimate.**

Measure of success, risk, and empirical risk

- Let us denote the probability distribution over $\mathcal{X}$ by $\mathcal{S}$

- The error of h is the probability to draw a random instance $\mathbf{x} \sim \mathcal{S}$ such that $h(\mathbf{x}) \neq f(\mathbf{x})$

$$L_{\mathcal{S},f}(h) := \mathbb{P}_{\mathbf{x}\sim\mathcal{S}}[h(\mathbf{x}) \neq f(\mathbf{x})]$$

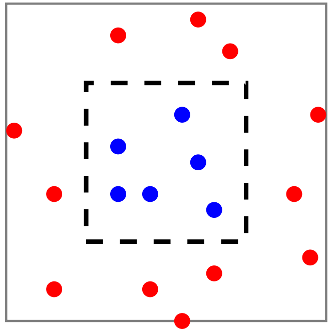
This is called the **generalization error**, **risk** or true error of h

Since the learner does not know $\mathcal{S}$ or f, learner cannot calculate the true error. But can compute the error incurred over the training set $\mathcal{D}$

$$L_{\mathcal{D}}(h) := \frac{\{i \in [N] : h(\mathbf{x}_i) \neq y_i\}}{N}$$

This is called the **empirical error or empirical risk** of h

Overfitting or memorizing the training samples



Areas of the **gray square** has area 2 and the **inner blue square** has area 1

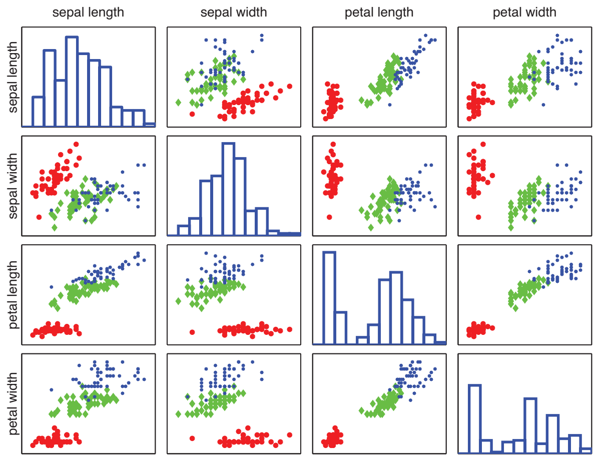
- Probability distribution $\mathcal{S}$ is such that instances are distributed uniformly within the gray square
- Labeling function, f, determines the label to be 1 if the instance is within the inner blue square, and 0 otherwise.

Suppose we use a predictor $h_{\mathcal{D}}(\mathbf{x}) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x, \\ 0 & \text{otherwise.} \end{cases}$ with $L_{\mathcal{D}}(h) = 0$

The true error with finite instances is, $L_{\mathcal{S},f}(h) = \mathbb{P}\big(h(x) \neq f(x)\big) = \mathbb{P}\big(f(x) = 1\big) = \dfrac{1}{2}.$

Classification examples

➢ Iris flowers: Setosa, Versicolor, and virginica



Feature extraction:
Features chosen by humans

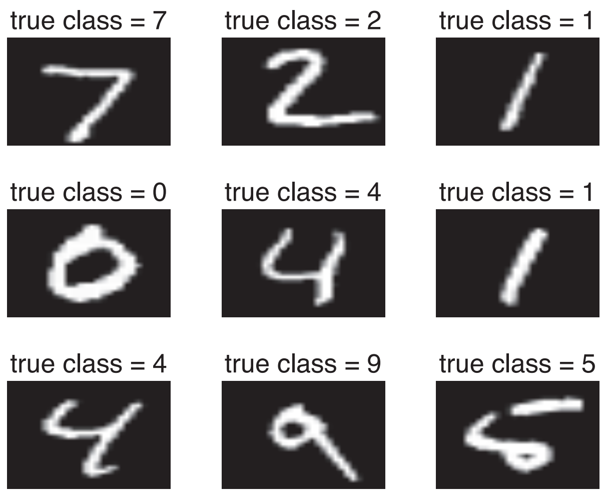
Images from MLPP

Hypothesis space:

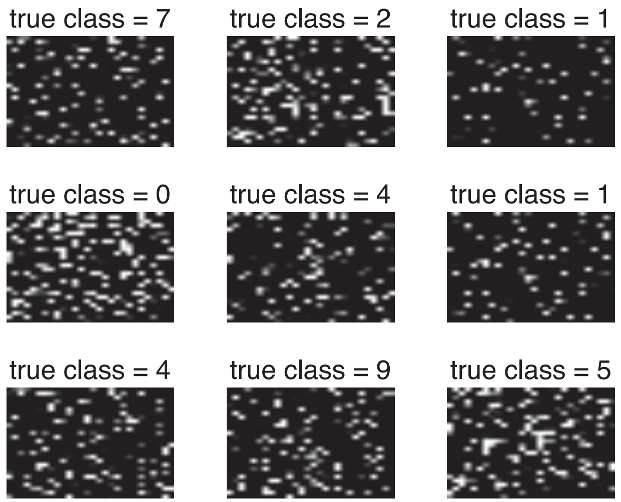
- 4 features and 3 class labels
- Assume, binary (0/1) features, we get 2^4 = 16 feature combinations and 3^16, i.e., about 430k potential combination of rules. This is the hypothesis space.

Classification example

MNIST digits



Features: 28×28 grayscale values in the range 0:255

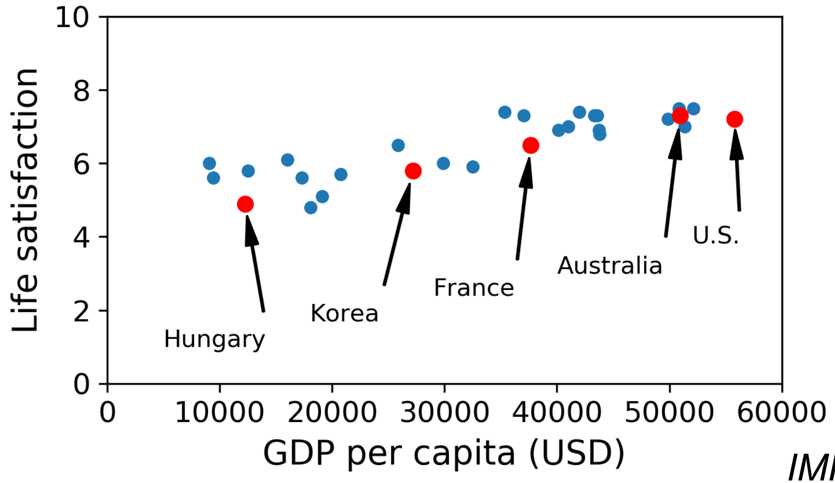


features permuted randomly

Generic classification methods ignore any structure in the input features, such as spatial Layout. Consequently, they can easily handle data that looks like picture in the right

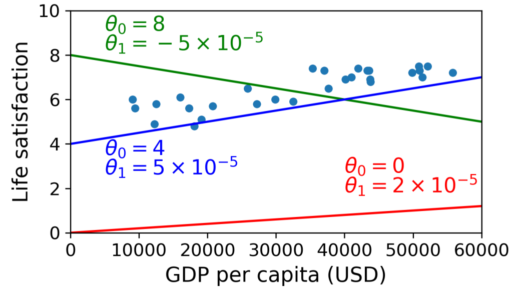
Regression example

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

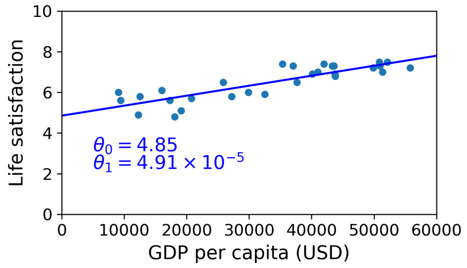


IMF data

Simple linear model: $\text{life\_satisfaction} = \theta_0 + \theta_1 \times \text{GDP\_per\_capita}$



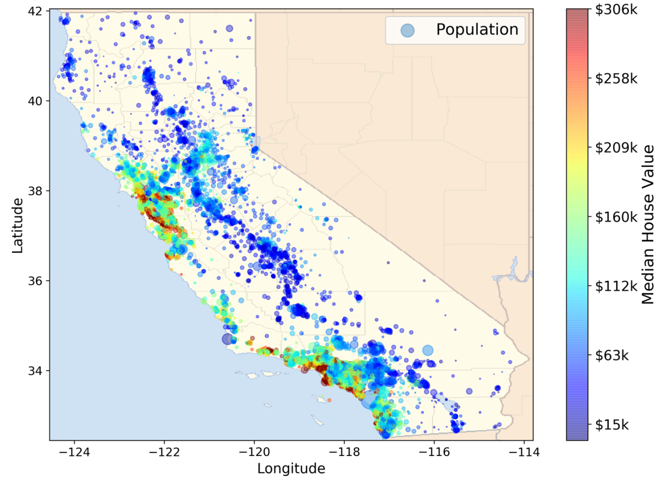
Many possible linear models



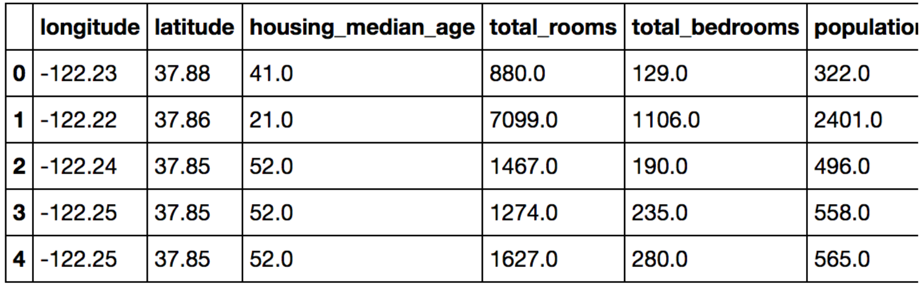
Model that fits the training data the best

Another regression example

- House price prediction



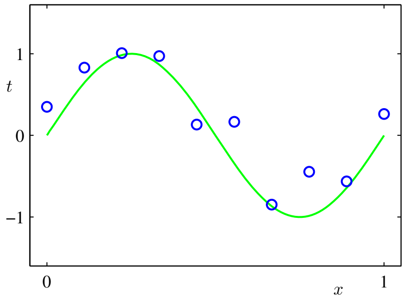
```
longitude
latitude
housing_median_age
total_rooms
total_bedrooms
population
households
median_income
median_house_value
ocean_proximity
```



	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	populatio
0	-122.23	37.88	41.0	880.0	129.0	322.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0

Parametric models

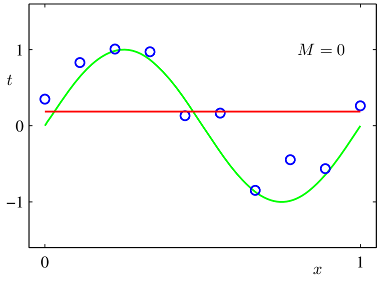
Polynomial curve fitting

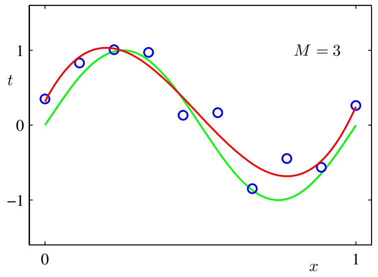


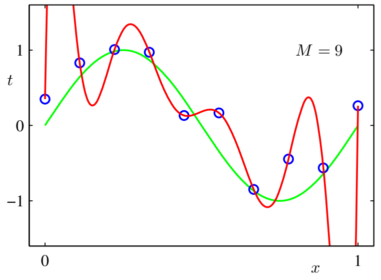
Parametric model: $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$

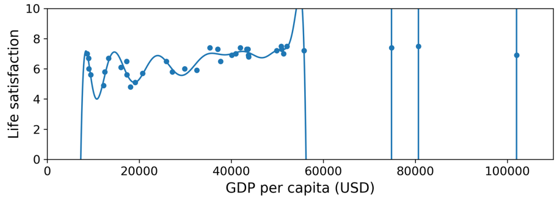
Error function: $E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$

Model selection:









underfitting

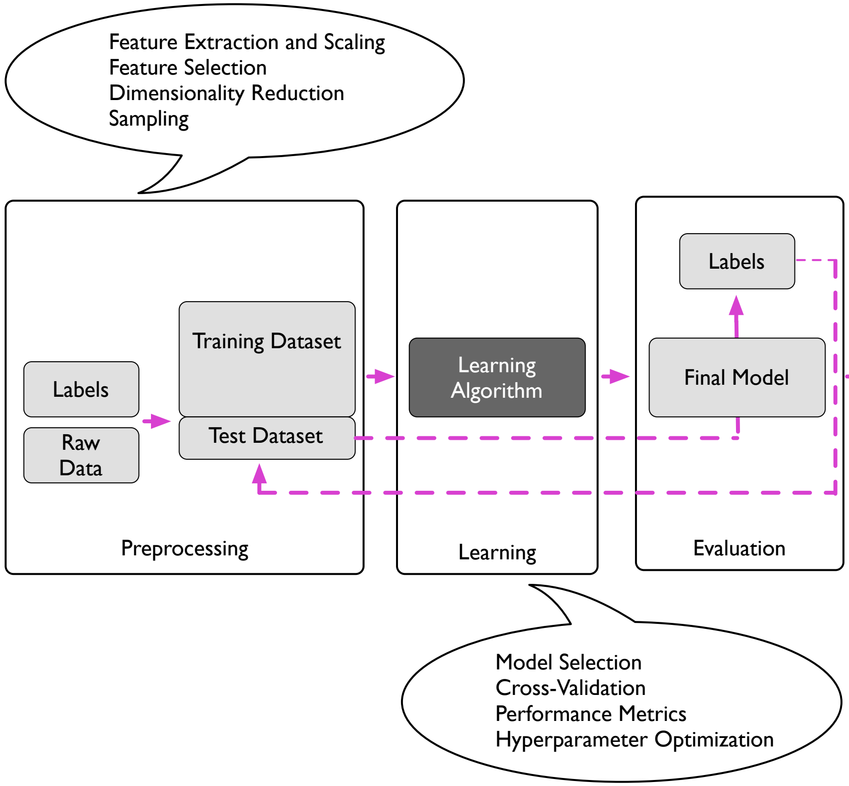
overfitting

Images from PRML

Hyperparameter tuning and model selection

- How do you select the model parameters?
  - best hyperparameter value that produces a model with the lowest generalization error
- **Holdout validation and Validation set:**
  - hold out part of the training set to evaluate several candidate models and select the best one
  - validation set is too small, then model evaluations will be imprecise:
    - you may end up selecting a suboptimal model by mistake
  - validation set is too large, then the remaining training set will be much smaller than the full training set
- **Cross-validation**: Use many small validation sets and average out all model evaluations.
  - Disadvantage: multiple training needed

ML pipeline for supervised learning



Stat 479, UWM, Sebastian Raschka

Machine learning scenarios: unsupervised learning

Descriptive or unsupervised learning

The goal is to find interesting patterns in data given inputs $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

➢ Less well-defined as problem as oftentimes it what kind of patterns we are looking for is unclear

- Clustering

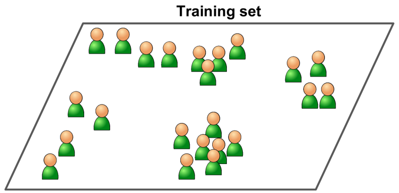
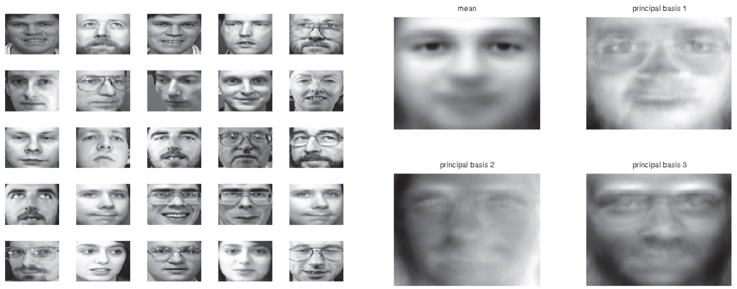


Image from HML

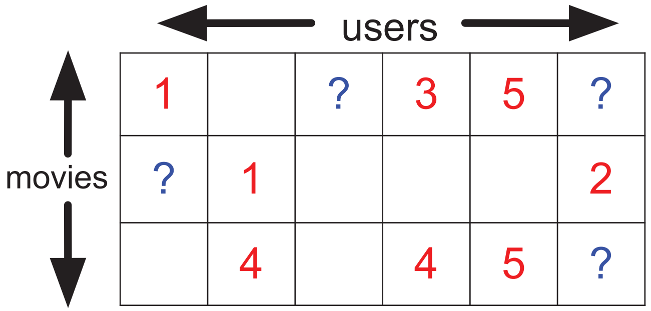
- Dimensionality reduction (PCA, CCA)



Left: 25 randomly chosen 64 ×64 pixel images from the Olivetti face database.
Right: The mean and the first three principal component basis vectors (eigenfaces).
Image from MLPP

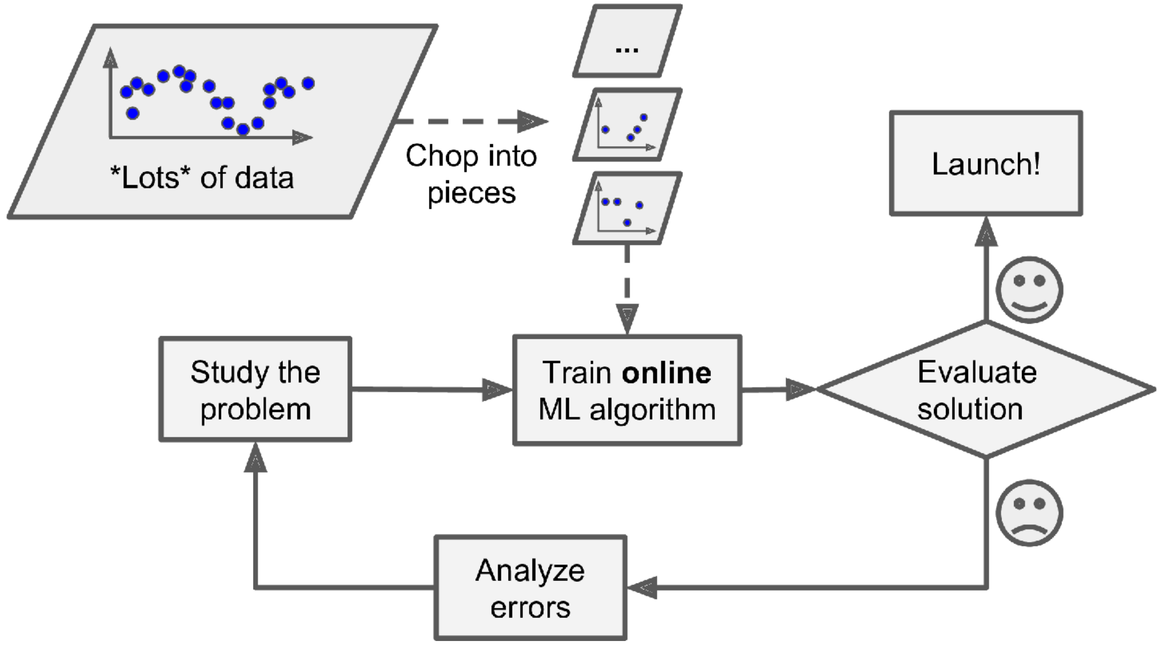
Matrix completion or collaborative filtering

Imputation problem: Predicting which movies people will want to watch based on how they, and other people, have rated movies which they have already seen.



1		?	3	5	?
?	1				2
	4		4	5	?

Mini batch vs. full-batch learning



Images from HML