

Chapter1. The Machine Learning Landscape

Types of ML

按有无监督分类：

1. Supervised Learning

- Classification
- Target numeric value prediction

Note: 一些回归算法可以用作分类，如logistic regression

2. Unsupervised Learning

- Clustering
- Anomaly detection and novelty detection 异常和新颖检测
 - Anomaly detection 异常检测：发现异常值
 - Novelty detection 新颖检测：发现原本数据集中不能被发现了的新实例
- Visualization and dimensionality reduction
 - 简化数据的同时最小化信息的损失，或称feature extraction
- Association rule learning 关联规则学习
 - 学习和发现大型数据库中变量之间的有意义关系的技术

3. Semisupervised Learning

处理有大量unlabeled sample、少量labeled sample的数据集

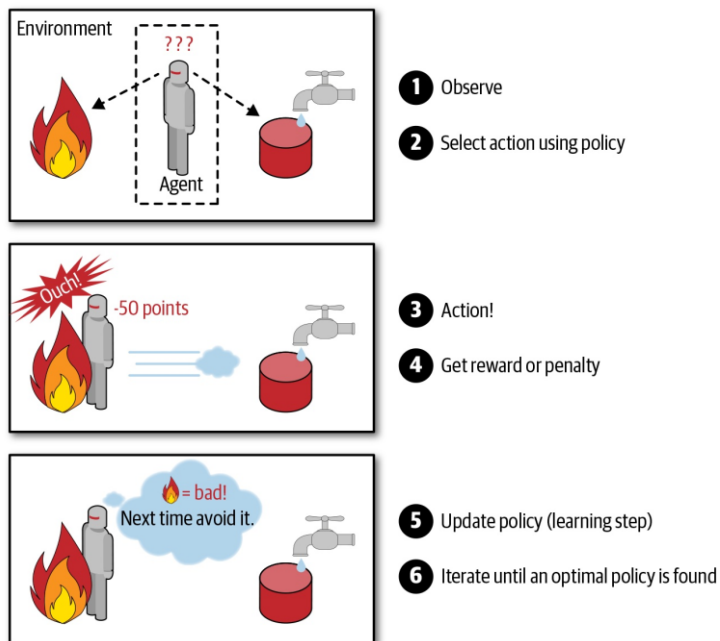
大部分的半监督学习算法=有监督学习+无监督学习

E.g deep belief networks(DBNs) 由 restricted Boltzmann machines (RBMs, 受限玻尔兹曼机) 堆叠而成，再使用监督学习算法进行微调。

4. Reinforcement learning

Agent, rewards/penalties, policy/strategy

Agent感知环境进行活动，对环境做出选择后得到相应的rewards/penalties，多次学习后得到最佳的policy/strategy



E.g: 机器人, AlphaGo(通过数百万次的游戏学习最佳的策略, 再自己和自己下棋)

按是否在线学习分类:

Batch and Online Learning

1. Batch learning(offline learning)

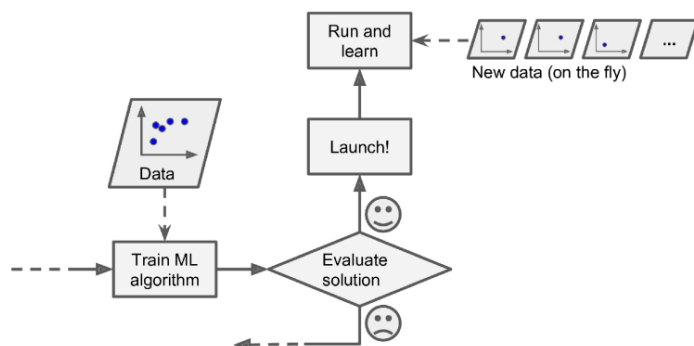
系统不能根据不断增长的数据进行学习

缺点:

- 对硬件要求高
- 目前的数据不足以训练模型 (如某个app的用户量)

2. Online learning

不断的输入新的数据(mini-batches), 如股票价格预测。



优点:

- 可以消耗更少的硬件资源, 如disk space, CPU等
- 可以抛弃之前的数据, 节省空间

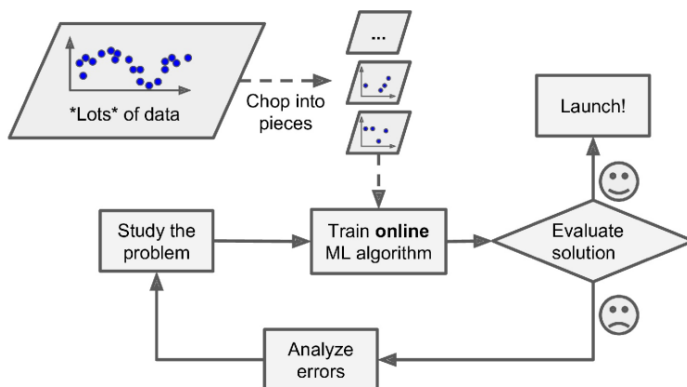
其他用处, 比如很大的数据集每次可以拆分成小的子集, 不断feed in, 但是在offline的情况下进行模型训练, 一般称为**out-of-core learning**. (为避免和online learning混淆,

一般称作incremental learning)

online learning的一个重要参数是**learning rate**:

- 太大的learning rate, 模型很快适应新的data, 但会很快丢失通过旧数据训练得到的模型。结果是只适应最新的数据。
- 太小的learning rate, 模型会变得懒惰, 学习很慢, 但同时对新数据中的噪声或异常值不敏感。

Online learning to handle huge datasets



Challenge on online learning

新的数据可能会导致模型质量下降, 解决办法是终止学习或回滚版本, 或对输入数据进行监控 (如异常检测等)。

按泛化方式分类:

Instance-based learning

new cases通过计算和learned examples的相似度生成

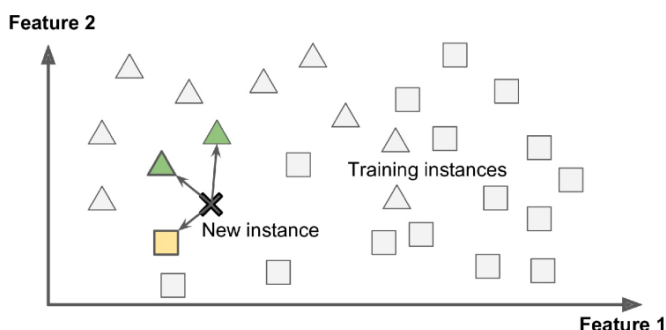
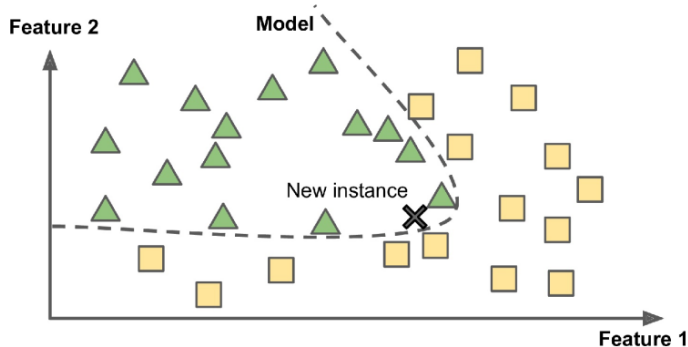


Figure 1-15. Instance-based learning

new instance和三角形的相似度 (可能是距离) 更近, 因此new instance被判为三角形

Model-based learning

通过一系列的sample生成一个模型，再做出predictions



一般流程:

- 观察数据
- 选择模型
- 训练模型
- 预测数据

Main Challenges of ML

- Insufficient Quantity of Training Data。数据量可能是影响一个模型性能的重要因素
- nonrepresentative training data。某些数据值的缺失。
 - Sampling noise: 数据集太小时，nonrepresentative data是偶尔产生的
 - Sampling bias: sampling method有缺陷时，可能也会产生nonrepresentative samples
- Poor-quality data。errors, outliers, noise导致模型很难发现潜在的pattern。因此，清理数据是很重要的。
 - 舍弃明显的outliers或手工修复errors。
 - 处理缺失值，舍弃样本或补充缺失值（使用中值），或训练两个模型（针对舍弃缺失值样本和包括缺失值样本）。
- Irrelevant Features。选择合适的features - feature engineering:
 - Feature selection, 选择最有用的features。
 - Feature extraction, 组合现有的一些features来得到新的features, e.g. PCA。
 - 通过收集新数据创建新的features。
- Overfitting the Training Data。模型参数太多，太复杂，导致泛化能力太差。
 - 简化模型，减少参数数量。

- 收集更多数据。
- 减少noise, fix errors, remove outliers.

E.g. Regularization:

对于存在两个参数 θ_1 和 θ_2 的线性模型，拥有二阶自由度（截距与斜率）。如果将其中一个固定为0（ θ_1 ），则变为一个自由度。如果将 θ_1 保持在一个很小的值，则自由度可以在1到2之间，则模型比两个参数的简单，比一个参数的复杂。因此，可以引入正则化项，这些参数称为hyperparameter。Hyperparameter是属于算法本身的参数，不属于模型参数，即不会因为训练过程的进行而变化。并且需要在训练前设置好参数值。

- Underfitting the Training Data。与Overfitting相反，原因是模型太简单。
 - 增加参数，选择更合适的模型。
 - 选择更好的features (feature engineering)
 - 减少模型的约束 (e.g. regularization hyperparameter)

Testing and Validating

- Split into **training set** and **test set**. 评价模型的好坏指标：
 - Generalization error (out-of-sample error) - The error rate on new cases
 - Traing error 低, generalization error 搞 - overfitting
- 一般是80%用作training set, 20%用作test set。但取决于数据集的大小。对于1000万的数据集，1%作为test set就已经足够。


Hyperparameter Tuning and Model Selection

使用**holdout validation**实现超参数的调整和model selection:

- Validation set (development set/ dev set) full data set = training set + test set + validation set
 - 在traning set上训练多个模型
 - 选择在validation set上表现最好的模型A
 - 将表现最好的模型A用traning set + validation set训练
 - 将A在test set上进行评估
- 存在问题：
 - 如果validation set太小, model evaludation不准确, 导致选择一个次佳的模型

- 如果validation set太大，剩余的training set会比training set + validation set小很多
- 最后所有候选模型要在 training set + validation set上进行训练，但之前的训练只是在很小的training set上进行，因此会导致偏差。
- 解决方法 - **Cross Validation**
 - 将full data set分成N份，其中1份作为validation set，N-1份合并作为training set（不需要test set）
 - 缺点：原本只需训练一次，现在要训练N次，时间变长。主要目的是防止overfitting，因此数据量足够大时不需要Cross Validation。

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5



 5-fold cross validation (image credit)

Data Mismatch

尽量使validation set和test set的数据相匹配/相近：将数据打乱，一半给validation set，一半给test set，确保相似的不会在同一个set里。

假如训练使用的数据不能很好的代表实际应用环境中的数据（图片识别中，网上下载的图片 and 手机拍摄的相差很大）。

到底是overfitting还是Data Mismatch数据不匹配造成？

尝试方法：

- 将一部分从网上下载的图片从training set抽出，放到train-dev set中。
- 用training set训练模型，用train-dev set评估模型。
- 如果模型表现得好，证明没有overfitting。如果模型在train-dev set表现差，就是因为data mismatch造成的。

解决Data Mismatch可以首先将网上下载的图片处理成近似于手机拍照的图片，再重新训练模型：

- 如果在train-dev上表现很差，则是由于overfitting。可以regularize model、增加训练数据量、清洗数据。

