

1. An introduction to machine learning and deep learning

Zhengtao Wang

About the curriculum

- 1. Introduction to Machine Learning: introductions, concepts, research fields, and a few examples
- 2. Supervised Learning(1): Naive Bayes, Support vector machine。
- 3. Supervised Learning(2): Decision Tree, Boosting
- 4. Unsupervised Learning: k-means, PCA, SVD
- 5. BP networks and gradient descent
- 6. Neural networks(1): principles
- 7. Neural networks(2): Layers and configurations
- 8. Neural networks(3): Structures and advanced models
- 9. Deep learning framework and applications: based on Keras

Recommended materials(MUST)

- Stanford online course: CS 229 Machine Learning, Andrew Ng.
- Stanford online course: CS 231n Convolutional Neural Networks for Visual Recognition, Fei-Fei Li, Justin Johnson, Serena Yeung.
- 廖雪峰在线Python教程
- 统计学习方法，李航，清华大学出版社

All participants MUST read/take all materials/courses during this curriculum.

Recommended materials(optional)

- LeetCode: Online OJ for general algorithms, for practicing python.
- 机器学习, 清华大学出版社, 周志华
- Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville

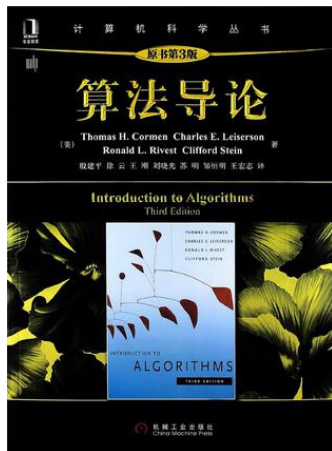
Contents

- Machine Learning: What, Why and How
- Three components in Machine learning algorithms
- Concepts:
 - Data, Dimension and Space
 - Supervised Learning and Unsupervised Learning
 - Training set, validation set and test set
 - Over-fitting and Under-fitting
 - Generative model and Discriminative model
 - Machine Learning and Deep Learning
- kNN: Lazy learning
- Logistic Regression: Discriminative model
- Homework #1

Machine Learning: What, Why and How

What: (almost) statistical inference

- Binary Search
- Quick sort
- Shortest path
- The multiplication of matrices
- ...



- Choose a potential date from candidates
- Whether to make lending to some loan applicant
- Recognize animals in a picture
- ...
- Predict the stocks based on past records
- Predict the winning number for a lottery based on past winning numbers
- ...

What: Statistical inference

- Machine learning gives computers the ability to learn without being explicitly programmed.
- Basically, Machine learning= Statistical learning
- Machine learning is especially a good choice when modelling latent relationships.
A typical case: Pattern Recognition.

Why: The core of AI

- Machine Learning is widely used in many real-world applications.
 - Here are some applications you may be aware of:
 - Automatic Driving
 - Siri
 - AlphaGo
 - And here are something you may not realize:
 - Google/Baidu/Bing search
 - Typing with modern IME
 - Prisma



Why: The core of AI

- Machine Learning is the core of AI.
 - Let Computer see: Computer Vision
 - Image classification
 - Object detection
 - Image semantic segmentation
 - Object Tracking
 - ...
 - Let Computer hear: Speech recognition
 - ...
 - Let Computer understand: Natural Language Processing
 - ...

How: Foundations

- Prerequisites
 - Mathematics(Insights are more important):
 - Matrix Theory
 - Probability and Statistics
 - Calculus(a little)
 - Optimization Theory(a little)
 - Programming:
 - Principles of Computer and Object-Oriented Programming(a little)
 - Linux OS(a little)
 - Python and Python scientific computation environment
 - At least 1 deep learning framework.
 - Machine learning theory
 - Just read all materials we recommended at the previous page

How: Research

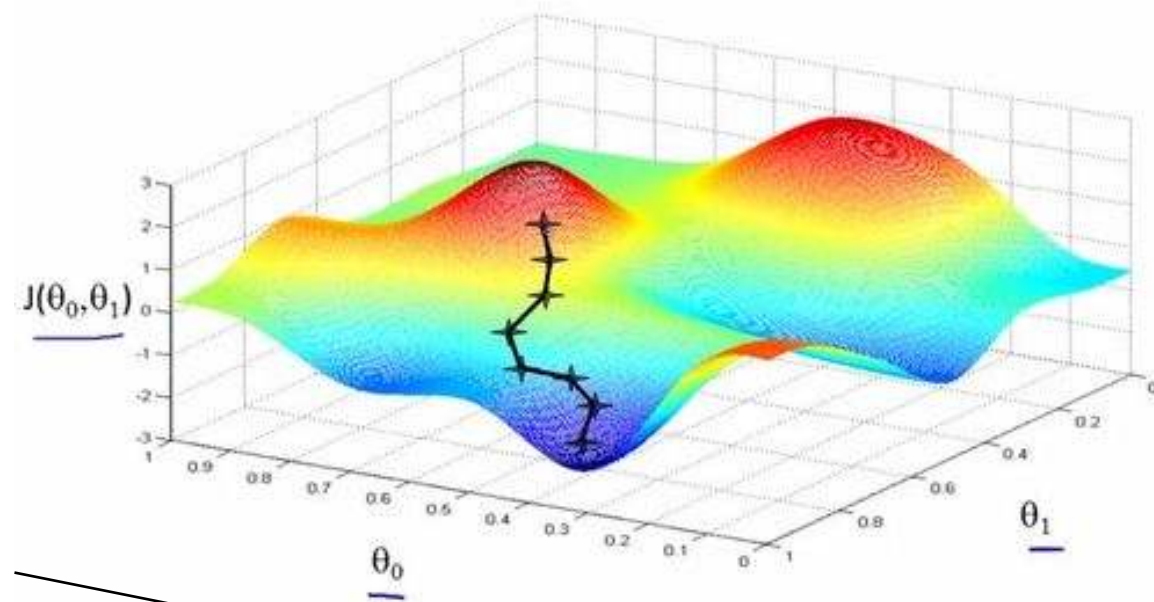
- Top conferences:
 - AAAI, IJCAI, NIPS, ICML, ICLR, CVPR
 - For computer vision researchers: ECCV, ICCV
- Newest research results:
 - arXiv: Computer Vision and Pattern Recognition
- How to:
 - Choose a research topic
 - Find the newest papers on top conferences
 - Read the papers along the reference chain.

Three components in Machine learning
algorithms

Machine learning algorithms

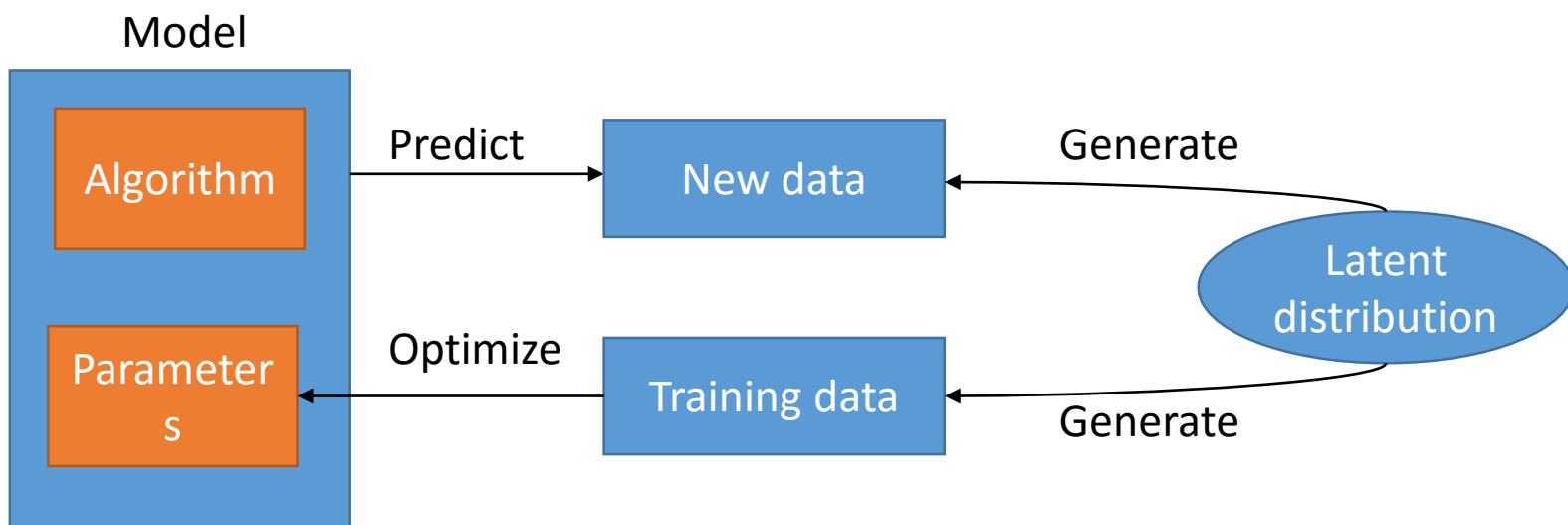
- Model: Methods and Assumptions
- Objective(Loss): The target of the model
 - Evaluate the performance of the model.
 - Guide the updating of parameters
- Optimizer: The way to the target
 - Update parameters based on training data and objective

Loss/Objective
 $L(D, \theta)$

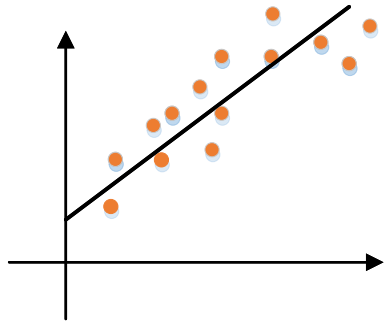


Param axis 0

Param axis 1



An example: Least Square Method

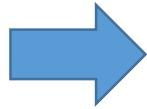


- Model
 - The data distribute around a straight line
 - $y = wx + b$
- Objective:
 - The data points should be as close to a straight line as possible
 - $\min_{w,b} \frac{1}{N} \sum_{i=1}^N (y_i - (wx_i + b))^2$
- Optimizer:
 - Convex problem
 - $w = \frac{\sum_{i=1}^m y_i (x_i - \bar{x})}{\sum_{i=1}^m x_i^2 - \frac{1}{m} (\sum_{i=1}^m x_i)^2}$
 - $b = \frac{1}{m} \sum_{i=1}^m (y_i - wx_i)$

Concepts

Data, Feature and Space

- Data: Representation of real-world objects.
 - Texts
 - Images
 - Videos
- Data -> feature(vector)
 - Feature Engineering is very important
 - But difficult
 - But we have deep learning now.
- A vector with length N is a point in N -dim space.(Data Space)
 - Each axis in N -dim space is a field.
 - Good feature: cohesion within alike points, discrimination between different points.



DIGITS

Supervised Learning and Unsupervised Learning

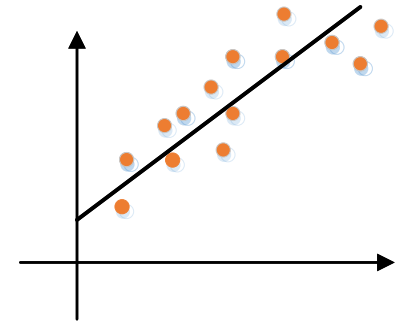
- Supervised Learning:
 - Target: classification/regression
 - Data: (X, Y)
- Unsupervised Learning:
 - Target: Find the inner structure of data
 - Data: only X
- Reinforcement learning

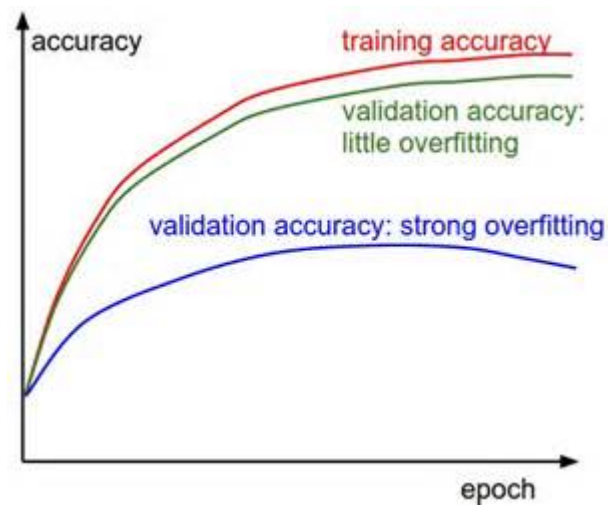
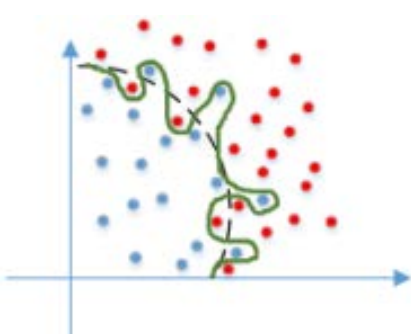
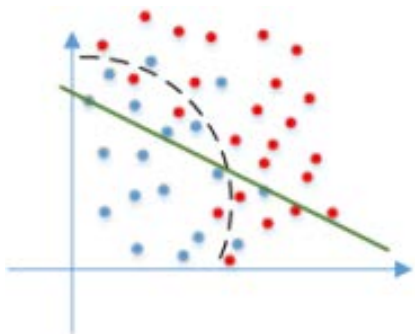
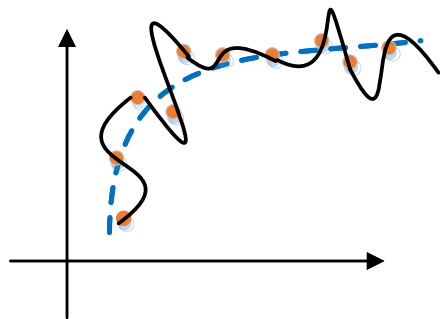
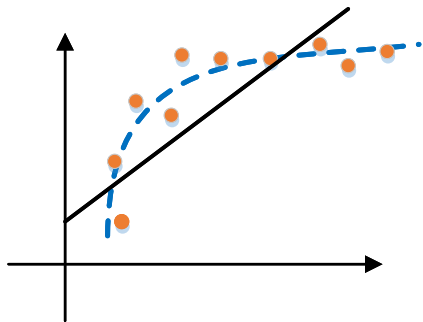
Training set, validation set and test set

- Training set:
 - Dataset for training the model -> For determining the parameters
- Validation set:
 - Dataset for validation. -> select models
- Test set:
 - Dataset for test the model -> evaluate the performance/generalization ability
- Principles:
 - Model never see validation set and test set during training
 - Performance on validation set is not trustable as you may expected, you should not adjust your parameters based on performance on validation set.
 - However, **cross-validation** is much better.

Over-fitting, Under-fitting and Generalization

- Over-fitting
 - Model is too complex, while dataset is too simple
- Under-fitting
 - The opposite case
- Under-fitting is not a problem
- You can't eliminate over-fitting, you can just try to control it.
- **Generalization** is the ultimate target of machine learning.





Generative model and Discriminative model

- Machine learning = make predictions = $P(y|x)$
- Discriminative model:
 - model $P(y|x)$ directly, reflects the diversities between different classes
- Generative model:
 - model $P(x, y)$, reflects the similarity of samples within classes.
 - $P(x, y) = P(x|y)P(y) \rightarrow$ model $P(x|y)$
 - $P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)P(y)}{P(x)}$
 - Our model
 - Prior
 - Not important

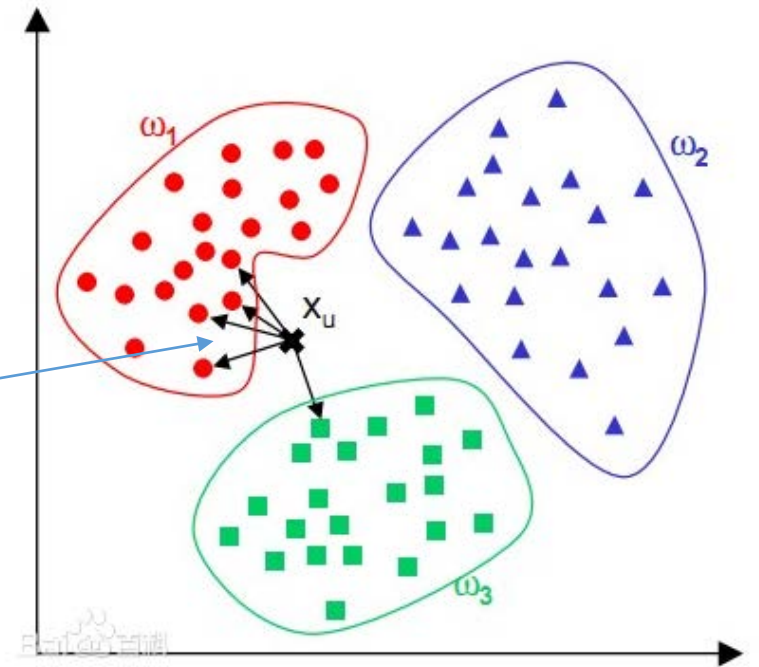
Machine Learning and Deep Learning

- Deep Learning is a part of machine learning
- But it is very different to traditional machine learning methods:
 - Big data support
 - Huge amount of parameters
 - End to End
 - Black box
 - Strong Practicality
 - Personal experience
 - Special-designed tools

kNN: easy and lazy

- Model:
 - Given a dataset D , for sample s , the prediction of s is determined by its k neighbors.
- Objective: None (No trainable parameters)
- Optimizer: None (No trainable parameters)

How to define “distance”



Logistic Regression: discriminative model

- Model:

For linear model $y = w^T x + b$. If the relationship between x and y is not linear:

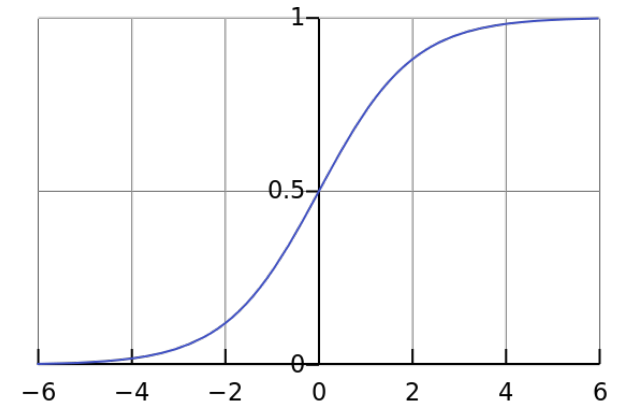
$$g(y) = w^T x + b \rightarrow y = g^{-1}(w^T x + b)$$

In binary classification, $y \in (0,1)$, we need g^{-1} that: Generalized Linear Model

- Mapping $w^T x + b$ to $(0,1)$ for all x
- (at least) 1-order differentiable

Thus our model is:

$$y = \frac{1}{1 + e^{-(w^T x + b)}}$$




What is the objective of LR

Objective:

- Misclassification rate / Accuracy ?

$$y = \frac{1}{1 + e^{-(w^T x + b)}} \in (0,1)$$

 Metrics (ROC, AUC...)

Since $y \in (0,1)$, If:

$$y \Rightarrow P(y = 1|x)$$

Then:

$$1 - y \Rightarrow P(y = 0|x)$$

Let

$$\theta^T = (w^T, b)$$

We have:

What is the objective of LR

$$P(y = 1|x) = \frac{1}{1 + e^{-\theta^T x}} = \frac{e^{\theta^T x}}{1 + e^{\theta^T x}}$$

And:

$$P(y = 0|x) = \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}} = \frac{1}{1 + e^{\theta^T x}}$$

Thus we could optimize the model through maximize the likelihood function, which is also our objective function

$$\begin{aligned} l(\beta) &= \sum_{i=1}^m \ln P(y_i|x_i; \theta) \\ &= \sum_{i=1}^m \ln [y_i p_1(x_i; \theta) + (1 - y_i) p_0(x_i; \theta)] \\ &= \sum_{i=1}^m \ln \left(\frac{y_i e^{\theta^T x} + 1 - y_i}{1 + e^{\theta^T x}} \right) \end{aligned}$$

What is the objective of LR

Maximize the likelihood = minimize the negative likelihood:

$$\begin{aligned} l(\theta) &= - \sum_{i=1}^m \ln \left(\frac{y_i e^{\theta^T x} + 1 - y_i}{1 + e^{\theta^T x}} \right) \\ &= \sum_{i=1}^m [\ln(1 + e^{\theta^T x}) - \ln(y_i e^{\theta^T x} + 1 - y_i)] \end{aligned}$$

Note that $y_i \in \{0,1\}$:

- For $y_i = 1$: $\ln(y_i e^{\theta^T x} + 1 - y_i) = \ln(e^{\theta^T x}) = \theta^T x = y_i \theta^T x$
- For $y_i = 0$: $\ln(y_i e^{\theta^T x} + 1 - y_i) = \ln(1) = 0 = y_i \theta^T x$

Thus:

$$l(\theta) = \sum_{i=1}^m [\ln(1 + e^{\theta^T x}) - y_i \theta^T x]$$

2-order
differentiable



How to optimization

Target:

$$\theta = \underset{\theta}{\operatorname{argmin}} l(\theta)$$

Newton methods (2-order differentiable)

$$\theta^{t+1} = \theta^t - \left(\frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^T} \right)^{-1} \frac{\partial l(\theta)}{\partial \theta}$$

Gradient Descent (1-order differentiable)

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial l(\theta)}{\partial \theta}$$

Homework

My friend Hellen has been using some online dating sites to find different people to go out with. She realized that despite the site's recommendations, she didn't like everyone she was matched with. After some introspection, she realized there were three types of people she went out with:

- People she didn't like
- People she liked in small doses
- People she liked in large doses

Hellen has been collecting data for a while and has 1,000 entries. A new sample is on each line, and

Hellen has recorded the following features:

- Number of frequent flyer miles earned per year
- Percentage of time spent playing video games
- Liters of ice cream consumed per week

Can you help to classify these people?

Homework

- Implement kNN with pure python code
- You can import only standard libraries and numpy
- The result must be the average of 5-fold cross-validation
- You must submit your code and results to github: MoyanZitto/2017_ml_avc2. You can find the dataset there.
- Learn to use github through “廖雪峰的git教程” if you are not familiar with git.