MATH466/MATH766
Math of machine learning
01/15 Lecture 3 classification models

References:

- Ch2, 4, 12 of The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman

Todays contents:

- logistic regression
- SVM (support vector machine)
- kNN (k nearest neighborhood)

Important concepts:

- MLE (maximum likelihood estimation)
- support vector
- relaxation
- parametric and non-parametric models

Recommend reading:

-

warm up:

1. Assume (x_i, y_i) are i.i.d samples from $p(x,y)$

   $P(\{(x_i, y_i)\}_{i=1}^n) =$ _____

2. The normal direction to the hyperplane

   $w^T x + b = 0$ is _____

3. Consider two vectors $a, b \in \mathbb{R}^d$.

   the length of projection of a on b is _____

0. Regression and Classification :

data $\{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

$$\min_{f \in \mathcal{F}} R(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$$

	Regression	Classification
range of f	usually an interval in $\mathbb{R}$ e.g. $[0,1]$, $\mathbb{R}$	usually a discrete set in $\mathbb{R}$ e.g. $\{0,1\}$, $\{-1,1\}$, $\{0,\cdots,9\}$
choice of ℓ	usually $\ell(\hat{y}, y) = (\hat{y}-y)^2$ ↓ mean square error (MSE)	usually $\ell(\hat{y}, y) = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{if } \hat{y} = y \end{cases}$ ↓ accuracy

1. Logistic Regression

We consider binary classification as an example. $y \in \{-1, 1\}$

See HW1 for multiclass case

1.1 Training

Q: As a classification model, why is it called "regression"?

Let $p(x) := \mathbb{P}(y=1 \mid x)$, range of $p(x)$ is $[0,1]$

learn $p(x)$ from regression

→ do classification based on $p(x)$

Difficulty: $p(x) \in [0,1]$, not easy to parameterize directly

Solution: WLOG, assume $p(x) \in [0, 1)$

then $\frac{p(x)}{1-p(x)} \in (0, +\infty)$

$\log \frac{p(x)}{1-p(x)} \in \mathbb{R}$ ($\log \frac{p}{1-p}$ is called the logit of p)

parametrize this instead

For example, if we parameterize $\log \frac{p(x)}{1-p(x)}$ by $w^T x + b$, $w \in \mathbb{R}^d$, $b \in \mathbb{R}$

then $\frac{p(x)}{1-p(x)} = \exp(w^T x + b)$

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

$p(x; w, b) = \frac{1}{1 + \exp(-(w^T x + b))}$

Under this parameterization, the probability of observing (x_i, y_i) is

$$\begin{cases} \text{if } y_i = 1, \; p(x_i; w, b) = \dfrac{1}{1+\exp(-(w^T x_i + b))} \\ \text{if } y_i = -1, \; 1 - p(x_i; w, b) = \dfrac{\exp(-(w^T x_i + b))}{1+\exp(-(w^T x_i + b))} \end{cases}$$

$$= \frac{1}{1+\exp(w^T x_i + b)}$$

$$= \frac{1}{1+\exp(-y_i(w^T x_i + b))}$$

Following the principle of maximum likelihood estimation (MLE)

choose (w, b) to maximize the probability of observed data

$$\max_{(w,b)\in\mathbb{R}^{d+1}} \prod_{i=1}^{n} \frac{1}{1+\exp(-y_i(w^T x_i + b))}$$

↑ not easy to deal with

Notice that (w, b) maximize $R(w,b)$ $(R(w,b) > 0)$

iff it minimize $-\log R(w,b)$

$$\min_{(w,b)\in\mathbb{R}^{d+1}} \sum_{i=1}^{n} \log\left(1+\exp(-y_i(w^T x_i + b))\right) \qquad (*)$$

We will discuss how to solve $(*)$ in optimization module

1.2. Prediction

Assume that we have solve (*) for (w, b)

Q: new input $x \xrightarrow{?}$ label y

compute $p(x; w, b)$. $\begin{cases} y = 1 \text{ if } p(x) > \frac{1}{2}, \\ y = -1 \text{ if } p(x) < \frac{1}{2}, \end{cases}$

notice that $p(x) > \frac{1}{2} \iff w^T x + b > 0$

$p(x) < \frac{1}{2} \iff w^T x + b < 0$

therefore $\begin{cases} y = 1 \text{ if } w^T x + b > 0 \\ y = -1 \text{ if } w^T x + b < 0 \end{cases}$

The hyperplane $\{x : w^T x + b = 0\}$ is called decision boundary.

Parameterize $\log \frac{p(x)}{1-p(x)}$ with linear model

leads to linear decision boundary

1.3. Generalization

Q: What if the data cannot be separated by linear decision boundary?

① Parameterize $\log \frac{p(x)}{1-p(x)}$ by $f(x;\theta)$, θ are parameters

$$p(x;\theta) = \frac{1}{1+\exp(-f(x;\theta))}$$

$$\min_{\theta \in \mathcal{H}} \sum_{i=1}^{n} \log\left(1+\exp(-y_i f(x_i;\theta))\right)$$

decision boundary $f(x;\theta) = 0$

(derivation left as exercise)

② Map x to $\phi(x)$

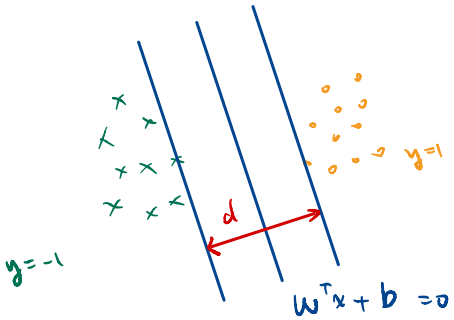
such that $\{(\phi(x_i), y_i)\}_{i=1}^{n}$ are linearly separable

This leads to "kernel method".

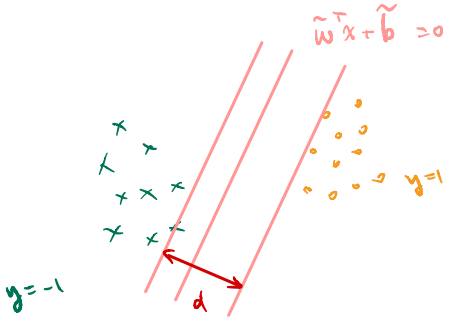
We will see an example in HW1 and talk about more details in later lectures.

2. Support Vector Machine (SVM)

2.1 Hard-margin



When the two classes of data are linearly separable and there are more than one plane perfectly separating them, which will you choose?



In general,

larger margin dist. d → more robust prediction

we want to find the hyperplane that creates the largest margin.

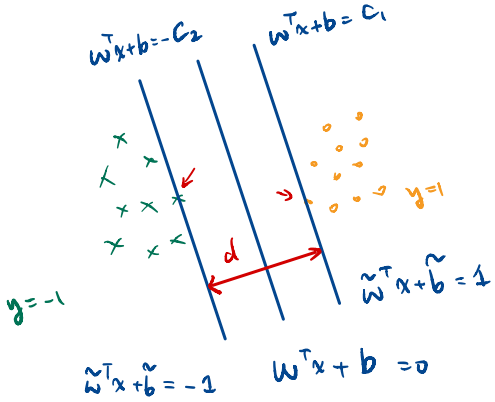
Assume that $\exists (w,b)$, s.t. $w^Tx+b=0$ perfectly separates the two classes,

then WLOG, we can assume for any

x_i with $y_i=1$, $f(x_i)=w^Tx_i+b>0$

x_i with $y_i=-1$, $f(x_i)=w^Tx_i+b<0$

$c_1, c_2 > 0$



By rescaling and shifting (w,b),

we can assume WLOG,

for any

x_i with $y_i=1$, $f(x_i)=w^Tx_i+b \geq 1$

x_i with $y_i=-1$, $f(x_i)=w^Tx_i+b \leq -1$

x_i such that $f(x_i)=1$ or -1

i.e. samples on the margin are called support vectors

$$\tilde{w}=\frac{2}{c_1+c_2}w$$

$$\tilde{b}=\frac{2b-c_1+c_2}{c_1+c_2}$$

The distance we want to maximize:

the distance between $w^T x + b = 1$ and $w^T x + b = -1$ $= \frac{2}{\|w\|_2}$

Problem formulation:

$$\min_{(w,b) \in \mathbb{R}^{d+1}} \quad \frac{1}{2} \|w\|_2^2$$

$$\text{s.t.} \quad y_i (w^T x_i + b) \geq 1, \quad i = 1, \cdots, n$$

Q: Why not work with $\max_{(w,b)} \frac{2}{\|w\|_2}$?

Q: How to deal with the constraint ?

Answer in Optimization module.

2.2 Soft-margin (relaxation)

If the data cannot be perfectly separated by a hyperplane

then the hard-margin problem does not admit a feasible solution.

We can relax the constraint and penalize the violation

$$\min_{w,b,\xi} \quad \frac{1}{2} \|w\|_2^2 + \lambda \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \cdots, n$$

Equivalently, you will see the following formulation in literature

$$\min_{w,b} \quad \frac{1}{2} \|w\|_2^2 + \lambda \sum_{i=1}^{n} \underbrace{\max(0, 1 - y_i (w^T x_i + b))}_{\text{hinge loss}}$$

unconstrained problem, but non-smooth objective.

3. K-Nearest Neighbourhood

Different from previous methods, kNN is **non parametric**.

The technique do not rely on assumptions about
the underlying distribution of the data and the form of prediction functions

Because they are unstructured, they may not be useful for understanding
the relation between the features and class outcome.
But they can be very effective.

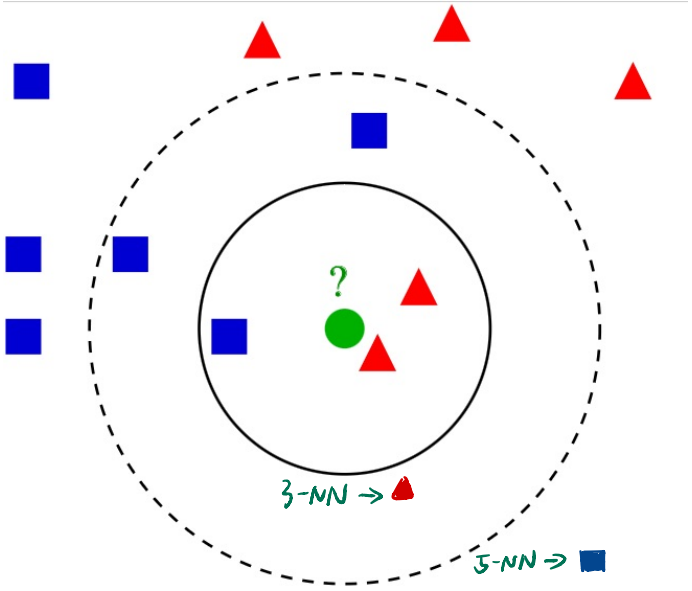
The idea of k-NN is

Majority Vote

Given a query point x_0,

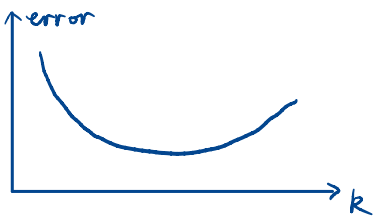
we find the k training points $x_{(r)}$
$r = 1, 2, \cdots, k$ that is closest in distance
to x_0.

if the features are quantitative and
real-valued, we can use Euclidean
distance $d_{(i)} = \| x_{(i)} - x_0 \|$



then the class of x_0 is determined by majority vote.

- pre-process : deal with qualitative and ordinal features (ref [HTF09] ch14)
  standardize each of the features to have
  mean 0 and variance 1.
- choice of distance : depends on data features and prior knowledge.
  e.g. to compute the distance between images,
  one can use sum of pointwise distance square
  or "optimal transport distance".
- choice of k : in general, u-shape curve



for large k, one way to improve is

changing uniform vote to weighted vote.

e.g. weighted by $\frac{1}{d_{(i)}}$

- how to conduct k-NN efficiently for large dataset and high-dimensional data

  (can be a topic for course project)

K-NN classifier is especially useful when the decision boundary is very irregular. It is proved that asymptotically a k-NN classifier won't perform "too bad".

This k-nearest neighbourhood idea can also be used for regression and clustering (later in this course)