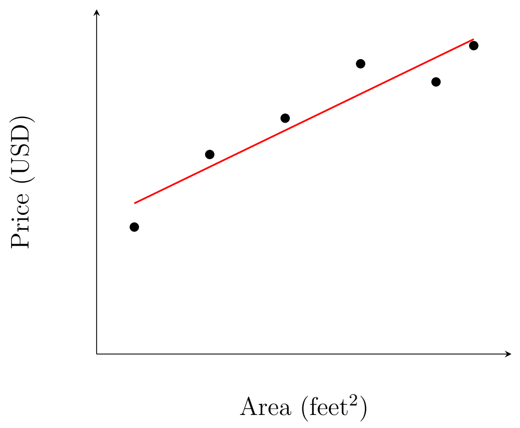
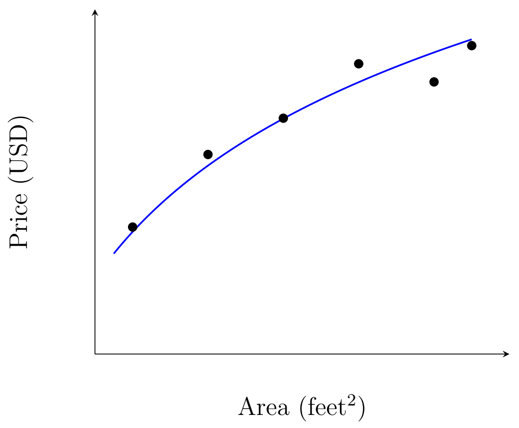
Lecture 2

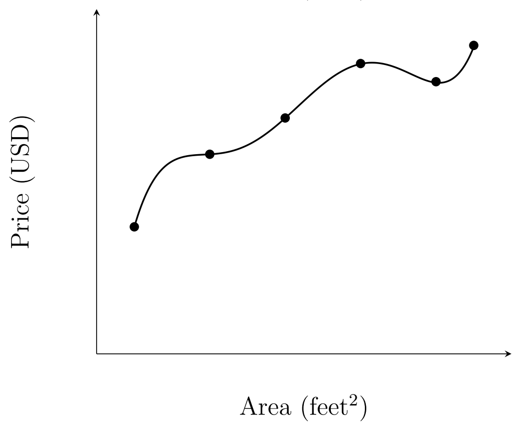
1. Locally weighted regression
2. Linear regression: probabilistic interpretation
3. Logistic regression
4. Perceptron

1 Locally weighted regression

Consider three different models for our previous example.







- underfitting
- good fit
- overfitting

In the first case we fit using hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$ and obviously that our model is not very accurate (in machine learning we call it **underfitting**).

For the second case we fit the model with hypothesis $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ and it seems that this model will be accurate enough. For the last example we fit the polynomial of the fifth order and it will fit data exactly, but our intuition says us that this model is not good. In machine learning we call it **overfitting**.

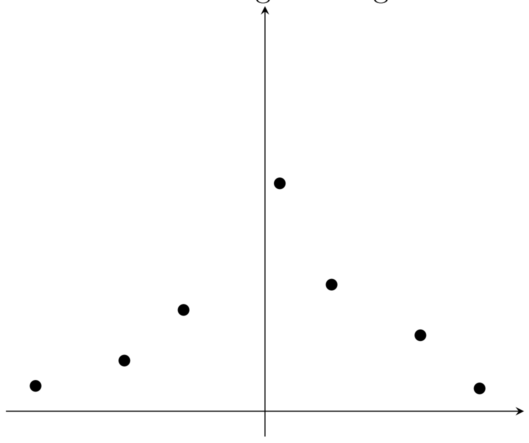
As we can see we can increase number of parameters if we increase the number of samples.

Definition. Parametric learning algorithm is the algorithm with fixed number of parameters (for example, linear regression).

Definition. Nonparametric learning algorithm is the algorithm where number of parameters grows with m.

In this section we consider the example of nonparametric algorithm: locally weighted regression (loess or lowess).

Given the following training set



In linear regression to evaluate h at certain x we fit θ to minimize $\sum\limits_i \left(y^{(i)} - \theta^T x^{(i)}\right)^2$ and return $\theta^T x$.

In loess we look at the points from dataset that are closest to x and fit linear regression for these points only. In mathematical language we fit θ to minimize

$$\sum_i w^{(i)} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

where $w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$ and τ is bandwidth parameter.

If $|x^{(i)} - x|$ is small, then $w^{(i)} \approx 1$.

If $|x^{(i)} - x|$ is large, then $w^{(i)} \approx 0$

The weight is proportional to the height of the bell-shaped curve. If τ is large the bell shape is wider.

The disadvantage of loess is that we need to fit the linear regression each time we want to predict.

2 Linear regression: probabilistic interpretation

In the previous lecture we agreed that the cost function $J(\theta)$ is defined as a sum of squared differences. The reasonable question is why we choose the function $J(\theta)$ in such form? In this section we look at the same problem from different point of view.

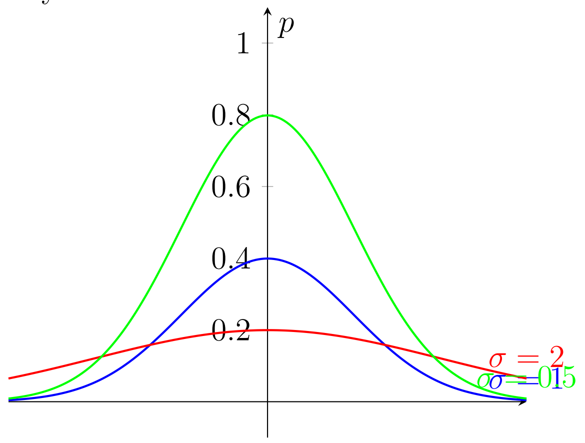
We assume that $y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$, where $\varepsilon^{(i)}$ is an error term distributed normally:

$$\varepsilon^{(i)} \sim N(0, \sigma^2) \text{ (gaussian distribution).}$$

The density for this error is defined as

$$p(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\varepsilon^{(i)})^2}{2\sigma^2}\right),$$

where σ is a standard deviation and shows the width of the bell-shaped density function.



Then

$$p(y^{(i)}|x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

In other words,

$$y^{(i)}|x^{(i)}, \theta \sim N(\theta^T x^{(i)}, \sigma^2)$$

If we assume that θ is not a random variable, but some value, we say that we parametrize distribution by θ. Notice that $\varepsilon^{(i)}$'s are independently identically distributed. We define the likelihood as

$$L(\theta) = p(y|x;\theta) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right).$$

The idea of likelihood is that we take $p(y|x;\theta)$ and consider it as a function of θ. How to choose parameters θ?

Maximum likelihood estimation: choose θ to maximize $L(\theta) = p(y|x;\theta)$. We define

$$\begin{aligned} l(\theta) = \ln L(\theta) = \ln \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) = \\ = \sum_{i=1}^{m} \ln\left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)\right] = \\ = m \ln \frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} -\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}. \end{aligned}$$

So maximizing $l(\theta)$ is the same as minimizing

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2$$

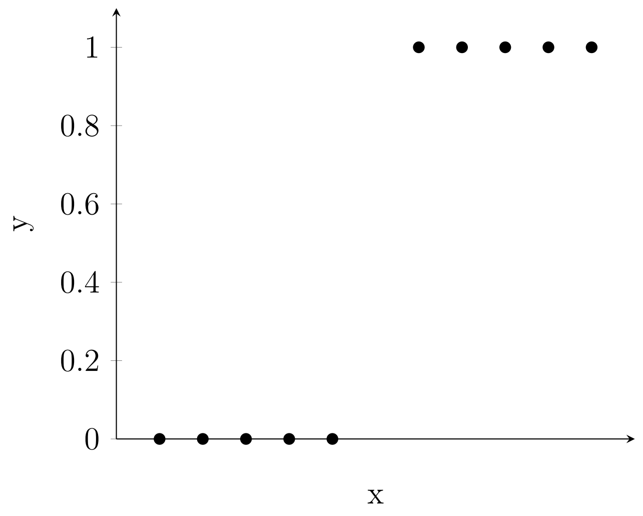
Notice that the standard deviation σ is not important for the optimization.

Exercise. Assuming that $y \sim N(\mu, \sigma^2)$ (that means that we predict constant value μ for all test examples), what is the best value of μ?

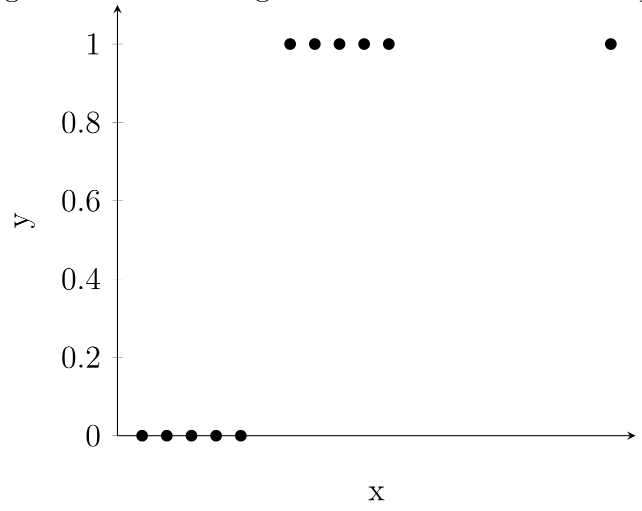
Hint: $p(y|x;\mu) = \dfrac{1}{\sqrt{2\pi}\sigma} \exp\left(-\dfrac{(y-\mu)^2}{2\sigma^2}\right)$

3 Logistic regression

Let us build our first classification algorithm. The simplest classification problem claims that the output $y \in \{0, 1\}$.

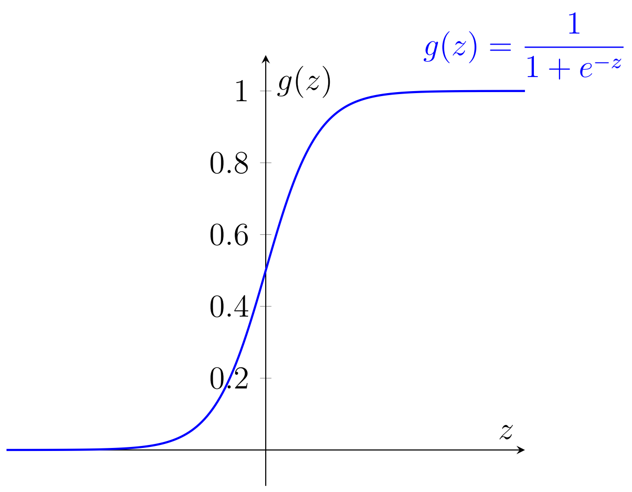


One way to solve this problem is to train linear regression, and after we take some threshold for the straight line. Sometimes it works, but in general it is not a good idea. Here is the example, when this does not work.



For the second case linear regression gives very bad result. The good idea is to change our hypothesis such that $h_\theta(x) \in [0, 1]$. But with this assumption linear function is not the best choice for $h_\theta(x)$. We choose $h_\theta(x)$ as follows:

$$\begin{aligned} h_\theta(x) &= g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}, \\ g(z) &= \frac{1}{1 + e^{-z}} - \text{ sigmoid function or logistic function.} \end{aligned}$$



To understand the logistic regression from the probabilistic point of view we define the probability density function as

$$\begin{aligned} p(y=1|x;\theta) &= h_\theta(x) \\ p(y=0|x;\theta) &= 1-h_\theta(x), \end{aligned}$$

then

$$p(y|x;\theta) = h_\theta(x)^y \left(1-h_\theta(x)\right)^{1-y}.$$

As before, the likelihood

$$L(\theta) = p(y|X;\theta) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) = \prod_{i=1}^{m} h_\theta(x^{(i)})^{y^{(i)}} \left(1-h_\theta(x^{(i)})\right)^{1-y^{(i)}}$$

and we maximize the log-likelihood

$$l(\theta) = \ln L(\theta) = \sum_{i=1}^{m} y^{(i)} \ln h_\theta(x^{(i)}) + (1-y^{(i)}) \ln(1-h_\theta(x^{(i)})).$$

To do find the value for θ that maximizes the log-likelihood we apply gradient ascent:

$$\theta := \theta + \alpha \nabla_\theta l(\theta),$$

where

$$\frac{\partial}{\partial\theta_j} l(\theta) = \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}.$$

Then gradient ascent can be written as

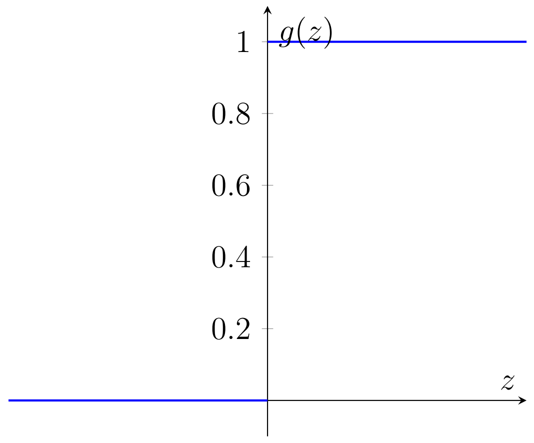
$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)},$$

which is exactly the same rule as for least square regression, except that $h_\theta(x)$ is different.

4 Perceptron

Define the function

$$g(z) = \begin{cases} 1 & \text{if } z \geqslant 0, \\ 0 & \text{otherwise.} \end{cases}$$



If the hypothesis is expressed as $h_\theta(x) = g(\theta^T x)$, then learning rule becomes:

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

It turns out that it is very difficult to use probabilistic semantics in this learning rule. But in the future we are going to use this rule as a building block of some algorithms.