50.021 -AI

Alex

Week 01: Ordinary Least Squares (Linear regression), Basis Functions

Recap

We assume we have an input space $\mathcal{X}$, and output space $\mathcal{Y}$. We want to predict for $x \in \mathcal{X}$ a value $y \in \mathcal{Y}$ by a mapping $f : \mathcal{X} \to \mathcal{Y}$.

Want to find a function $f : \mathcal{X} \to \mathcal{Y}$ that minimizes some notion of loss $L(f(x), y)$ between prediction $f(x)$ on data sample x and the ground truth y for the sample x.

$$E_{(x,y)\sim P}[L(f(x), y)]$$

Ordinary Least Squares (Linear regression), Basis Functions

Input space $\mathcal{X}$ is a vector space $\mathcal{X} \ni x = (x^{(1)}, \dots, x^{(D)}) \in \mathbb{R}^{1\times D}$. Desired output are continuous real values (speed, blood pressure, beer flow rate).

Choose mapping

$$f_w(x) = x \cdot w = \sum_{d=1}^{D} x_d w_d,\ w \in \mathbb{R}^{D\times 1}$$

What does a linear mapping represent? important for vector → real number mapping

Lets visualize the mapping $g_{w,b}(x) = w \cdot x + b$ in three ways:

1. For what set of points x_1, x_2 we have $g_{w,b}(x_1) = g_{w,b}(x_2)$?

2. What is the set of points x where the prediction is zero (the zero set), that is $g_{w,b}(x) = 0$?

3. What is the set of points x where the prediction is a constant, that is $g_{w,b}(x) = c$?

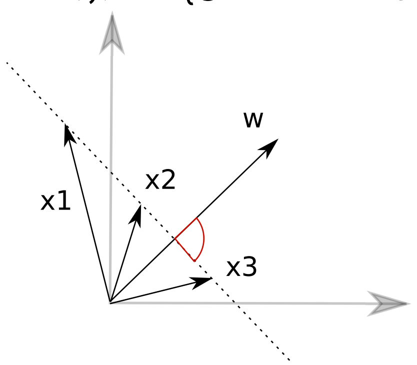
Consider a fixed (w, b), and see what points x gets mapped onto the same values for the term $g_{w,b}(x) = w \cdot x + b$ inside the sign.

A. $g(x_1) = w \cdot x_1 + b = w \cdot x_2 + b = g(x_2)$ if $w \cdot x_1 = w \cdot x_2$

B. $w \cdot x_1 = w \cdot x_2$ if $w \cdot (x_1 - x_2) = 0$ – The difference vector between x_1 and x_2 is orthogonal to the vector w.

point 1

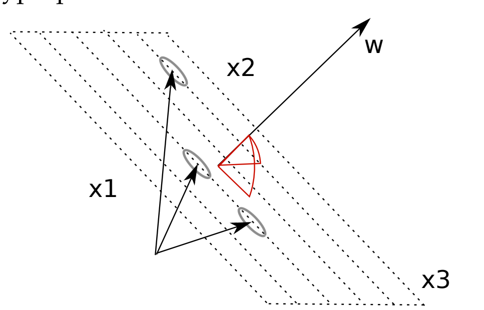
What is the space of all vectors orthogonal to a vector w? It is a plane. (1 dimension less than 3D)



x_1,x_2,x_3 all have the same $f(x_i)$ – their vector difference lies in a plane orthogonal to vector w. Also their length **as projected into the direction of** w is the same. Note: x_1 and x_2 can have different euclidean lengths - depending on the shift along the plane of vectors orthogonal to w.

$w \cdot (x_1 - x_2) = 0$ has the meaning: you can shift x_1 by adding any vector in the plane of vectors orthogonal to w, and you get another vector x_2 with same inner product

The same also holds for 3 or more dimensions. For n dimensions the plane of orthogonal vectors has $n-1$ dimensions. So for 3 dims it is a two dim hyperplane.



Take away: $g_{w,b}(x_1) = g_{w,b}(x_2)$ if the difference vector $x_1 - x_2$ is orthogonal to w, that is it lies in the orthogonal hyperplane of w.

Choose $x \perp u$

Point 2: What is the set of points x where the sign of the prediction switches, that is $g_{w,b}(x) = 0$?

e.g. $w = (w_1, w_2)$

Firstly, we can apply what we have found out about $g_{w,b}(x_1) = g_{w,b}(x_2)$. $g_{w,b}(x_1) = g_{w,b}(x_2) = 0$ if the difference vector $x_1 - x_2$ is

then $x = (-w_2, w_1) \cdot \lambda$

$x = (-\lambda w_2, \lambda w_1)$

$\therefore f_w(x) = w \cdot x = 0$

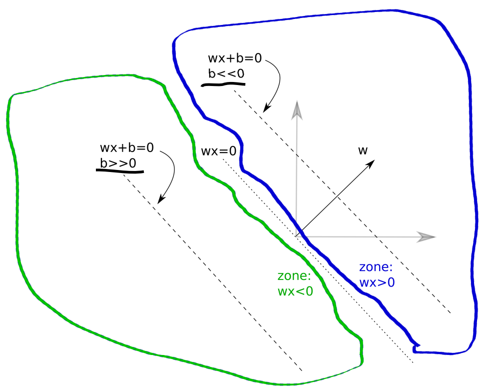
orthogonal to w, but now we also require: $w \cdot x_1 = -b$ – that depends on the bias.

Secondly, we know: $wx > 0$ for all points that are on that side of the hyperplane, in which w points to.

In order to understand how the bias b influences the zero set, lets consider three cases: $b = 0$, $b >> 0$ and $b << 0$. We know that for $b = 0$: $f(x) = w \cdot x = 0$ holds for the zero vector $x = 0$. That means: if $b = 0$, then the zero set contains the zero vector $x = 0$. Also: the zero set are all those vectors x which are orthogonal to vector w. See the graphic below.

The bias b shifts the zero set corresponding to $w \cdot x + b = 0$ parallel to the direction of w. Positive $b > 0$ shift the hyperplane to the side of the hyperplane orthogonal to w where $wx < 0$ and vice versa (if b is negative, then the shift will be to the side of the hyperplane orthogonal to w where $wx > 0$). See the graphic. Large values of $|b|$ shift it away quite far.

negative



positive

As a summary: the linear mapping $g(x) = w \cdot x + b$ has a zero set which is the plane of points

$$\boxed{x = \{u + -b\frac{w}{\|w\|^2}, u \text{ such that } w \cdot u = 0\}}$$

In this representation:

- u such that $w \cdot u = 0$ is the hyperplane of vectors u orthogonal to w.
- the vector $-b\frac{w}{\|w\|^2}$ shifts the hyperplane parallel to direction of w.

That holds because

$$\begin{aligned} w \cdot x + b &= w \cdot (u + -b\frac{w}{\|w\|^2}) + b \\ &= w \cdot u + w \cdot (-b\frac{w}{\|w\|^2}) + b \\ &= 0 + -b\frac{w \cdot w}{\|w\|^2} + b = 0 + -b\frac{\|w\|^2}{\|w\|^2} + b = 0 \end{aligned}$$

Point 3: What is the set of points x where the prediction is a constant, that is $g_{w,b}(x) = c$?

Choose $x = c \cdot \frac{w}{\|w^2\|}$

$\therefore f_w(x) = w \cdot c \cdot \frac{w}{\|w^2\|}$

$= c \cdot \frac{\|w^2\|}{\|w^2\|} = c$ #

This can be answered by reducing it to a zero set:

$$g_{w,b}(x) = wx + b = c$$
$$wx + (b - c) = 0$$

So the set of points x such that $g_{w,b}(x) = c$ is just the zero-set of $g_{w,b-c}(x)$. (modified bias parameter b)

Loss function for regression

Loss function for a pair (x, y):

$$L(f(x), y) = (f(x) - y)^2$$

OLS objective:

$$w^* = \text{argmin}_w \sum_{i=1}^{n} (x_i \cdot w - y_i)^2$$

for a given dataset $D_n = \{(x_i, y_i)\}$. Then $f_{w^*}(x) = x \cdot w^*$ is the selected mapping.

Rare case: Can be solved explicitly for w. Write in matrix form:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1^{(1)}, \ldots, x_1^{(D)} \\ x_2^{(1)}, \ldots, x_2^{(D)} \\ \vdots \\ x_n^{(1)}, \ldots, x_n^{(D)} \end{pmatrix} \in \mathbb{R}^{n \times D}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

then:

$$\sum_{i=1}^{n} (x_i \cdot w - y_i)^2 = (X \cdot w - Y)^T \cdot (X \cdot w - Y)$$

Solve the minimization problem by computing the gradient for w and setting it to zero.

$$D_w((X \cdot w - Y)^T \cdot (X \cdot w - Y)) = 2X^T \cdot (X \cdot w - Y) = 0$$
$$(X^T \cdot X) \cdot w = X^T \cdot Y$$
$$w = (X^T \cdot X)^{-1} X^T \cdot Y$$

condition: the inverse exist

if the inverse exists.

Demo: `t1()` shows a solution.

e.g. $\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}$ is not invertable

Basis functions

Linear functions try to fit a line - can be too restricted for fitting many datasets. Alternative: Map data $x_1, \dots, x_n$ into some feature space by some mapping ϕ, then do a linear regression on $\phi(x_1), \dots, \phi(x_n)$

more dimensions, better predictions (but not necessarily due to risk of overfitting)

Example: polynomial basis functions

$$\phi(x) = \begin{pmatrix} 1 \\ x \\ x^2 \\ x_3 \\ \dots \\ x^F \end{pmatrix}$$

(for a vector do this on each dimension)

What is the relevance of basis functions beyond direct usage in linear regression?

- Radial basis functions are able to represent very general functions (see in class coding)
- A layer in a neural network can be interpreted as: being a set of (learned, not fixed) basis functions that get used in the next layer.

Demo: `t2()` shows a solution for varying values of F
Demo: `t3()` shows train and test error for varying values of F
Problem: for n data points a polynomial with $F = n - 1$ achieves zero training error, but test error is too high – overfitting

Regularization

Overfitting: low training error, high test error. Reason: during learning one picks up too much of the noise in the data, and learns weights that listen to noise signals.

One way to deal with it: avoiding weights w getting too large. How to do that ? Add a penalty on the euclidean length of w

$$\text{argmin}_w \sum_{i=1}^{n} (x_i \cdot w - y_i)^2 + \underbrace{\lambda \|w\|^2}_{\text{penalty (to be minimized too)}}$$

Overfitting: looking at the data too closely. Regularization – avoid to do that too fine-grained look. Formally: regularization is a penalty on large parameter weights.

in matrix form

$$\sum_{i=1}^{n} (x_i \cdot w - y_i)^2 + \lambda \|w\|^2 = (X \cdot w - Y)^T \cdot (X \cdot w - Y) + \lambda w \cdot w$$

$$
\begin{aligned}
D_w((X \cdot w - Y)^T \cdot (X \cdot w - Y) + \lambda w \cdot w) = 2X^T \cdot (X \cdot w - Y) + 2\lambda w = 0 \\
(X^T \cdot X + \lambda I) \cdot w = X^T \cdot Y \\
w = (X^T \cdot X + \lambda I)^{-1} X^T \cdot Y
\end{aligned}
$$

One can see: for any $\lambda > 0$ the matrix $A = (X^T \cdot X + \lambda I)$ has only positive eigenvalues, so it is always invertible. a matrix A has only positive eigenvalues if for all $v \neq 0$ we have $v^T A v > 0$. This can be shown. Assume $v \neq 0, \lambda > 0$, then

Solution always exists

$$
\begin{aligned}
& v^T(X^T \cdot X + \lambda I)v = v^T X^T \cdot Xv + v^T \lambda I v \\
& = (Xv)^T Xv + \lambda v^T v = \underbrace{\|Xv\|^2}_{\geq 0} + \lambda \|v\|^2 \\
& \geq \lambda \|v\|^2 > 0
\end{aligned}
$$

proof

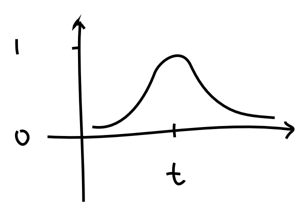
Moreover: adding λI to $X^T \cdot X$ puts a lower bound on eigenvalues of $(X^T \cdot X + \lambda I)$, so for the inverse the eigenvalues become upper-bounded. "better-conditioned inverse".

Demo: `t5()` shows train and test error for varying values of $\lambda = \exp(\gamma), \gamma \in \{-50, -40, -30, -20, -15, -10, -1, 0, 1, 10\}$ and $F = 9$ – this value of F overfitted in the last example notable (because $n = 10 = F + 1$), but with regularization it becomes manageable. Regularization allows to use a high dimensional feature space, and still find a reasonable solution

Example: radial basis functions

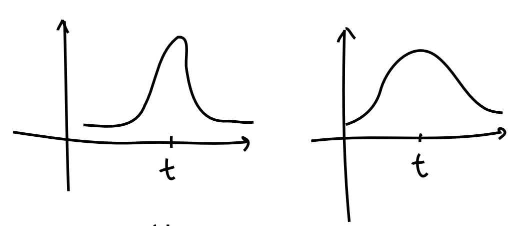
Given a dataset $(x_1, \ldots, x_n)$, the gaussian radial basis function mapping is defined for any sample x as:

$$
\begin{aligned}
\phi(x) = (rbf(x, x_1), rbf(x, x_2)), \ldots, rbf(x, x_n)) \in \mathbb{R}^n \\
rbf(u, v) = \exp(-\frac{\|u - v\|^2}{\lambda^2})
\end{aligned}
$$



Implement it, and see its impact for different choices of kernel width λ in `learn.py`

Call `tRBF([val1,val2,val3])` with different values of the kernel width. For a too small kernel width one will have train error zero.



small λ large λ