SPR Day 07

Probabilistic Linear Regression

Reading: Bishop Sec 3.1

we'll cover
- Linear Basis Function models
- Maximum likelihood
- "Least Squares" closed form solution
- Penalized maximum likelihood

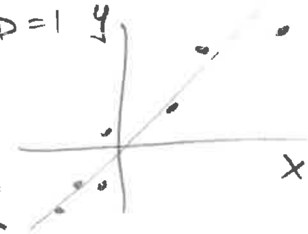
Reading: Bishop Sec 3.3

we'll cover
- Gaussian-Gaussian model
- Posterior for weights
- MAP estimator

next time → predictive posterior

Linear Regression — Standard Features

If $D=1$



Goal: Given dataset $\{x_n, t_n\}_{n=1}^{N}$

with $x_n \in \mathbb{R}^D$ and $t_n \in \mathbb{R}$

want to <u>predict</u> t_* given x_*

Assume: "Linear" model, which means prediction function is linear function of input x_n

$$y(x_n, w) = w_0 + w_1 x_{n1} + w_2 x_{n2} + \cdots w_D x_{nD}$$

$$= \sum_{d=0}^{D} w_d x_{nd} \qquad \left(\text{defining } x_{n0} = 1 \ \forall n\right)$$

$$= w^T x_n \qquad \text{inner product of two } D+1 \text{ vectors}$$

Often can define "smarter" features by transforming input x_n into another feature space via $\phi(x_n)$

Define $\phi(x_n) = [\,1 \quad \phi_1(x_n) \quad \phi_2(x_2) \cdots \phi_{M-1}(x_n)\,]$

sometimes called "basis" function

M total entries, includ[ing] "always 1" feature

$\phi_M(x_n)$ can be non-linear!

$$x_{n1}^2 \quad \text{or} \quad x_{n1} x_{n3} \quad \text{or} \quad \cos(x_{n4}) \quad \text{or} \ldots$$

Key idea is that we define a feature transform function $\phi(x_n)$ in advance, with known size M.

"Featurized" model for prediction:

$$(x_n, w) = \sum_{m=1}^{M} w_m \phi_m(x_n) = w^T \phi(x_n)$$

Note: our predictions will <u>not</u> be perfect.
Need to tolerate some noise.
Let's define a probabilistic approach.

Likelihood of observing output t_n given input x_n

$$p(t_n | x_n, w, \beta) = \mathcal{N}(t_n | \overbrace{w^T \phi(x_n)}^{\text{mean}}, \overbrace{\beta^{-1}}^{\text{variance}})$$

notation for Normal PDF

1D variable in $\mathbb{R}$

β^{-1} is $\mathrm{Var}[t_n]$

β is <u>precision</u> of t_n

If we assume <u>all</u> N observations are i.i.d. from this distribution

$$p(t | x, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n | w^T \phi(x_n), \beta^{-1})$$

Taking log of both sides and simplifying

$$\log p(t | x, w, \beta) = \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi) - \beta\frac{1}{2}\overbrace{\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2}^{\text{sum of squared errors}}$$

Key idea: Can treat this as a log likelihood,
apply ML ideas to estimate w_{ML}, β_{ML}

step 1 — step 2

View log likelihood as function of w, β, then try to maximize by taking gradients & setting equal to ϕ & solving

Step 3

Step 1

$$\mathcal{L}(w,\beta) = \frac{N}{2}\log\beta - \beta\frac{1}{2}\sum_n (t_n - w^T\phi(x_n))^2 + \text{const wrt } w, \beta$$

Step 2 Gradient wrt w_{ML}

$$\nabla_w \mathcal{L}(w,\beta) = \text{zero} + -\frac{1}{2}\beta \sum_n \nabla_w \left(t_n^2 - 2t_n w^T\phi(x_n) + w^T\phi(x_n)^2\right)$$

$$= \text{zero} + + \beta\sum_n \left[t_n\phi(x_n) - \beta\frac{1}{2}\nabla_w\left[w^T\phi(x_n)\right]^2\right]$$

$$= + \beta\sum_n \left[t_n\phi(x_n) - \beta\frac{1}{2}2\left[w^T\phi(x_n)\right]\phi(x_n)\right]$$

by chain rule

$$= \beta\sum_{n=1}^{N}\left(\underset{\text{scalar}}{t_n} - \underset{\text{scalar}}{w^T\phi(x_n)}\right)\underset{\text{vector size } M}{\phi(x_n)^T}$$ ✓

Step 3 Set grad=0, solve for w_{ML}

$$\vec{0} = \beta\sum_{n=1}^{N}(t_n - w^T\phi(x_n))\phi(x_n)^T$$

$$\vec{0} = \sum_{n=1}^{N} t_n\phi(x_n)^T - w^T\underbrace{\sum_{n=1}^{N}\phi(x_n)\phi(x_n)^T}_{M\times M \text{ matrix}}$$

$$\underset{1\times M}{+t\Phi} = \underset{1\times M}{+w^T}\underset{M\times M}{\Phi^T\Phi} \xrightarrow{\text{transpose both sides}} \Phi^T t = \Phi^T\Phi w$$

$$\boxed{w_{ML} = (\Phi^T\Phi)^{-1}\Phi^T t}$$

$M\times M$ $M\times N$ $N\times 1$

$= M\times 1$ ✓

Thus, the maximum likelihood estimator w_{ML} for parameter w is given by:

$$w_{ML} = \left(\Phi^T \Phi\right)^{-1} \Phi^T t$$

where
$$t = \begin{array}{cc} & 1 \\ N & \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix} \end{array} \qquad \Phi = \begin{array}{cc} & M \\ N & \begin{bmatrix} 1 & \phi_1(x_1) & \cdots & \phi_{M-1}(x_1) \\ \vdots & & & \\ 1 & \phi_1(x_N) & \cdots & \phi_{M-1}(x_N) \end{bmatrix} \end{array}$$

Only exists when inverse of $\Phi^T\Phi$ exists,
so that matrix must be full rank (rank M).
Often, so long as #datapoints > #features, we'll be in good shape.
$N > M$

If inverse does not exist, can't estimate a unique w_{ML}

What about ML estimate of β? Our precision parameter?

Same process (Step 1, 2, 3) yields:

$$\beta_{ML}^{-1} = \frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \left(t_n - w_{ML}^T \phi(x_n)\right)^2$$

Sum of squared error when we plug in ML estimate of w

Looks similar to σ^2_{ML} for general 1D Gaussians.

Penalty term goal: avoid weight coefficients with large values

Penalized ML Estimator

Suppose we optimize

$$\max_{w} \ \mathcal{L}(w, \beta) - \underbrace{\lambda \sum_{m=1}^{M} w_m^2}_{\text{penalty term}}$$

sum of squares, also could write $\sum_m w_m^2 = w^T w$ inner product of w with itself

with strength scalar $\lambda > 0$ controling strength of penalty

With this objective, we find

$$w^* = \left(\lambda I_M + \Phi^T\Phi\right)^{-1} \Phi^T t$$

this is always rank M and always invertible

$M \times M$ with λ on diag. all zero off diagonal $\begin{bmatrix} \lambda & & 0 \\ & \lambda & \\ 0 & & \ddots \; \lambda \end{bmatrix}$

Towards a full probabilistic model for regression

Goal:

All unknown parameters (weight vector $w \in \mathbb{R}^M$, precision $\beta^{-1} > 0$) are treated probabilistically.

For now, we'll assume $\beta^{-1} > 0$ is fixed known. Simpler.

Equivalently, we need to define a joint model

$$p(t, w \mid x, \beta)$$

$$= \underbrace{p(t \mid x, w, \beta)}_{\text{likelihood}} \cdot \underbrace{p(w \mid \beta)}_{\text{prior}}$$

Why? Given this joint, we can talk about posterior beliefs about parameters after seeing data

$$p(w \mid \{x_n, t_n\}_{n=1}^N, \beta)$$

- Can just take MAP estimate instead of ML (better with limited data)
- Can use samples from posterior to assess uncertainty

We can also make good predictions about new data

Use the predictive posterior.

$$p(t_* \mid x_*, \{x_n, t_n\}_{n=1}^N) = \int_w \int_\beta p(t_* \mid x_*, w, \beta)\, p(w \mid \{x_n, t_n\}, \beta)\, dw\, d\beta$$

From Joint Gaussian Distribution to Marginal + Conditional

We have vector $x = [x_1, x_2, x_3, \ldots x_{D-1}, x_D]$ in D dimensions

Consider <u>any</u> exclusive partition of indices into A and B

e.g. if $D = 4$, then $A = \{1\}$, $B = \{2,3,4\}$

or

$A = \{2,3\}$, $B = \{1,4\}$

Define $x \sim \mathcal{N}(\mu, \Sigma)$ assuming we've reordered dims so we count first thru A then B

this means: $\begin{bmatrix} x_A \\ \hline x_B \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_A \\ \hline \mu_B \end{bmatrix}, \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \right)$

$D \times 1$ vector

$D \times D$ cov matrix

Bishop Section 2.3.1 + 2.3.2 has lots of math deriving the following:

What is the conditional $p(x_A | x_B)$?

What is the marginal $p(x_A)$?

$p(x_B)$?

You should understand the math, but we'll skip math and show results

From Joint Gaussian to Marginal + Conditional

$$\begin{bmatrix} x_A \\ \hline x_B \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_A \\ \hline \mu_B \end{bmatrix}, \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \right)$$

whole covariance matrix
Σ is $D \times D$
has an inverse
$\Lambda = \Sigma^{-1}$
$= \begin{bmatrix} \Lambda_{AA} & \Lambda_{AB} \\ \Lambda_{BA} & \Lambda_{BB} \end{bmatrix}$

Marginal

$$p(x_A) = \int p(x_A, x_B)\, dx_B$$

$$= N(x_A \mid \mu_A, \Sigma_{AA})$$

Conditional

$$p(x_A \mid x_B) = \frac{p(x_A, x_B)}{p(x_B)}$$

$$= N\left(x_A \mid \mu_A - \Lambda_{AA}^{-1} \Lambda_{AB} (x_B - \mu_B),\ \Lambda_{AA}^{-1}\right)$$

dim check: $A{\times}1 - (A{\times}A)(A{\times}B)(B{\times}1 - B{\times}1)$, $A{\times}A$ ✓
$= A \times 1$ ✓