SPR Day 07

Probabilistic Linear Regression

Reading: Bishop PRML Sec 3.1

and Sec. 3.3

Outline:

(1) Linear Regression: A Probabilistic View

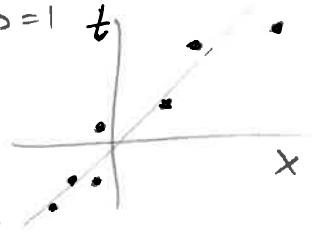
(2) ML estimator + Connections to "Least Squares"

(3) Gaussian-Gaussian regression model to give uncertainty to weights

(4) MAP estimator + Connections to "Penalized Least Squares" aka Ridge Regression

Probabilistic View of Linear Regression w/ Known Feature Transforms

Linear Regression — Standard Features



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 predict t_* given x_*

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

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

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

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

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

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

sometimes called "basis" function

M total entries, including "always 1" feature

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

x_{n1}^2 or $x_{n1} x_{n3}$ or $\cos(x_{n4})$ or ...

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

"Featurized" model for prediction:

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

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

Likelihood of observing output t_n given input x_n

$$p(t_n \mid x_n, w, \beta) = \mathcal{N}\left(t_n \mid \underbrace{w^T\phi(x_n)}_{\text{mean}}, \underbrace{\beta^{-1}}_{\text{variance}}\right)$$

$\mathcal{N}$: notation for Normal PDF

t_n: 1D variable in $\mathbb{R}$

β^{-1} is $\text{Var}[t_n]$

β is precision of t_n

If we assume all N observations are i.i.d. from this distribution

likelihood

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

Taking log of both sides and simplifying

sum of squared errors

$$\log p(t \mid X, w, \beta) = \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi) - \beta\,\frac{1}{2}\sum_{n=1}^{N}\left(t_n - w^T\phi(x_n)\right)^2$$

log likelihood

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

ML Estimation for weight vector + precision

View log likelihood as function of w, β, (step 1) then try to maximize (step 2) by taking gradients & setting equal to $\vec{0}$ & solving (step 3)

Step 1

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

Step 2 Gradient wrt w_{ML}

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

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

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

by chain rule

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

(t_n: scalar; $w^T\phi(x_n)$: scalar; $\phi(x_n)^T$: vector size M) ✓

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

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

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

($\sum_{n=1}^{N}\phi(x_n)\phi(x_n)^T$: M×M matrix)

$$+t\Phi = +w^T\Phi^T\Phi \longrightarrow \Phi^T t = \Phi^T\Phi w$$

(transpose both sides)

dimensions: $1\times M$ = $1\times M$ $M\times M$ → $M\times 1$ = $M\times M$ $M\times 1$

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

$M\times M$ $M\times N$ $N\times 1$ = $M\times 1$ ✓

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

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.

Penalized ML Estimator

Suppose we optimize (log likelihood; penalty term)

$$\max_w \; \alpha(w, \beta) + \lambda \sum_{m=1}^{M} w_m^2$$

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

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

With this objective, we find

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

this is always rank M and always invertible

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

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

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

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

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

We can also use posterior to make predictions about new data

Use the predictive posterior.

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

Key difference: Average over all w vectors, weighted by posterior density.
Do NOT just commit 100% to one w vector.

What is so special about the Gaussian?

Given

Two independent Gaussian r.v.s

$$x \sim \mathcal{N}(\mu_x, \sigma_x^2)$$
$$y \sim \mathcal{N}(\mu_y, \sigma_y^2)$$

Can Show That

Their joint is Gaussian

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \right)$$

Two linearly-dependent Gaussian r.v.s

$$x \sim \mathcal{N}(\mu_x, \beta_x^{-1})$$
$$y \sim \mathcal{N}(mx+b, \beta_y^{-1})$$

key: y only depends on x via mean which is linear func. of x

Their joint is Gaussian

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ m\mu_x + b \end{bmatrix}, \begin{bmatrix} \beta_x + \beta_y m^2 & -\beta_y m \\ -\beta_y m & \beta_y \end{bmatrix}^{-1} \right)$$

See PRML 2.103 and 2.113

A joint distribution over a partitioned vector

$$[-x-]^T = [-x_A-|-x_B-]^T$$

$$\begin{bmatrix} x_A \\ x_B \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}, \left[\begin{array}{c|c} \Sigma_{AA} & \Sigma_{AB} \\ \hline \Sigma_{AB}^T & \Sigma_{BB} \end{array}\right] \right)$$

Marginal is Gaussian PRML Eq. 2.98

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

Conditional is Gaussian PRML Eq. 2.96

$$p(x_A | x_B) = \mathcal{N}(\mu_{A|B}, \Lambda_{AA}^{-1})$$

$$\mu_{A|B} = \mu_A - \Lambda_{AA}^{-1} \Lambda_{AB} (x_B - \mu_B)$$

Two linearly dependent Gaussians

$$x \sim \mathcal{N}(\mu, \Lambda^{-1})$$
$$y|x \sim \mathcal{N}(Ax+b, L^{-1})$$

Posterior is Gaussian

$$p(x|y) = \mathcal{N}(\mu_{x|y}, \Sigma)$$

see formula in PRML Eq. 2.116

Posterior of Gaussian-Gaussian linear regression

The Gaussian-Gaussian model for regression

Assume $\beta^{-1} > 0$ known precision

Prior

$$p(w) = \mathcal{N}(m_0, S_0)$$

$m_0 \in \mathbb{R}^M$

S_0 is $M \times M$ covariance matrix (symmetric, pos. def.)

Likelihood

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

↖ linear function of w!

iid each example

Turns out, posterior is given by (see Bishop 2.116)

$$p(w \mid t, X, \beta) = \mathcal{N}(m_N, S_N)$$

where $m_N = S_N (S_0^{-1} m_0 + \beta \Phi^T t)$

$$S_N^{-1} = S_0^{-1} + \beta \Phi^T \Phi \quad \text{Eq. 3.50}$$

Check: dimensions,

what if no data?

What is the MAP estimator for linear regression?

$$w_{MAP} = \underset{w \in \mathbb{R}^M}{\operatorname{argmax}} \log p(w \mid \{x_n, t_n\}_{n=1}^N)$$

we've shown this is Gaussian: $N(m_N, S_N)$

thus,

$$w_{MAP} = m_N$$

Why? Use property that mode of any Gaussian is its mean parameter

$$= \left(S_0^{-1} + \beta \Phi^T \Phi\right)^{-1} \left(S_0^{-1} m_0 + \beta \Phi^T t\right)$$

Consider a prior that favors zero mean + diag. covariance

$$m_0 = \vec{0} \quad \text{and} \quad S_0 = \alpha^{-1} I_M = \begin{bmatrix} \alpha^{-1} & & \\ & \alpha^{-1} & \\ & & \ddots \end{bmatrix}$$

$$S_0^{-1} = \alpha I_M$$

then our MAP simplifies to

$$w_{MAP} = \left(\alpha I_M + \beta \Phi^T \Phi\right)^{-1} \beta \Phi^T t$$

$$= \frac{\beta}{\beta} \left(\frac{\alpha}{\beta} I_M + \Phi^T \Phi\right)^{-1} \Phi^T t$$

factor $\frac{1}{\beta}$ out of inverse

$$= \left(\frac{\alpha}{\beta} I_M + \Phi^T \Phi\right)^{-1} \Phi^T t$$

Looks familar! Equivalent to our sum-of-squares penalized linear regression w/ $\lambda = \frac{\alpha}{\beta}$