Sparse regression

Tuesday, February 27, 2024 9:05 AM

Today sparse regression and dictionary learning

Problem 1: Given time series data, can we fit dynamics to it in a Bayesian context?

Given $\vec{y}_d = \left(y^d_{t_1} \cdots y^d_{t_N} \right)$

Hypothesize $\begin{cases} \dot{y} = F(t, y \mid \theta) \\ y(t=0) = y_0 \end{cases}$

Minimize a norm $\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{L}\left(y - \left(y_0 + \int_0^t F(t) \right) \right)$

Problem 2 Assume direct measurements of relevant DOFs z are unavailable

$$z = G(y \mid \theta_1)$$

$$\begin{cases} \dot{z} = F(z, t \mid \theta_2) \\ z(t=0) = G(y(t=0)) \end{cases}$$

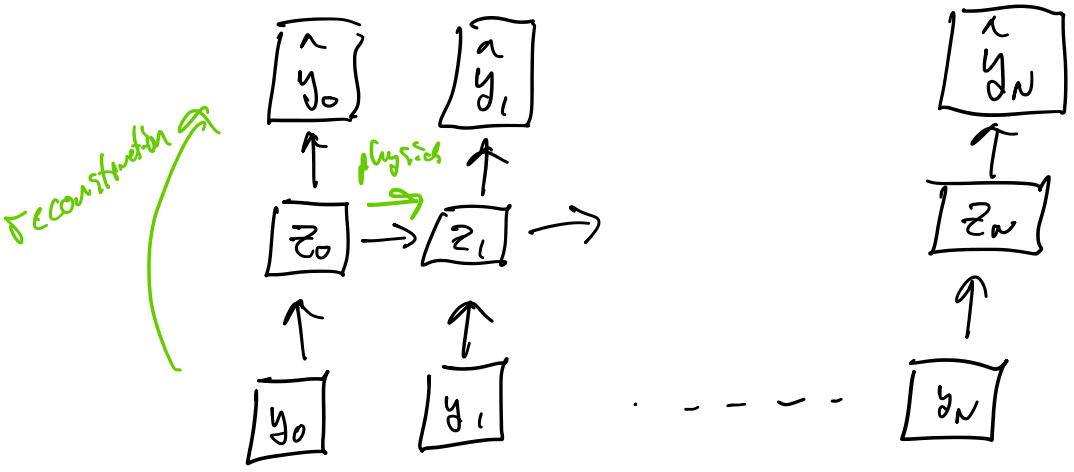
examples Image-based simulation

Candidate architectures

Candidate [illegible]

G → auto encoders

F → neural ODEs, universal differential eqns, dictionary learning



We'll start w/ one simple strategy called SINDy

"Sparse Identification of nonlinear dynamics" 2016 Brunton Proctor Kutz

IDEA $\dot{x} = \Theta(x)\,\xi$

$\Theta(x)$: dictionary; ξ: unknown coefficients

SPARSITY

Want to only choose a few terms, so data reveals governing ODE

"Occams Razor", Parsimony

Sparse Reg vs Polynomial Fitting

Not combining basis functions to approximate more complicated ones

ex images of pendulum swinging

Know $\ddot{z} = -kz$ is model

Can we infer θ coordinate from images?

Necessary Background

Revisiting overfitting (Murphy §4.5, § 11.3)

$$\mathcal{L}(\theta;\lambda) = \left[\frac{1}{N}\sum_{i=1}^{N} l(y_i, \theta; x_i)\right] + \lambda C(\theta)$$

(Reg. Param; complexity)

Typical Bayesian setting

$l = \log p(y_i|x_i,\theta)$

$C(\theta) = \log p(\theta)$ ← prior

def Maximum a posteriori (MAP) estimation

$$\hat{\theta} = \arg\max_{\theta} \log p(\theta|\mathcal{D})$$

$$\hat{\theta} = \arg\max_{\theta} \log \left(\frac{p(\mathcal{D}|\theta)\, p(\theta)}{p(\mathcal{D})} \right)$$

$$\hat{\theta} = \arg\max_{\theta} \left[\log p(\mathcal{D}|\theta) + \log p(\theta) \right]$$

likelihood, prior

note the unknown marginal likelihood $p(\mathcal{D})$ drops out

Bayesian regression revisited

Approximate $y(x) = \sum_i \phi_i(x)\, c_i$

$$= \Phi(x)^T c$$

Model $p(y|x,c) = N(y;\ \mu = \Phi(x)^T c,\ \sigma^2)$

$$\mu_{MLE} = \Phi(x) \left(\Phi(x)\Phi(x)^T\right)^{-1} \Phi(x)\, y$$

Ridge Regression

Adopt the prior

$$p(c) = N(c;\ 0,\ \lambda^{-1} I)$$

$$\mathcal{L} = \log p(y|x,c)\, p(c) =$$

$$\log N(y;\ \Phi(x)^T c,\ \sigma^2)\, N(c;\ 0,\ \lambda^{-1} I)$$

$$= \text{const} - \frac{1}{2} \log \sigma^2 - \frac{1}{2} \left(\Phi(x)^T c - y \right)^2 - \frac{1}{2} c^2$$

$$= \sum_d -\frac{1}{2}\log\sigma^2 - \frac{1}{2}\left(\frac{P(x_d)^T c - y_d}{\sigma}\right)^2 - \frac{1}{2}\frac{c^2}{\lambda^2}$$

$$0 = \nabla_c \mathcal{L} = \nabla_c \sum_d \left(\frac{\| P(x_d)^T c - y_d \|^2}{\sigma^2} + \frac{1}{\lambda^2}\|c\|^2 \right)$$

$$0 = \sum_d 2\left(P(x_d)^T c - y_d\right) P(x_d) + 2\left(\frac{\sigma}{\lambda}\right)^2 c$$

$$\sum_d \left(P(x_d) P(x_d)^T + \frac{\sigma^2}{\lambda^2} I \right) c_{MAP} = \sum_d P(x_d)\, y_d$$

↑ ridge stabilization

- penalizing magnitude of coefficients
- aka - ridge regression
  - l_2 regularization
  - weight decay
- Gives invertible matrix in small data limits

For further reading see Murphy §11.7
- Conjugate priors
- Bayesian Linear regression

Lasso / L^1 regression for sparsity

- many applications want coefficients to not just be small but as few nonzero as possible

not just be small but as few nonzero entries as possible

Goal

L_0 norm $\|w\|_0 = \sum_{d=1}^{D} \mathbb{1}_{|w_d|>0}$ minimization

- Important for feature selection & dictionary learning

Least Absolute shrinkage + selection operator (Tibshirani 96)

$$\mathcal{L}(w) = -\log p(\mathcal{D}|w) - \log(w|\lambda)$$

If

$$w|\lambda \sim \prod_d \text{lap}(w_d \mid \mu=0, b=\lambda^{-1})$$

$$\text{lap}(w|\mu,b) = \frac{1}{2b}\exp\left(-\frac{|w-\mu|}{b}\right)$$

Then $\mathcal{L}(w) = \|Xw-y\|^2 + \lambda\|w\|_1$

Note that for other norms $\|w\|_p = \left(\sum |w|^p\right)^{1/p}$ we get sparser solutions, but $p<1$ is nonconvex

- ℓ_1-norm is tightest convex relaxation of ℓ_0-norm

Why sparse?

$$\min_w -\log p(y|w) + \lambda\|w\|_1$$

is Lagrangian associated w/

$$\min_w -\log p(y|w)$$

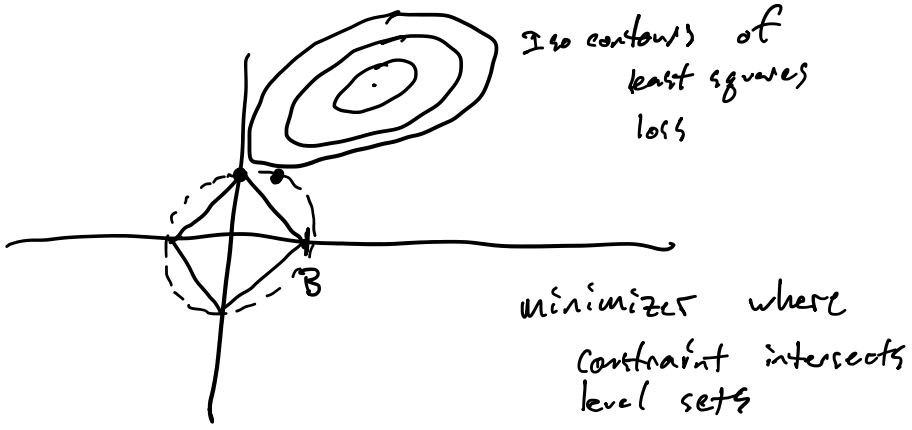
$$\min_w \; -\log p(y|w)$$

$$\text{s.t.} \quad \|w\|_1 \leq B$$ ← parameter imposing tightness of penalty

Compare to ridge regression

$$\min \; -\log p(y|w)$$

$$\text{s.t.} \quad \|w\|_2 \leq B$$



Iso contours of least squares loss

minimizer where constraint intersects level sets

Issue Gradient undefined precisely where entries get sparse

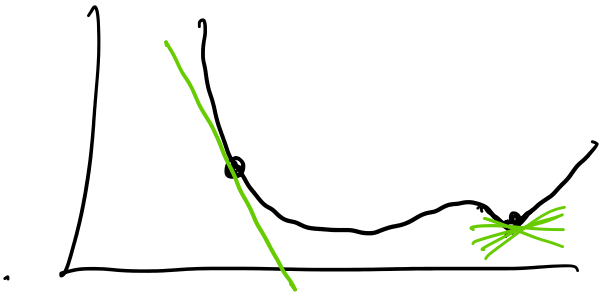
Def Subgradient

$f: \mathbb{R}^n \to \mathbb{R}$, $g \in \mathbb{R}^n$ is a subgradient if for all $z \in \text{dom}(f)$

$$f(z) \geq f(x) + g^T(z-x)$$

← think of like first order Taylor series

Denote $g(x) = \partial f(x)$



At cont. pt, one unique subdifferential
At a kink, many

ex $f(x) = abs(x)$

$$\partial f(x) = \begin{cases} -1 & x < 0 \\ [-1, 1] & x = 0 \\ 1 & x > 0 \end{cases}$$

any number between $[-1, 1]$

Back to lasso (summary of results, see 11.4.3 for details)

$$\mathcal{L}(w) = \underbrace{\|Xw - y\|_2^2}_{①} + \lambda \underbrace{\|w\|_1}_{②}$$

One can show

$$\frac{\partial}{\partial w_d} ① = a_d w_d - c_d$$

$$a_d = \sum_{n=1}^{N} x_{nd}^2$$

intercept as residual → $$c_d = \sum_{n=1}^{N} x_{nd} \left(y_n - \vec{w}_{-d}^T \vec{x}_{n,-d} \right)$$

where $\vec{z}_{-d} = \langle z_1, \ldots z_{d-1}, z_{d+1}, \ldots z_m \rangle$

Taking subderivative to acct for discont.

$$\partial_{w_d} (① + ②) = (a_d w_d - c_d) + \lambda \partial_{w_d} \|w\|_1$$

$$= \begin{cases} a_d w_d - c_d - \lambda & \text{if } w_d < 0 \\ [-c_d - \lambda, -c_d + \lambda] & \text{if } w_d = 0 \\ a_d w_d - c_d + \lambda & \text{if } w_d > 0 \end{cases}$$

At $0 = 2a_d(① + ②)$ we have 3 solutions

$$\hat{w}_d(c_d) = \begin{cases} (c_d + \lambda)/a_d & c_d < -\lambda \\ 0 & c_d \in [-\lambda, \lambda] \\ (c_d - \lambda)/a_d & c_d > \lambda \end{cases}$$

Compactly

$$\hat{w}_d = \text{SoftThreshold}\left(\frac{c_d}{a_d}, \lambda/a_d\right)$$

$$\text{SoftThreshold}(x, \delta) = \text{sign}(x)\left(|x| - \delta\right)_+$$

First introduction to Markov Chains

Consider time series data

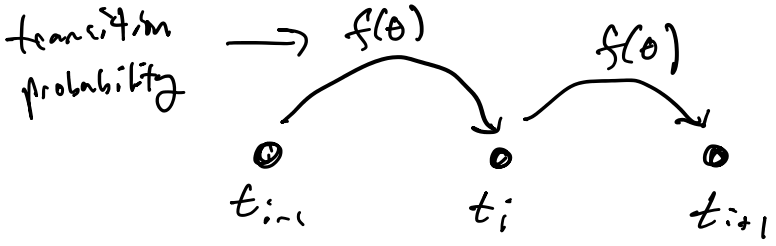
$$\vec{y}_d = \left(y_{t_1}^d, \ldots y_{t_N}^d\right)$$

By chain rule could write

$$p(\vec{y}) = p(y_{t_1})\, p(y_{t_2} | y_{t_1})\, p(y_{t_3} | y_{t_2}, y_{t_1}) \cdots$$
$$= \prod_{i=1}^{N} p(y_{t_i} | y_{t_{1:i-1}})$$

For first-order Markov process, assume

$$p(y_{t_i} \mid y_{t_{1:i-1}}) = p(y_{t_i} \mid y_{t_{i-1}}) = f(\theta)$$



So $$p(\vec{y}_d) = p(y_1) \prod_{i=2}^{n} p(\theta_i \mid y_{i-1})$$

initial condition

Note as a special case a stochastic differential equation

$$dX = f(x)\,dt + g(x)\,dB_t$$

Euler-Maruyama $$X^{n+1} - X^n = f(x^n)\Delta t + g(x^n)\xi$$

$$\xi \sim N(0, \Delta t)$$

then $$p(x^{n+1} \mid x^n) = N\left(x^n + \Delta t\, f(x^n),\ \Delta t\, g(x^n)^2\right)$$