Lecture 3

Monday, January 29, 2024 9:34 AM

New homework assignment posted to canvas

Due 2/8

Plan for today
- Mixture models
  ↳ over fitting
- Regression theory
  - finite elements
  - local polynomial reproduction
  - curse of dimensionality
  - intro to DNNs ← no black boxes!

Some final comments on building probability distributions

Multivariate Gaussian

$y \in \mathbb{R}^N$

$y \sim N(\mu, \Sigma)$ (det)

$$p(y) = 2\pi^{-N/2} |\Sigma| \exp\left[-\frac{1}{2}(y-\mu)^T \Sigma^{-1} (y-\mu)\right]$$

ex If $\Sigma = \text{diag}(\sigma_i^2)$ show that y may be viewed as joint distribution of indep univariate gaussians

Mixture of experts

Let $C \sim \text{categorical}(\pi)$

$\text{cat}(\pi)$

$0 \le \pi \le 1$

$\sum_i \pi_i = 1$

Define MOE

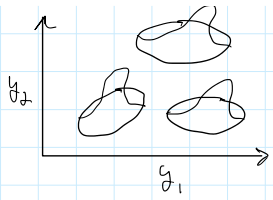
$$p(y|x) = \sum_c \underbrace{p(y|c,x)}_{\text{expert model}} \underbrace{p(c|x)}_{\text{gating function}}$$

IDEA: Tool For Blending Multiple Models Potentially Allowing Specialization

Example Gaussian Mixture model

$$p(y) = \sum_{c=1}^{N_c} p(c)\, p(y|c)$$

$$\approx \sum_{c=1}^{N_c} \pi_c\, N(y; \mu_c, \Sigma_c)$$



Some exercises

- Compute $p(c|y)$

$$p(c|y) = \frac{p(y|c)\,p(c)}{p(y)} \qquad \text{Bayes Thm}$$

$$= \frac{p(y|c)\,p(c)}{\sum_{c'} p(y|c')\,p(c')} \qquad \text{Total Probability}$$

$$= \frac{\pi_c\, N(y;\mu_c,\Sigma_c)}{\sum_{c'} \pi_{c'}\, N(y;\mu_{c'},\Sigma_{c'})}$$

Remark → Mixture models allow Bayesian inference of cluster ownership

Exercises Compute $\mathbb{E}_{c\sim p(c|y)}[c]$

$\mathrm{Var}_{c\sim p(c|y)}[c]$

Overfitting in MLE

Consider fitting two clusters to scattered data w/ max. likelihood

$$NLL(\pi,\mu,\Sigma) = \log p(y_1,\dots,y_d) = \sum_{i=1}^{d} \log p(y_d)$$

$$= \sum_{i=1}^{d} \log \sum_{c} \pi_c\, p(y_d|c)$$

Remark: First time we've gotten log of sum → not nice for optimization

Take $\pi_{c_1} = 1$

$$\mu_{c_1} = y_{d_1}$$
$$\Sigma_{c_1} = \varepsilon^2 I << 1$$

Then $NLL = -\sum_{i=1}^{d} \log N(y_d; y_{d_{11}}, \varepsilon^2 I)$

$$= -\log N(y_{d_1}; y_{d_1}, \varepsilon^2 I) - \sum_{i \neq 1}^{d} \log N(y_d; y_{d_1}, \varepsilon^2 I)$$

$$= C + \frac{d}{2} \log \varepsilon + \frac{1}{2} \frac{\cancel{0}}{\varepsilon^2} + \text{STUFF}$$

Note that $\lim_{\varepsilon \to 0} NLL = -\infty$

So that we can make the NLL arbitrarily small by attaching a cluster to a single point

Overfitting is a danger of MLE

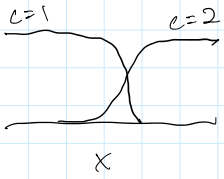
Soln. Stabilization for GMM

$$N(y | \mu_c, \Sigma_c + \varepsilon^2 I)$$

$\varepsilon^2 > 0$ fixed Background Noise

Another MoE example — Piecewise Polynomial Regression

$$p(y|x) = \sum_c p(y|c,x)\, p(c|x)$$



$$p(c|x) = \text{cat}(\sigma(wx+\beta))$$
$$p(y|c,x) = N(y; A_c x + B_c, \sigma_c^2)$$

Classical Regression — Introductory Functional Analysis

Let $L_2(\Omega) = \{ f \mid \int_\Omega f^2 dx < \infty \}$

$C_1(\Omega) = \{ f \mid$

Finite Elements Let $0 = x_0 < \cdots < x_N = 1$

Let $\mathcal{S}$ be a linear space of functions $\mathcal{S} = \text{span}(\{v_1, \ldots, v_N\})$ satisfying

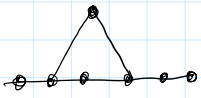
1) $v \in C^0([0,1])$ continuous on unit interval

2) $v|_{[x_i, x_{i+1}]}$ is a linear polynomial

3) $v(0) = 0$

Define $\phi_i(x)$ by requiring "Kronecker delta" property

$$\phi_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$$



ML def of PL functions

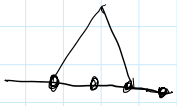
Recall $\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & \text{else} \end{cases}$

ex Express $\phi_i(x)$ in terms of ReLU

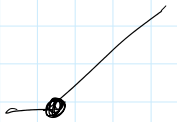
$$\phi_i(x) = A\,\text{ReLU}(x - x_{i-1}) + B\,\text{ReLU}(x - x_i) + C\,\text{ReLU}(x - x_{i+1})$$

$\phi_i(x_j) = \delta_{ij}$ implies

$$1 = \phi_i(x_i) = A(x_i - x_{i-1}) + 0 + 0$$
$$0 = \phi_i(x_{i+1}) = A(x_{i+1} - x_{i-1}) + B(x_{i+1} - x_i) + 0$$
$$0 = \phi_i(x_{i+2}) = A(x_{i+2} - x_{i-1}) + B(x_{i+2} - x_i) + C(x_{i+2} - x_{i+1})$$

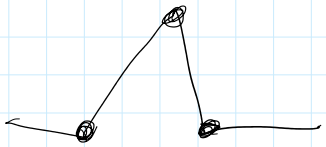


$$A = \frac{1}{x_i - x_{i-1}} \qquad B = \frac{-A(x_{i+1} - x_{i-1})}{x_{i+1} - x_i} \qquad C = \frac{-A(x_{i+2} - x_{i-1}) - B(x_{i+2} - x_i)}{x_{i+2} - x_{i+1}}$$



Activation 1

Activation 2



Activation 3

Lemma ϕ_i form a basis for S

Pf - ϕ_i is a basis if $\sum c_i \phi_i(x_j) = 0$ implies $c_i = 0$ for all i

- From Kronecker-δ property

$$\sum_i c_i \phi_i(x_j) = \sum_i c_i \delta_{ij}$$

$= c_j = 0$

Thm $\mathcal{S} \subseteq L^2([0,1])$ if

$f \in$

Define The interpolant $v_I \in \mathcal{S}$ satisfying

$$v_I(x_j) = v(x_j) \quad \text{for all } j$$

is $$v_I(x) = \sum_i v(x_i)\,\phi(x_i)$$

Pf $$v_I(x_j) = \sum_i c_i\,\phi_i(x_j)$$

$$= \sum_i c_i\,\delta_{ij}$$

$$= c_j$$

Thm Let $h = \max_i |x_{i+1} - x_i|$. If u'' defined

Then $$|u(x) - u_I(x)| \le \frac{1}{2} h^2 \max_{x \in [0,1]} |u''(x)|$$

Pf for $x \in [x_i, x_{i+1}]$

- $u(x) - u_I(x) = \int_{x_i}^{x} (u - u_I)'(t)\,dt$ Fundamen Thm Calcu
- mean value thm $\Rightarrow$ can find $x^* \in [x_i, x_{i+1}]$ s.t.

$$u'(x^*) = \frac{u(x_{i+1}) - u(x_i)}{x_{i+1} - x_i} = u_I'$$

so $$(u - u_I)'(x^*) = 0$$

F.T.C. $(u-u_I)'(t) = \int_{x^*}^{t} (u-u_I)''(s)$

$$= \int_{x_i}^{x} \int_{x^*}^{t} (u-u_I)''(s)\, ds\, dt$$

$$= \int_{x_i}^{x} \int_{x^*}^{t} u''(s)\, ds\, dt$$

$$|u-u_I|(x) \le \max_{s\in[x_i, x_{i+1}]} u''(s) \int_{x_i}^{x} \int_{x^*}^{t} ds\, dt$$

$$|u-u_I| \le \frac{1}{2} h^2 \left(\max_{[0,1]} u'' \right)$$ ∎

We can understand this informally

→ the interpolant can match the constant + linear term of the Taylor series on each element. Integrating the remainder on the element gives h^2 scaling

Exercise Develop higher-order polynomial spaces using ReLU networks (HW problem)

First glimpse at DNN

ds

an
alue
heorem

(
$= 0$

Def A shallow neural network of width N

$$NN: \mathbb{R}^{N_{in}} \to \mathbb{R}^{N_{out}}$$

with ReLU activation consists of a "hidden layer"

$$L_H(x) = \text{ReLU}(Wx + b)$$

$$W \in \mathbb{R}^{N \times N_{in}}, \quad b \in \mathbb{R}^N$$

composed w/ a "linear layer"

$$L_L(x) = Cx$$

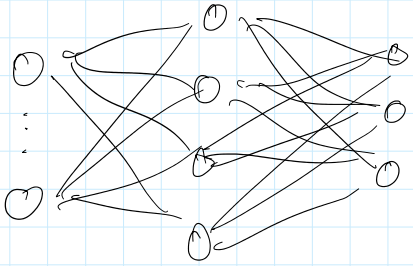
$$C \in \mathbb{R}^{N_{out} \times N}$$

$$NN_\theta(x) = L_L \circ L_H(x)$$

and we denote trainable params

$$\theta = \{W, B, C\}$$

We draw this as a network w/ weights + biases represented as edges



crude model of brain

Some history This is an example of a

Some history

This is an example of a "Multilayer Perceptron" (MLP)

More generally a deep NN / dense / feed forward can consist of many layers and other nonlinear activation functions

$$y(x) = L_L \circ L_{H_M} \circ \cdots \circ L_{H_1}(x)$$

$$L_{H_i}(x) = \sigma(W_i x + b_i)$$

σ: activation, W_i: weight, b_i: bias

1958 - single hidden, trainable linear only

1970 - first backprop

2003 - renewed interest due to architectures

Thm Consider $N_{in} = N_{out} = 1$ (1D scalar data)

$$\mathcal{D} = (x_i, y_i)_{i=1}^{N_D} \qquad N_D = N - 1$$

satisfying $x_i \neq x_j \quad \forall i, j$

then

$$\min_{\theta} \sqrt{\frac{1}{N_D} \sum_{i=1}^{N_D} |NN_\theta(x_i) - y_i|^2} \leq C h^2$$

Pf Set: biases to match nodes
weights to 1
linear layer to interpolant of data

— This is drastically over simplistic

- Does an interpolant exist for arbitrary scattered data?
- How does that depend on dimension
- A NN can emulate PW linears in 1D — what about higher dimensions?
- When can a NN beat a traditional finite element space?
- How do we initialize a DNN when training w/ gradient descent?

Understanding the curse of dimensionality
(See "scattered data approximation")

(See "Scattered data approximation" By Holger Wendland)

- Consider a scattered dataset

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^{N} \qquad x \in \mathbb{R}^d, \; y \in \mathbb{R}$$

- Aim to build a "quasi-interpolant" of $\mathcal{D}$

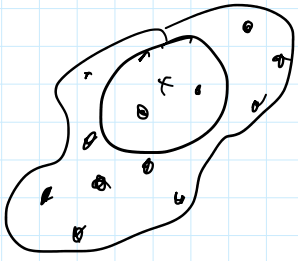
$$s(x) = \sum_{j=1}^{N} y_j \, u_j(x)$$

(u_j: ← TBD)

where $s \approx y$ at x

- Def fill-distance

$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{1 \le j \le N} \|x - x_j\|_2$$



the largest distance one needs to go to find a data pt.

- def A local polynomial reproduction of degree ℓ assigns u_j s.t.

polynomial reproducing

(1) $\sum_j p_j \, u_j(x) = p(x) \quad \forall \, p \in \Pi_\ell(\mathbb{R}^d)$

"not too big"

(2) $\sum_j |u_j(x)| < C_1 \quad \forall \, x \in \Omega$

compact support

(3) $u_j(x) = 0$ if $\|x - x_j\|_2 \ge C_2 \, h_{X,\Omega}, \; \forall \, x \in \Omega$

compact support

(3) $u_j(x) = 0$ if $\|x - x_j\|_2 \geq C_2 h_{x,\Omega}$, $\forall x \in \Omega$

For a polynomial reproduction of order m on a compact Ω

$$|f(x) - s(x)| \leq C h_{x,\Omega}^{m+1} |f|_{C^{m+1}}$$

where $|f|_{C^{m+1}} = \max_{|\alpha|=m+1} \|D^\alpha f\|_{L^\infty}$

sketch proof

Pf Let $p \in \Pi_m$ be a polynomial of ~~order~~ m. Then

$$|f(x) - s(x)| \leq |f(x) - p(x)| + |p(x) - \sum_j f_j u_j(x)|$$

$$= \quad " \quad + |\sum p_j u_j(x) - f_j u_j(x)|$$

$$\leq |f(x) - p(x)| + \sum |p_j - f_j| |u_j(x)|$$

$$\leq \|f - p\|_{L^\infty(B(x, C_2 h_{x,\Omega}))} \left(1 + \sum_j |u_j(x)|\right)$$

$$\leq (1 + C_1) \|f - p\|_{L^\infty(B(x, C_2 h_{x,\Omega}))}$$

Taking p as the Taylor polynomial

$$\|f - p\|_{L^\infty(B(x, C_2 h_{x,\Omega}))} \leq C h_{x,\Omega}^{m+1} |f|_{C^{m+1}}$$

@ So error depends on scaling of $h_{X,\Omega}$

→ how does that depend on dimension?

<u>def</u> separation distance $q_X = \frac{1}{2} \min_{i \neq j} \|x_j - x_i\|$

<u>def</u> data sites X are quasi-uniform if

$$q_X < h_{X,\Omega} < c_{qu} q_X \quad , \quad c_{qu} > 0$$

<u>informally</u> → $q_X \sim h_{X,\Omega}$

<u>thm</u> Given N quasi-uniform data sites

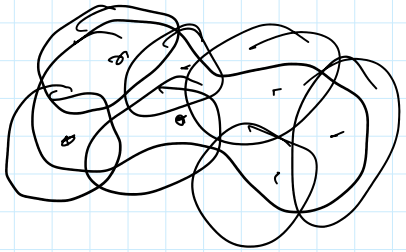
$$c N^{-1/d} \leq h_{X,\Omega} \leq C N^{-1/d}$$

<u>Pf</u> (⇒) $\Omega \subseteq \bigcup_{i=1}^{N} B(x_i, h_{X,\Omega})$

$$\text{vol}(\Omega) \leq \sum \text{vol}\left(B(x_i, h_{X,\Omega})\right)$$

$$\leq C N h^d$$

$$N^{-1/d} < C h$$

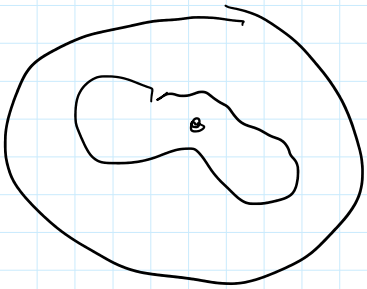


(⇐) Since Ω bounded, there is a x_0, R s.t.

Since Ω bounded, there

is a x_0, R s.t

$$\Omega \subseteq B(x_0, R)$$



$$\bigcup_{i=1}^{N} B(x_i, q_x) \subseteq \Omega$$

disjoint balls

$$N q_x^d \leq \mathrm{vol}(\Omega)$$

$$q_x \leq C N^{1/d}$$

quasi-uniformity $\quad h_{x,\Omega} \leq C N^{-1/d}$

Punchline To hit a given error

$$\|u - s\| \sim C h^{m+1} \sim C N^{-\frac{m+1}{d}}$$