# MLPR
## final notes

linear regression

total square error: $E(\underline{w}, b) = (\underline{y} - \underline{f})^T(\underline{y} - \underline{f})$

can add bias as extra column of 1's to $X$ and a $w_0$ to $\underline{w}$

RBS form: $\exp\left(-(\underline{x} - \underline{c})^T(\underline{x} - \underline{c})/h^2\right)$

center

width ($h$)

regularization:

$$E_\lambda(\underline{w}; \underline{y}, \Phi) = (\underline{y} - \Phi\underline{w})^T(\underline{y} - \Phi\underline{w}) + \lambda\underline{w}^T\underline{w}$$

$$= (\underline{y}' - \Phi'\underline{w})^T(\underline{y}' - \Phi'\underline{w}) \quad \text{where } \underline{y}' = \begin{bmatrix} \underline{y} \\ \underline{0}_k \end{bmatrix}, \; \Phi' = \begin{bmatrix} \Phi \\ \sqrt{\lambda}\,\mathbb{I}_k \end{bmatrix}$$

model evaluation:

gen. error $E = \mathbb{E}_{p(\underline{x}, y)}\left[L(y, f(\underline{x}))\right] = \int L(y, f(\underline{x}))\, p(\underline{x}, y)\, d\underline{x}\, dy$

use mean test error: $\frac{1}{M}\sum_{m=1}^{M} L\left(y^{(m)}, f(\underline{x}^{(m)})\right) \quad \underline{x}^{(m)}, y^{(m)} \sim p(\underline{x}, y)$

univariate Gaussian: $p(z) = N(z, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2\sigma^2}(z-\mu)^2\right)$

error bars: $\text{var}[\bar{x}] = \frac{\sigma^2}{N}$

typical deviation: $\mu \pm \frac{\hat{\sigma}}{\sqrt{N}}$ $\Rightarrow$ can apply this to errors in test set

multivariate Gaussians

$$\begin{aligned} \text{cov}[\underline{y}] &= \mathbb{E}[\underline{y}\underline{y}^T] - \mathbb{E}[\underline{y}]\mathbb{E}[\underline{y}]^T \\ &= \mathbb{E}[A\underline{x}\underline{x}^T A] - \mathbb{E}[A\underline{x}]\mathbb{E}[A\underline{x}]^T \quad \text{if } \underline{y} = A\underline{x} \text{ and } x \sim N(0, 1) \\ &= A\,\mathbb{E}[xx^T]A^T \\ &= AA^T \\ &\equiv \Sigma \end{aligned}$$

note on determinants: $|\Sigma| = |AA^T| = |A||A^T| = |A|^2$

$$p(\underline{z}) = N(\underline{z}; \mu, \Sigma) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{D/2}}\exp\left(-\frac{1}{2}(\underline{z} - \mu)^T\Sigma^{-1}(\underline{z} - \mu)\right)$$

Classification

Bayes classifier:

$$\begin{aligned} P(y = k \mid \underline{x}) &\propto p(\underline{x} \mid y)\, P(y = k) \\ &\propto N(\underline{x}; \mu_k, \Sigma_k)\, \pi_k \quad \text{where } \pi_k = \frac{\sum_n \mathbb{I}(y^{(n)} = k)}{N} \\ &= s_k \end{aligned}$$

$$P(y = k \mid \underline{x}, \theta) = \frac{s_k}{\sum_{k'} s_{k'}}$$

"Naive Bayes": features are independent, i.e., $\Sigma_k$ is diagonal for all $k$

natural when features $\underline{x}$ are binary:

$$P(\underline{x} \mid y = k, \theta) = \prod_d P(x_d \mid y = k, \theta) = \prod_d \theta_{d,k}^{x_d}(1 - \theta_{d,k})^{1 - x_d}$$

regression and gradients:

calculate square error: $\underline{r}^T\underline{r} = (\underline{y} - X\underline{w})^T(\underline{y} - X\underline{w})$

then take deriv: $\nabla_{\underline{w}} \underline{r}^T\underline{r} = -2X^T\underline{y} + 2X^TX\underline{w}$

now can do:

closed form: $\underline{w} = (X^TX)^{-1}X^T\underline{y}$ (after setting $\nabla_{\underline{w}}\underline{r}^T\underline{r}$ to zero)

iterative: $\underline{w} \leftarrow \underline{w} - \eta \nabla_{\underline{w}}[\underline{r}^T\underline{r}]$

logistic regression:

$$NLL = -\sum_{n=1}^{N} \log\left[\sigma(\underline{w}^T\underline{x}^{(n)})^{y^{(n)}} (1-\sigma(\underline{w}^T\underline{x}^{(n)}))^{1-y^{(n)}}\right]$$

$$= -\sum_{n=1}^{N} \log \sigma(z^{(n)} \underline{w}^T\underline{x}^{(n)})$$

$$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1-\sigma(a))$$

$$\nabla_{w} NLL = -\sum_{n=1}^{N} (1-\sigma_n)\, z^{(n)}\underline{x}^{(n)}$$

approx grad: $\frac{\partial f(w)}{\partial w} \approx \frac{f(w+\frac{\varepsilon}{2}) - f(w - \frac{\varepsilon}{2})}{\varepsilon}$ (i.e., $\frac{\Delta y}{\Delta x}$)

softmax / robust solutions:

$$f_k = \frac{\exp(\underline{w}^{(k)T}\underline{x})}{\sum_{k'} \exp(\underline{w}^{(k')T}\underline{x})} \qquad \nabla_{w^{(k)}} \log f_c = (y_k - f_k)\underline{x}$$

a robust model randomly sets $y$ to $\frac{1}{2}$ w/ probability $\varepsilon$

$\begin{cases} \text{exp: real} \to \text{pos} \\ \text{log: pos} \to \text{real} \end{cases}$

$\begin{cases} \text{sigmoid: real} \to (0,1) \\ \text{logit: } (0,1) \to \text{real} \end{cases}$

neural nets

don't set weights to zero or to $\mathcal{N}(0,1)$ $\Rightarrow$ set as $\frac{0.1\ \mathcal{N}(0,1)}{\sqrt{K}}$

chain rule: $\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x}\frac{dx}{dr} + \frac{\partial f}{\partial y}\frac{dy}{dr}$

$$\frac{\partial f}{\partial u_i} = \sum_{d=1}^{D} \frac{\partial f}{\partial x_d}\frac{x_d}{\partial u_i} \quad \text{(generalized)}$$

$$\frac{\partial f}{\partial X_{ab}} = \bar{X}_{ab}$$

automatic differentiation: $Z = XY \Rightarrow \bar{X} = \bar{Z}Y^T$ and $\bar{Y} = X^T\bar{Z}$ (ordinary matrix operations: not $O(M^4)$!)

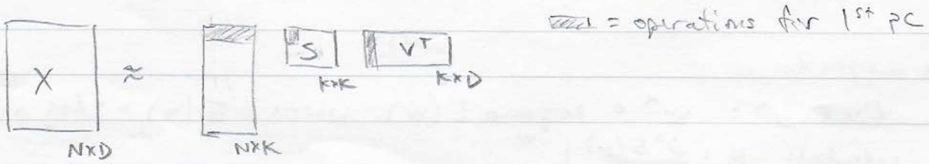
backprop the error signals: $\delta_i^{(l)} = \frac{\partial E}{\partial a_i^{(l)}} = \bar{a}_i^{(l)}$

MLPR
final notes

PCA

denoising autoencoder: essentially dropout

$X \approx USV^T$




$US \approx XV$ is data projected down to k dims

$USV^T \approx XVV^T$ is data projected back up (but it's low-rank k)

Bayesian regression:

- conjugate prior is one where the product of the prior and the likelihood combines to give a dist w/ the same functional form as the prior

$$p(\underline{w}\,|\,D) \propto p(D\,|\,\underline{w})\,p(\underline{w})$$

$$\propto \prod_n N(y^{(n)};\, \underline{v}^T \underline{\phi}(x^{(n)}),\, \sigma^2)\, N(\underline{w};\, 0,\, \sigma^2_{prior})$$ assuming we're updating a Gaussian and have a Gaussian prior

Bayesian inference and prediction:

prediction: $p(y\,|\,\underline{x}, D) = \int p(y, \underline{w}\,|\,\underline{x}, D)\, d\underline{w}$

$$= \int p(y\,|\,\underline{x}, \underline{w})\, \underbrace{p(\underline{w}\,|\,D)}_{\text{see last section}}\, d\underline{w}$$

(posterior)

$p(\underline{w}\,|\,D) = N(\underline{w};\, w_N, V_N)$ if Gaussian

linear and Gaussian models can be solved in closed form

for linear: $\mu_f = E_{\underline{w}}[\underline{x}^T \underline{w}] = x^T w_N$

$\text{var}[f] = \underline{x}^T V_N \underline{x}$ $\quad \left(= E\left[(\underbrace{x^T\underline{w} - x^T w_N}_{f - \mu_f})^T (\underbrace{x^T\underline{w} - x^T w_N}_{f - \mu_f})\right]\right)$

$$p(y\,|\,D, \underline{x}) = N(y;\, \underline{x}^T w_N,\, \underline{x}^T V_N x + \sigma^2)$$

add this for noise going from f to y

Bayesian model selection:

$$p(y\,|\,X, M) = \int p(y\,|\,X, \underline{w}, M)\, \overbrace{p(\underline{w}\,|\,M)}^{\text{small for overly complex models}}\, d\underline{w}$$

can do marginal likelihood to estimate hyperparams (e.g., $\underset{\frac{1}{\sqrt{\alpha}}}{\sigma^2_{prior}}$ and $\underset{\text{"}\sigma\text{" here}}{\sigma^2_y}$)

$$p(\underline{y}\,|\,X, \alpha, \sigma) = \int p(y\,|\,X, \underline{w}, \sigma)\, p(\underline{w}\,|\,\alpha)\, d\underline{w}$$ maximize this to find $\alpha$ and $\sigma$

GPs:

joint dist: $p\left(\begin{bmatrix} \underline{y} \\ \underline{f}_* \end{bmatrix}\right) = N\left(\begin{bmatrix} \underline{y} \\ \underline{f}_* \end{bmatrix};\, \underline{0},\, \begin{bmatrix} K + \sigma_n^2 \mathbb{I} & K_* \\ K_* & K_{**} \end{bmatrix}\right)$

$$k(x^{(i)}, x^{(j)}) = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D} (x_d^{(i)} - x_d^{(j)})^2 / \ell_d^2\right)$$

$\sigma_f$ ↳ typical vertical height

$\ell_d$ ↳ "typical dist. b/t turning points"

can compute hyperparams by max likelihood

$\max \log(\underline{f}\,|\,X, \sigma) = \max \log N(\cdot)$

or by integrating $p(\underline{f}_*\,|\,\underline{y}, X) =$ marginalize out $\sigma$...

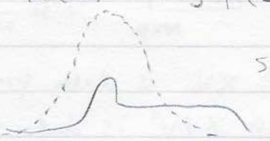
Bayesian logistic regression and Laplace approx

$$\text{MAP}: \underline{w}^* = \arg\max_{\underline{w}} [\log p(\underline{w} \mid D)] = \arg\max_{\underline{w}} \Big[\log \overbrace{p(D \mid \underline{w})}^{\text{likelihood}} - \overbrace{\frac{1}{2\sigma_w^2}}^{\text{"}\lambda\text{"}} \underline{w}^T \underline{w}\Big]$$

↳ basically regularized MLE

Laplace approx

1. find $\underline{w}^*$: $\underline{w}^* = \arg\min_{w} \overset{\text{energy}}{E(\underline{w})}$ where $E(\underline{w}) = -\log p(\underline{w}, D)$
2. calculate $H_{ij} = \frac{\partial^2 E(\underline{w})}{\partial w_i \partial w_j}\Big|_{w = w^*}$
3. set $p(\underline{w} \mid D) \approx N(\underline{w}; \underline{w}^*, H^{-1})$



sometimes not so great

variation inference:

$$D_{KL}(p \| q) = \int p(z) \log \frac{p(z)}{q(z)} dz$$

usually want $D_{KL}(q \| p)$ where $q$ approximates $p$

$$D_{KL}(q(\underline{w}; \alpha) \| p(\underline{w} \mid D)) = \int q(\underline{w}; \alpha) \log \frac{q(\underline{w}; \alpha)}{p(\underline{w} \mid D)} d\underline{w} \rightarrow = \frac{p(D \mid w)\, p(w)}{p(D)}$$

$$= \underbrace{-\int q(\underline{w}; \alpha) \log p(\underline{w} \mid D) d\underline{w}}_{\text{cross-entropy ... small = good}} + \underbrace{\int q(\underline{w}; \alpha) \log q(\underline{w}; \alpha) d\underline{w}}_{\text{(neg) entropy: want entropy to be large (spread out)}}$$

$$= \underbrace{\mathbb{E}_q[\log p(D \mid \underline{w})] + \cdots}_{J(q)} + \log p(D)$$

only the neg log likelihood term can't be computed in closed form

$$-\mathbb{E}_{N(\underline{w}; \underline{m}, V)}[\log p(D \mid w)]$$

So use MC

Gaussian mixture models:

$$p(\underline{x}^{(n)} \mid \theta) = \sum_k p(\underline{x}^{(n)}, z^{(n)} = k \mid \theta)$$
$$= \sum_k p(x^{(n)} \mid z^{(n)} = k, \theta) \overbrace{p(z^{(n)} = k \mid \theta)}^{\pi_k}$$
$$= \sum_k \pi_k N(\underline{x}^{(n)}; \mu^{(k)}, \Sigma^{(k)})$$

use EM to take NLL of $p(D \mid \theta)$

E: set soft responsibilities : $r_k^{(n)} = p(z^{(n)} = k \mid x^{(n)}, \theta) = \frac{\pi_k N(\cdot)_k}{\sum_l \pi_l N(\cdot)_l}$

M: update params $\theta = \{\pi, \{\underline{\mu}^{(k)}, \Sigma^{(k)}\}$

$$\mu^{(k)} = \frac{1}{r_k} \sum_{n=1}^{N} r_k^{(n)} x^{(n)}$$

$$\Sigma^{(k)} = \cdots$$