Welcome!

We'll start at ~1:33PM

Poll!

detailed feedback form after class

SPR Day 18
2020-04-01

start early!

Agenda Today

- Q&A about CP3
- Q&A about day 18 content
- A few short exercises

Post on Piazza!

submit code to autograder

Coming Monday: Midterm exam recap

post solutions

Upcoming Due Dates

- HW3 due tonight (Apr 1, 11:59pm)
- CP3 due Sunday (Apr 5, 11:59pm)
- Quiz3 released Tue Apr 7, due Wed Apr 8 11:59pm ET
---- Will be entirely online via gradescope
---- Time limit: 20 minutes
---- Multiple choice / short answers
- HW4 and CP4 released early next week

Tips for using autograd

CP3

write the code Numpy

automatic differentiation

When writing functions that need to be autograd-able

```
def f(x):
    return ag_np.sum(...)
```

- do not use "a[k] = ..." (assignment to an element of an array)

g = autograd.grad(f)

- do not use "ag_np.asarray” (allocating new memory)

$g(x) \triangleq \frac{\partial}{\partial x} f$

- do not assign to attributes “~~self.mu_KD = ...~~”

```
class GMM
    def fit(---)
        def calc_loss(---):
            ~~self.mu~~
            ~~self.stddev~~
        calc_grad = autograd.grad(calc_loss)
```

use local variables

Useful code example

(to avoid assignment a[k] = ...)

```
def calc_my_func(mu_KD, sigma_KD):
    list_of_arr = []
    for k in range(K):
        # compute array with shape (N,)
        arr_k_N = f(mu[k], sigma[k])
        list_of_arr.append(arr_k_N)
    arr_KN = ag_np.vstack(list_of_arr)
    return ag_np.sum(arr_KN)
```

(KN)

autograd

equiv:

```
arr_KN = zeros(K,N)
for k in range(K)
    arr_KN[k] = f(...)
```

easier — tricky chainrule

def fit(self, x_ND, ...):

$\boxed{\begin{array}{l} a = 1 \\ b = 3 \end{array}}$

def calc_loss(vec):

can access a, b, x_ND

local functions "inherit" any variables

try it!

Q&A: CP3

$\boxed{\text{calc\_EM\_loss}}$

calc_loss (from LBFGS)

$$\mathcal{L}(x, r, \pi, \mu, \tau)$$

$$\boxed{-\mathbb{E}_q\left[\log \frac{p(x,z)}{q(z)}\right]} + \text{penalty}(\tau)$$

$$-\log p(x \mid \pi, \mu, \tau) + \text{penalty}(\tau)$$

for $K=1$ EM should converge fast

E $\quad r^* \leftarrow \arg\max_r \mathcal{L}$

$$\mathcal{L}(x, r^*, \pi, \mu, \tau) = \log p(x \mid \pi, \mu, \tau)$$

assertions check this

M $\quad \pi, \mu, \tau \leftarrow \mathcal{L}(x, r, \pi, \mu, \tau)$

EMEMEMEM

loss_m $\leq$ loss_e $+ 10^{-9}$

Q&A: CP3

Marginal likelihood

$$\log p(x|\pi,\mu,\sigma) = \sum_n \sum_k \pi_k \text{ logNormPDF}(x_n | \mu_k, \sigma_k^2)$$

total = 0

for n in range(N):

log_p_K = log_pi_K + $\boxed{\text{log\_lik\_K}}$(x[n], ...)

logNormPDF

(K)

total += logsumexp(log_p_K)

for n in N

for k in K

def calc_neg_log_lik(   ): $\pi_k \, \mathrm{Norm}(x_n \mid \mu_k, \Sigma_k)$

list

for k in range(K):

log_p_N =

n N

list.append($1 \times N$)

$K \times N$

train_neg_log_lik

valid_neg_log_lik

negative loss is fine!

smaller => better

$-\log p(x \mid \text{——})$

PDF of reals   PDF ≥ 0

$-\infty \leq \log \mathrm{PDF} \leq +\infty$

Q&A: Day 18 EM for HMMs

"Emission"

Want: Estimate $\underset{\text{initial state prob}}{\pi}, \underset{\text{transition}}{A}, \overbrace{\mu, \sigma}^{\text{likelihood data given state}}$ given $\{x_t\}_{t=1}^{T}$ $\quad x_1 \ldots x_T$

Max Lik: $\max_{\pi, A, \mu, \sigma} \log p(x_{1:T} \mid \pi, A, \mu, \sigma)$ $\quad$ Sum rule requires K^T

EM: $\mathcal{L}(x, s, \pi, A, \mu, \sigma) \leq \log p(x_{1:T} \mid \pi, A, \mu, \sigma)$

E $\quad \boxed{s \leftarrow \arg\max_s \mathcal{L}}$ $\quad$ FORWARD α $\quad s \leftarrow f(\alpha, \beta)$

BACKWARD β $\quad r \leftarrow g(\alpha, \beta)$

M $\quad \pi, A, \mu, \sigma \leftarrow \arg\max \mathcal{L}(x, s, \ldots)$ $\quad \pi \leftarrow$ $\quad A \leftarrow$ $\quad \mu \leftarrow$ $\quad \sigma \leftarrow$

Q&A: Day 18 EM for HMMs

$$\mathcal{L}(x, s, \dots) = \mathbb{E}_q\left[\log \frac{p(x,z)}{q(z)}\right]$$

$$= \mathbb{E}_q\left[\log p(x,z) \overbrace{\boxed{-\log q(z|s)}}^{\text{needed}}\right]$$

π, A, μ, σ

That's all!

$$p(x) = \boxed{\sum_k \pi_k \, \mathrm{NormPDF}(x \mid \mu_k, \Sigma_k)}$$

$y \sim \mathrm{Norm}(\mu_k, \Sigma_k)$

$\mathrm{Cov}[y] = \Sigma_k$

$\mathbb{E}[yy^T] = \Sigma_k + \mu_k \mu_k^T$

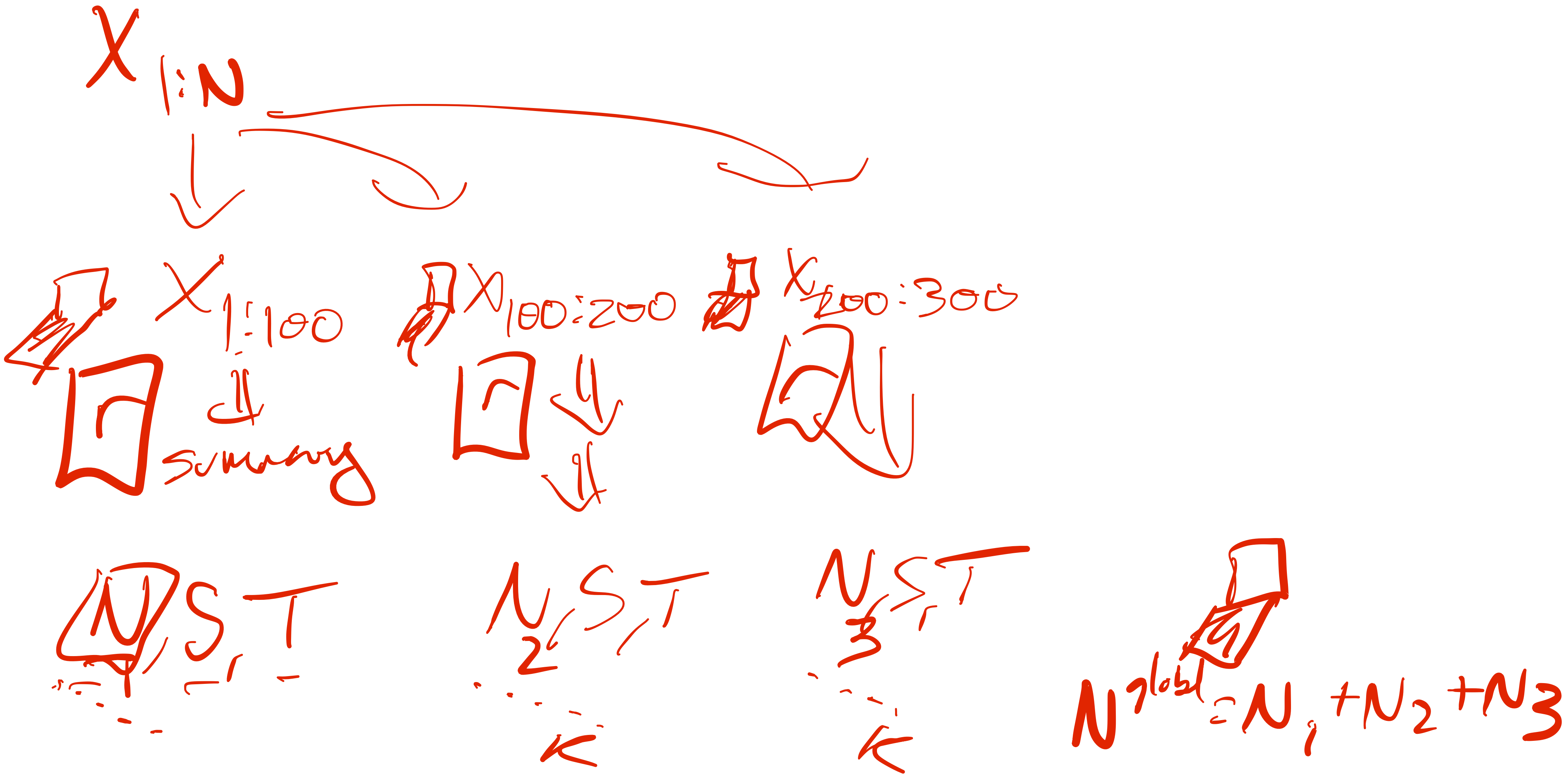
$$\mathrm{Cov}(x) = \underbrace{\mathbb{E}[xx^T]}_{\text{do this!}} - \mathbb{E}[x]\underbrace{\mathbb{E}[x]^T}_{\text{part a}}$$

$$\mathbb{E}[xx^T] = \int p(x)\, xx^T dx$$

$$= \int \sum_k \pi_k \, \mathrm{NormPDF}(x \mid \mu_k, \Sigma_k) \boxed{xx^T} dx$$

HW3 question from Andrew

Q&A: Day 18 EM for HMMs



9.25 $= f(\mu_k, N_k, S_k, T_k)$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

Assume x_n and μ_k scalars

$$\sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)$$

$$\sum_n \gamma_{nk} (x_n^2 - 2 x_n \mu_k + \mu_k^2)$$

$$\left[\sum_n \gamma_{nk} x_n^2\right] - \left[2 \sum_n \gamma_{nk} x_n \mu_k\right] + \left(\sum_n \gamma_{nk} \mu_k^2\right)$$

$$(\mu_k^2) \underbrace{\sum_n \gamma_{nk}}_{N_k(\gamma)}$$

$$\sum_n \sum_k \gamma_{nk} \underline{\log \text{NormPDF}}(x_n \mid \mu_k, \Sigma_k)$$

$$\left[\log 2\pi^{\frac{D}{2}} + \log|\Sigma_k| + -\frac{1}{2}\underbrace{(x_n - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}_{v^T \Sigma v}\right]$$

Exercise 1: Write out an expression for the complete HMM log likelihood (assume you have observed per-tstep state assignments z and per-tstep data x)

Exercise 2: Explain why forward algorithm is an instance of dynamic programming