ISTA 421 / INFO 521 - Final Written Assignment

Due: Monday, December 12, 5pm

15 pts total undergraduate / 20 pts total graduate

The total points here comprise 5% out of the 15% of the Final Assignment

Shuo Yang

Graduate

Instructions

You must work on the problems **INDEPENDENTLY**, not in groups.

The work on each problem must be your own.

Include in your final submission the pdf of written answers along with separate files for any python scripts that you write in support of answering the questions (although no scripts are needed for this assignment; if you do write scripts, clearly note in your pdf written answers which script filenames were used). You are required to create either a .zip or tarball (.tar.gz / .tgz) archive of all of the files for your submission and submit the archive to the D2L dropbox by the date/time deadline above.

NOTE: Problem 3 is **required for graduate students only**; undergraduates may complete them for extra credit equal to the point value.

(FCMA refers to the course text: Rogers and Girolami (2012), *A First Course in Machine Learning.*)

1. [4 points] Adapted from **Exercise 5.5** of FCMA p.204:

   For a Bayesian classifier with multinomial class-conditionals with M-dimensional parameters $\mathbf{q}_c$, compute the posterior Dirichlet for class c when the prior over $\mathbf{q}_c$ is a Dirichlet with constant parameter α and the observations belonging to class c are the N_c observations $\mathbf{x}_1, ..., \mathbf{x}_{N_c}$.

   Solution.
   We can compute the posterior Dirichlet by muliplying prior Dirichlet with likelihood.

$$\begin{aligned} p(\mathbf{q_c}|\mathbf{x_n}) &\propto p(\mathbf{x_n}|\mathbf{q_c})p(\mathbf{q_c}) \\ &\propto \prod_{j=1}^{M} q_{c_j}^{x_{n_j}} \prod_{j=1}^{M} q_{c_j}^{\alpha_j - 1} \\ &\propto \prod_{j=1}^{M} q_{c_j}^{x_{n_j}+\alpha_j-1} \end{aligned}$$

2. [4 points] Adapted from **Exercise 6.1** of FCMA p.234:

   Derive the EM update for the variance of the dth dimension and the kth component, σ_{kd}^2, when the cluster components have a diagonal Gaussian Likelihood:

$$p(\mathbf{x}_n|z_{nk} = 1, \mu_{k1}, ..., \mu_{KD}, \sigma_{k1}^2, ..., \sigma_{kD}^2) = \prod_{d=1}^{D} \mathcal{N}(\mu_{kd}, \sigma_{kd}^2)$$

   Solution.
   Let $\mathcal{L} = p(\mathbf{x}_n|z_{nk} = 1, \mu_{k1}, ..., \mu_{KD}, \sigma_{k1}^2, ..., \sigma_{kD}^2)$ and rewrite the diagonal Gaussian Likelihood in logarithmic form:

$$\begin{aligned} \log \mathcal{L} &= \sum_{d=1}^{D} \log \mathcal{N}(\mu_{kd}, \sigma_{kd}^2) \\ &= \sum_{d=1}^{D} \log(\frac{1}{\sqrt{2\sigma_{kd}^2 \pi}} e^{-\frac{(\mathbf{x_n}-\mu_{kd})^2}{2\sigma_{kd}^2}}) \end{aligned}$$

   Take the partial derivative of σ_{kd}^2 and set it to zero to find out the EM update of σ_{kd}^2:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \sigma_{kd}^2} &= \frac{\partial(-\frac{1}{2}\log(2\sigma_{kd}^2\pi) - \frac{(\mathbf{x_n}-\mu_{kd})^2}{2\sigma_{kd}^2})}{\partial \sigma_{kd}^2} \\ &= -\pi \frac{1}{2\sigma_{kd}^2 \pi} + \frac{(\mathbf{x_n} - \mu_{kd})^2}{2} \frac{1}{\sigma_{kd}^4} \end{aligned}$$

   Set the above partial derivative to 0 and we get the EM update of σ_{kd}^2:

$$\hat{\sigma}_{kd}^2 = (\mathbf{x_n} - \mu_{kd})^2$$

3. [5 points; **Required only for Graduates**] Adapted from **Exercise 6.6** of FCMA p.235:

   Derive an EM algorithm for fitting a mixture of Poisson distributions. Assume you observe N integer counts, $x_1, ..., x_N$. The likelihood is:

$$p(\mathbf{X}|\mathbf{\Lambda}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} \exp\{-\lambda_k\}}{x_n!}$$

Solution. <Solution goes here>

4. [3 points]

   Consider the 2-bit XOR problem for which the entire instance space is as follows: In each row, x_1

t	x_1	x_2
-1	-1	-1
$+1$	-1	1
$+1$	1	-1
-1	1	1

   and x_2 are the coordinates and t is the class for the point. These instances are not linearly separable, but they are separable with a polynomial kernel. Recall that the polynomial kernel is of the form $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \top \mathbf{x}_j + c)^d$ where c and d are integers. Select (by hand!) values for c and d that yield a space in which the instances above are linearly separable. Write down the mapping Φ to which this kernel corresponds, write down $\Phi(x)$ for each instance above, and write down the parameters of a hyperplane in the expanded space that perfectly classifies the instances (there are a range of possible hyperplanes, just pick one set of hyperplane parameters that satisfies separating the points – you are not maximizing the margin here, just coming up with one possible separating hyperplane). Again, this can be done without writing code or deriving an analytic solution!

   Solution.
   We choose $c = 1$ and $d = 2$. Thus the polynomial kernel is:

   $$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \top \mathbf{x}_j + 1)^2$$

   where $\mathbf{x}_i = [x_{i1}, x_{i2}]\top$ and $\mathbf{x}_j = [x_{j1}, x_{j2}]\top$. Therefore, we have:

   $$\begin{aligned}\kappa(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i \top \mathbf{x}_j + 1)^2 \\ &= ((x_{i1}x_{j1} + x_{i2}x_{j2}) + 1)^2 \\ &= 1 + x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{j1}x_{i2}x_{j2} \\ &= \Phi(\mathbf{x}_i)\mathbf{x}_j \top\end{aligned}$$

   Thus the mapping function for this 2nd degree polynomial kernel is:

   $$\Phi(\mathbf{x}) = (1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2)$$

   The mapping for each instance above:

   $$\begin{aligned}\Phi((-\mathbf{1}, -\mathbf{1})) &= (1, 1, \sqrt{2}, 1, -\sqrt{2}, -\sqrt{2}) \\ \Phi((-\mathbf{1}, +\mathbf{1})) &= (1, 1, -\sqrt{2}, 1, -\sqrt{2}, \sqrt{2}) \\ \Phi((+\mathbf{1}, -\mathbf{1})) &= (1, 1, -\sqrt{2}, 1, \sqrt{2}, -\sqrt{2}) \\ \Phi((+\mathbf{1}, +\mathbf{1})) &= (1, 1, \sqrt{2}, 1, \sqrt{2}, \sqrt{2})\end{aligned}$$

   One set of hyperplane parameters that can linearly separate the points is $x_1x_2 = 0$.

5. [4 points] Adapted from **Exercise 4.4** of FCMA p.166:

   Given the expression for the area of a circle, $A = \pi r^2$, and using only uniformly distributed random variates, devise a sampling approach for computing π. Describe your method in detail and provide you script to do the estimation – this script should be called `pi_sample_estimate.py`.

Solution.
As we can see in the Figure 1, a circle with radius $r = 1$ and area $= \pi 1^2 = \pi$ is inscribe inside a rectangle whose area $= 2 * 2 = 4$. If we sample points uniformly within the rectangle and the number of samples N is large enough, then we would have approximately $M = N * \frac{\pi}{4}$ points fall into the circle. Thus we can calculate π as $\pi \approx 4 * \frac{M}{N}$. We know that a point (x, y) is within the circle if $x^2 + y^2 \leq r^2$.

The code is implemented in `pi_sample_estimate.py` and is pasted below:

Code Listing 1: `pi_sample_estimate.py`

```
import numpy as np

num_samples = 100000000
R = 1
xs = np.random.uniform(-1.0, 1.0, num_samples)
ys = np.random.uniform(-1.0, 1.0, num_samples)

distances2center = xs**2 + ys**2
points_fall_in_circle = distances2center[distances2center ¡= R**2]

print 'Number of points (N) sampled within the rectangle ([−1,−1],
[1,1]): {}'.format(num_samples)
print 'Number of points (M) fall into the circle inscribed inside
the rectangle: {}'.format(points_fall_in_circle.shape[0])
print 'Estimated PI = 4 ∗ M/N = {}'.format(4 *
float(points_fall_in_circle.shape[0]) / num_samples)
```

The output is shown below:

```
Number of points (N) sampled within the rectangle ([-1,-1], [1,1]): 100000000
Number of points (M) fall into the circle inscribed inside the rectangle: 78530364
Estimated PI = M/N = 3.14121456
```

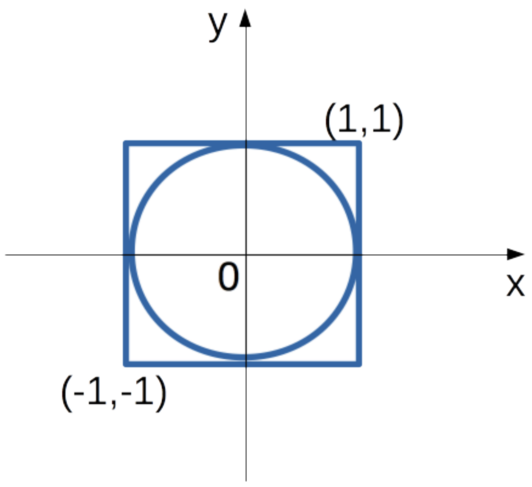

Figure 1: PI sampling illustration