IAML 2017 RESIT

1. a. Dimensionality reduction using PCA
   - start with correlated data
   - transform data to zero mean and unit variance
   - compute the covariance matrix
   - compute eigenvectors and eigenvalues of the cov. mat
   - pick the eigenvectors with the largest eigenvalues.
     e.g. sort λ such that $\lambda_1 > \lambda_2 \ldots > \lambda_n$
     pick the eigenvectors w/ eigenvalues that represent 90% or 95% of the cumulative variance.
   - project data onto the eigenvectors which are the PCs.
   - leaves us with uncorrelated data.

* zero mean, variance

$$\Sigma = \sum (x_i - \bar{x}) \cdot (y_i - \bar{y})$$

for two dimensions

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{x,y}^2 \\ \sigma_{x,y}^2 & \sigma_y^2 \end{bmatrix}$$

* find eigenvalues

$$\det(\Sigma - \lambda I) = 0$$

$$\sigma_x^2 = \frac{1}{n_x} \sum_i (x_i - \bar{x})^2$$

* eigenvectors have to be unit vectors i.e. $\|e_i\| = 1$

$$\sigma_y^2 = \frac{1}{n_y} \sum_i (y_i - \bar{y})^2$$

To get $\underline{z}$ from $\underline{x}$, $z = e^T x$

$$\sigma_{x,y}^2 = \frac{1}{N} \sum_i^n (x_i - \bar{x})^2 (y_i - \bar{y})^2$$

* $PCA_1 \rightarrow$ maximum variance
  $PCA_2 \rightarrow$ perpendicular to PCA_1

b. kNN Regression Algorithm

Project $\underline{x}$ onto the n-many eigenvectors to create $\underline{x}'$. then use the reduced dimension dataset to find the k-nearest training points by using a distance measure (Euclidean). Get the average of their label values to find the predicted score of $\underline{x}$.

Optimal k → divide the training set into training and validation and run multiple values of k on the validation set and calculate the error for each. Pick k that performs the best on the validation set.

c. Linear regression on the PCA-reduced $\underline{x}$, called $\underline{z}$.

$$f(\underline{z}) = \underbrace{\underline{\tilde{w}}^T \underline{z} + w_0}_{\text{parameters}}$$

Give a suitable objective function that could be used to determine $\underline{w}$ from training set.

Explain why it is probably a good idea to use PCA-reduced input vector $\underline{z}$ as input to LR model rather than raw input $\underline{x}$.

$$O(\underline{w}) = \sum_i (y_i - f(\underline{z}_i))^2$$

$$= (\emptyset^T \emptyset)^{-1} \underbrace{\emptyset}_{\text{design matrix (bias + training}}{}^T y$$

PCA with LR

- high dimensions is sparse
  - not a good representation of the data
- $D >> n$
  - solution is much more dense after reduced
- curse of dimensionality
  - as the number of feature/dimensions grows, the amount of data we need to generalise accurately grows exponentially.

d. Relative merits of KNN and LR

KNN - no training phase
- makes few assumptions about the data
- new training datapoints can be added easily

LR - easy to interpret
- less computationally costly
- faster test phase (easy to fit)

2. Naive Bayes

a. Define NB, use correct equations. Why is it Naive?

Naive Bayes is a generative model which creates a complete probability distribution model for each class.

It defines the probability of an instance $\underline{x}$ belonging to class y as

$$P(c|x) = \frac{P(x|c)P(c)}{\sum_i (x|C_k)P(C_k)}$$

If the input vector has multiple attributes, it is hard to compute the likelihood. Therefore, we assume conditional independence.

i.e. each data item is assumed to be from the same data model and is independent w/ each other.

Relevant probability distribution
discrete → calculate/count by hand and then calculate likelihood.
real-valued → create a Gaussian for each att. x_i and multiply all of them to find $P(x|k)$.

$$P(x|k) = P(x_1, \ldots, x_n / k) \prod_{i=1}^{n} P(x_i|k)$$

$$P(k|x) \propto P(x|k) \cdot P(k)$$

b. 'happiness' survey

$\underline{x}$ = (rich, married, healthy)

content → (1, 1, 1), (0, 0, 1), (1, 1, 0), (1, 0, 1) $4/9$

not content → (0, 0, 0), (1, 0, 0), (0, 0, 1), (0, 1, 0), (0, 0, 0) $5/9$

P(content | (0, 1, 1)) = ?

	0 rich	1 married	1 healthy
content	3/4	2/4	3/4
not content	1/5	1/5	1/5

$P((0,1,1) \mid \text{content}) = 1/4 \times 2/4 \times 3/4 = 6/64$

$P((0,1,1) \mid \text{not content}) = 4/5 \times 1/5 \times 1/5 = 4/125$

$$P(\text{content} \mid (0,1,1)) = \frac{6/64 \times 4/9}{6/64 \times 4/9 + 4/125 \times 5/9} = 0.7$$

What is the probability that a person who is 'not rich' and 'married' is 'content'?

$P((0,1) \mid \text{content}) = 1/4 \times 2/4 = 2/16$

$P((0,1) \mid \text{not content}) = 4/5 \times 1/5 = 4/25$

$$P(\text{content} \mid (0,1)) = \frac{2/16 \times 4/9}{2/16 \times 4/9 + 4/25 \times 5/9} = 0.38$$

c. Now we have the age of the respondent. How can this attribute can be added to NB classifier.

Age → real-valued, modeled by Gaussian.

How to estimate the relevant parameters?

- calculate mean and variance
- calculate $P(x_i \mid k)$, x_i = age
- multiply with the rest of the values

d. Ordinal vs. categorical

↳ some sort of hierarchy exist in the set.

e.g. low, medium, high

categorical - no relationship

- blue, yellow red

1-of-m encoding for categorical data. Why bad for ordinal?

- convert categories to binary attributes

instead of

object = red + green + blue

= (1, 1, 1)

↑

indicates prese

- does not preserve the order of attributes

e.g. distance of high → low >

medium → low is not preserved