# CS 434 – Assignment 1

## 1 Statistical Estimation

1. **Maximum Likelihood Estimation of $\lambda$ [8pts]** Assume we observe a dataset of occurrence counts $D = \{x_1, x_2, ..., x_N\}$ coming from $N$ i.i.d random variables distributed according to Pois$(X = x; \lambda)$. Derive the maximum likelihood estimate of the rate parameter $\lambda$. To help guide you, consider the following steps:

   (a) Write out the log-likelihood function $logP(D|\lambda)$

$$ln\left(\prod_{i=1}^{N} P(x_i|\lambda)\right) = ln\left(\prod_{i=1}^{N} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}\right) = \sum_{i=1}^{N}\left(ln\left(\frac{\lambda^{x_i} e^{-\lambda}}{x_i!}\right)\right)$$

$$= \sum_{i=1}^{N}\left(ln(\lambda^{x_i} e^{-\lambda}) - ln(x_i!)\right) = \sum_{i=1}^{N}\left(x_i ln(\lambda) - \lambda ln(e) - ln(x_i!)\right)$$

$$= \left(\sum_{i=1}^{N} x_i\right) ln(\lambda) - N\lambda - \sum_{i=1}^{N} ln(x_i!)$$

   (b) Take the derivative of the log-likelihood function with respect to $\lambda$

$$\frac{d}{d\lambda}\left(\sum_{i=1}^{N} x_i\right) ln(\lambda) - \lambda - \sum_{i=1}^{N} ln(x_i!)$$

$$= \frac{\left(\sum_{i=1}^{N} x_i\right)}{\lambda} - N$$

   (c) Set the derivative equal to zero and solve for $\lambda$ – call this maximizing value $\hat{\lambda}_{MLE}$

$$0 = \frac{\left(\sum_{i=1}^{N} x_i\right)}{\hat{\lambda}_{MLE}} - N$$

$$N = \frac{\left(\sum_{i=1}^{N} x_i\right)}{\hat{\lambda}_{MLE}}$$

$$\hat{\lambda}_{MLE} = \frac{\left(\sum_{i=1}^{N} x_i\right)}{N}$$

2. **Maximum A Posteriori Estimate of $\lambda$ with a Gamma Prior** [**8pts**] As before, assume we observe a dataset of occurrence counts $D = \{x_1, x_2, ..., x_N\}$ coming from $N$ i.i.d random variables distributed according to Pois$(X = x; \lambda)$. Further, assume that $\lambda$ is distributed according to a Gamma$(\Lambda = \lambda; \alpha, \beta)$. Derive the MAP estimate of $\lambda$. To help guide you, consider the following steps:

   (a) Write out the log-posterior $logP(\lambda|D) \propto logP(D|\lambda) + logP(\lambda)$

   $$lnP(D|\lambda) + lnP(\lambda) = lnP(D|\lambda) + ln\left(\frac{\beta^{\alpha}\lambda^{\alpha-1}e^{-\beta\lambda}}{\Gamma(\alpha)}\right)$$

   $$= lnP(D|\lambda) + ln(\beta^{\alpha}\lambda^{\alpha-1}e^{-\beta\lambda}) - ln(\Gamma(\alpha))$$

   $$= lnP(D|\lambda) + \alpha ln(\beta) + (\alpha - 1)ln(\lambda) + (-\beta\lambda) - ln(\Gamma(\alpha))$$

   (b) Take the derivative of $logP(D|\lambda) + logP(\lambda)$ with respect to $\lambda$

   $$\frac{d}{d\lambda}(lnP(D|\lambda)) + \frac{d}{d\lambda}(\alpha ln(\beta) + (\alpha - 1)ln(\lambda) + (-\beta\lambda) - ln(\Gamma(\alpha)))$$
   $$= \frac{\left(\sum_{i=1}^{N} x_i\right)}{\lambda} - N + \frac{\alpha - 1}{\lambda} - \beta = \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{\lambda} - \beta - N$$

   (c) Set the derivative equal to zero and solve for $\lambda$ – Call this the maximizing value $\hat{\lambda}_{MAP}$

   $$0 = \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{\lambda} - \beta - N$$
   $$\beta + N = \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{\lambda}$$
   $$\hat{\lambda}_{MAP} = \frac{\sum_{i=1}^{N} x_i + \alpha - 1}{\beta + N}$$

3. **Deriving the Posterior of a Poisson-Gamma Model** [**4pt**]. Show that the Gamma distribution is a conjugate prior to the Poisson by deriving expressions for parameters $\alpha P$ , $\beta P$ of a Gamma distribution such that $P(\lambda|D) \propto$ Gamma$(\lambda; \alpha P, \beta P)$.

   [Hint: Consider $P(D|\lambda)P(\lambda)$ and group like-terms/exponents. Try to massage the equation to looking like the numerator of a Gamma distribution. The denominator can be mostly ignored if it is constant with respect to $\lambda$ as we are only trying to show a proportionality ($\propto$).]

   $$P(D|\lambda)P(\lambda) = \prod_{i=1}^{N}\left(\frac{\lambda^{x_i}e^{-\lambda}}{x_i!}\right)\left(\frac{\beta^{\alpha}\lambda^{\alpha-1}e^{-\beta\lambda}}{\Gamma(\alpha)}\right)$$

$$= \frac{(\lambda^{\sum x_i} e^{-N\lambda})(\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda})}{\prod(x!)\Gamma(\alpha)} = \frac{\beta^\alpha \lambda^{\sum x_i + \alpha - 1} e^{-\beta\lambda - N\lambda}}{\prod(x_i!)\Gamma(\alpha)}$$

$$= \frac{\beta^\alpha \lambda^{(\sum(x_i)+\alpha)-1} e^{-\lambda(\beta+N)}}{\prod(x_i!)\Gamma(\alpha)}$$

$$\alpha P = \sum_{i=1}^{N}(x_i) + \alpha$$

$$\beta P = \beta + N$$

# 2 k-Nearest Neighbor (kNN)

4. **Encodings and Distance [3pt]**

$$\text{Private } \rightarrow [1, 0, 0]$$
$$\text{State Gov } \rightarrow [0, 1, 0]$$
$$\text{Never Worked } \rightarrow [0, 0, 1]$$

$$\text{Distance from Private to Never Worked } \rightarrow \sqrt{(1-0)^2 + 0 + (1-0)^2} = \sqrt{2}$$

One can clearly see that this will apply to any other combination of the three.

$$\text{Private } = 1$$
$$\text{State Gov } = 2$$
$$\text{Never Worked } = 3$$

$$\text{Distance from Private to Never Worked } \rightarrow \sqrt{(1-3)^2} = \sqrt{2}$$
$$\text{Distance from Private to State Gov } \rightarrow \sqrt{(1-2)^2} = 1$$

Based on these results, categorical encoding gives equal weight to each whereas ordinal makes some catergories further than others.

5. **Looking at Data [5pt]** What percent of the training data has an income >50k? Explain how this might affect your model and how you interpret the results. For instance, would you say a model that achieved 70% accuracy is a good or poor model? How many dimensions does each data point have (ignoring the id attribute and class label)? [Hint: check the data, one-hot encodings increased dimensionality]

About 25% of our training data has an income >50k. This will cause the model to have a bias towards predicting an income <50k. This should be okay as long as it is proportional to the total population. A model which has achieved a 70% accuracy is a poor model since this is likely achievable by guessing. There are 84 dimensions in each data point.

6. **Norms and Distances [3pt]** Distances and vector norms are closely related concepts. For instance, an $L_2$ norm of a vector $x$ (defined below) can be interpreted as the Euclidean distance between $x$ and the zero vector:

$$||x||_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$$

Given a new vector $z$, show that the Euclidean distance between $x$ and $z$ can be written as an $L_2$ norm

$$||x - z||_2 = \sqrt{\sum_{i=1}^{d} (x_i - z_i)^2}$$

9. **Hyperparameter Search [15pt]** What is the best number of neighbors ($k$) you observe? When $k = 1$, is training error 0%? Why or why not? What trends (train and cross-validation accuracy rate) do you observe with increasing $k$? How do they relate to underfitting and overfitting?

I found that $k = 99$ was the best $k$ value for me. When $k = 1$, training error was not 0%, this is because I remove the value I'm querying, so it doesn't just find itself. As $k$ increases, I saw that the accuracy got better then drop off. This happened as the model transitioned to underfit, to better, to overfit.