Deep Bayes Assignments 1 & 2

Rostov, Maxim - 11808470 (University of Amsterdam)
maxim.rostan@gmail.com

April 9, 2019

Assignment 1

Problem 1

The random variable ξ has Poisson distribution with the parameter λ. If $\xi = k$ we perform k Bernoulli trials with the probability of success p. Let us define the random variable η as the number of successful outcomes of Bernoulli trials. Prove that η has Poisson distribution with the parameter $p\lambda$.

Can directly see from the description that there is Bernoulli with k trials and can switch to model this as Binomial. From elementary courses in probability theory, Poisson distribution can be derived from Binomial distribution (Conditions: let the number of trials go to infinity $k \to \infty$, the probability of success in a single trail tend to zero $p \to 0$, and fix the mean of the Binomial $\lambda = k * p$). Thus, It is known that both distributions are of the same conjugate family (Note: Binomial distribution is in the exponential family only if we fix the number of trials, k).

Start with representing our rv variable η.

$$P(\eta = m) = \frac{k!}{m!(k-m)!} p^m (1-p)^{k-m}$$

k is just a fixed values for ξ, consider all possible values for ξ and weight with their probability:

$$P(\eta = m, \lambda, p) = \sum_{k=m}^{\infty} \frac{k!}{(k-m)!m!} p^m (1-p)^{k-m} \frac{\lambda^k e^{-\lambda}}{k!}$$

This pmf can be further simplified by bring terms that do not depend on k outside of the sum.

$$P(\eta = m, \lambda, p) = \frac{p^m e^{-\lambda}}{m!} \sum_{k=m}^{\infty} \frac{\lambda^k (1-p)^{k-m}}{(k-m)!}$$

Next, in order to employ the rules of summation over infinite series make the substitution $n = k - m$:

$$P(\eta = m, \lambda, p) = \frac{p^m e^{-\lambda}}{m!} \sum_{n=0}^{\infty} \frac{\lambda^{n+m} (1-p)^n}{n!}$$

$$P(\eta = m, \lambda, p) = \frac{(\lambda p)^m e^{-\lambda}}{m!} \sum_{n=0}^{\infty} \frac{(\lambda(1-p))^n}{n!}$$

From calculus:

$$\exp x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Thus, we can marginalize all possible values for n:

$$P(\eta = m, \lambda, p) = \frac{(\lambda p)^m e^{-\lambda p}}{m!}$$

Which is a Poisson distribution parameterized by mean/variance equal to $\lambda * p$.

Problem 2

A strict reviewer needs t_1 minutes to check assigned application to Deep|Bayes summer school, where t_1 has normal distribution with parameters μ_1 = 30, σ_1 = 10. While a kind reviewer needs t_2 minutes to check an application, where t_2 has normal distribution with parameters μ_2 = 20, σ_2 = 5. For each application the reviewer is randomly selected with 0.5 probability. Given that the time of review t = 10, calculate the conditional probability that the application was checked by a kind reviewer.

Two Gausssian distributions which identify the likelihoods for each type of the reviewer. Prior is equal-likely uniforms.

Identify, priors:

$$P(strict) = P(kind) = 0.5$$

Likelihoods ($i \in [1, 2]$; c = class of the review):

$$P(t_i|\mu_i, \sigma_i, c = i) \sim \mathcal{N}(\mu_i,\, \sigma_i^2)$$

As the time of review is given as a single point in time $t = 10$. Calculate likelihood from PDF of the Gaussian.

$$P(t_1|\mu_1, \sigma_1, c = 1) = 0.005$$

$$P(t_2|\mu_2, \sigma_2, c = 2) = 0.01$$

Evidence (use sum rule):

$$P(t) = P(t|kind)P(kind) + P(t|strict)P(strict) = 0.0025 + 0.005$$

Use Bayes rule. Calculate the class-conditional probability by multiplying likelihood with prior and normalizing by evidence.

$$P(kind|t) = (0.01 * 0.5)/0.0075 = 2/3$$

Assignment 2

Paper choice

Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles.* NeurIPS 2017.

Questions

a. How do authors change the NN to make it capable to estimate uncertainty for regression tasks?

MSE can be thought of as encompassing $variance^2 + mean + irreducibleuncertainty$. We cannot directly estimate uncertainty of the predictive distribution using MSE. Thus, the authors train the NN's according to measure that utilizes 'calibration of predictive uncertainty' (where 'calibration' is measure by proper scoring rules) by minimizing loss $L(\theta) = S(p_\theta, q)$ (where S is a proper scoring function). Hence, minimize the negative log-likelihood.

a. What is the distribution on the outputs, as defined by the NN architecture and loss? What is the distribution on the outputs, as defined by the NN architecture and loss?

So, just simple MLP would use (conditional) Gaussians to model loss/error. Authors choose to treat the output variable as a Gaussian with varying variance among the samples. If we add on ensemble of NN's, naturally arrive to the mixture of (conditional) Gaussians.

b. What are adversarial examples? What is the purpose of using them to train the ensemble? Can an object with an unchanged prediction be an adversarial example?

Adversarial examples (AE) are inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset. Strictly speaking, AE are used to not allow a NN to overfit; in the paper I think the purpose is two fold: prevent overfitting and generate more valid samples for training an ensemble. No, AE should make a NN to predict wrong, thus if we add noise and do not change the prediction we are not earning anything.

c. Let's imagine that somebody collected a dataset with many out-of-domain images or images with wrong labels. How can the proposed uncertainty estimation method be applied to clean the dataset from such objects?

We train an ensemble on the 'correct' images. Then, given the new dataset with many out-of-domain images, for images far different then those which were used during training we will have disagreeing NN's. None of the NN's in the ensemble were trained on exactly the same data. Hence, if at the test time we encounter image very different from the training data an ensemble is likely to produce disagreeing results. Thus, higher entropy and we can identify images which we are highly uncertain about (based on some threshold), see Figure 3. These images have probably came out of the domain (or just difficult to classify).