1 Formulaic coincidences

We observe a few remarkable coincidences for certain formulas in machine learning

Boosting: The empirical data set is weighted

$$\mathcal{D} = \{(w_n, x_n) : n \in \{1, \dots, N\}\} \tag{1}$$

Friedman (2000) interpreted boosting as optimization of an exponential error function

$$E = \sum_{n=1}^{N} \exp\{-t_n f(x_n)\} \tag{2}$$

Log-sum-exp is a common optimization function in convex optimization. Interestingly, the convex conjugate of this function is the negative entropy over a discrete distribution. This is used in Jaynes' derivation of the maximum entropy distribution.

$$\begin{aligned} f(x) &= \log\left(\sum_{n=1}^{N} e^{x_n}\right) \\ f^*(y) &= \sum_{n=1}^{N} y_i \log y_i \end{aligned} \tag{3}$$

In fact, with weights it's the posynomial form of geometric programming (Boyd,):

$$\begin{aligned} f(x) = \sum_{n=1}^{N} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{dk}} \\ \min \quad f(x) = \sum_{n=1}^{N} e^{\langle a_{0n}, y\rangle + b_{0n}} \\ \text{subject to} \quad \langle a_{mn}, y\rangle + b_{mn}, m \in \{1, \dots, M\} \end{aligned} \tag{4}$$

Another amazing coincidence is that log-sum-exp is the log normalization constant/log partition function of a multinoulli distribution, with unnormalized probabilities

$$\begin{aligned} Z &= \sum_{n=1}^{N} e^{t_n \log \tilde{p}_n} \\ \log Z &= \log\left(\sum_{n=1}^{N} e^{\langle t_n, \eta_n\rangle}\right) \end{aligned} \tag{5}$$

In the derivation of free energy, we observe that the upper bound of the free energy for a given energy function is the log normalizer of the Boltzmann distribution of that energy function

$$\text{Free energy} = E_{x\sim p(x)}\left[E(x)\right] - TE_{x\sim p(x)}\left[\log p(x)\right] \tag{6}$$

We also observe that with a given energy function and expected energy is constant, optimizing the Helmholtz free energy is equivalent to optimizing entropy.

An exponential family can be generated from an intrinsic measure from a Laplace transform, which literally calculates a normalization factor.

$$\mathcal{L}h(y) = \int e^{\langle y,x\rangle} h(x)dx \tag{7}$$

In a uniform discrete case, this is the sum of exponents. In the weighted discrete case, this is a weighted sum of exponents.

$$\begin{aligned}\mathcal{L}\left(\frac{1}{N}\sum_{n=1}^{N}\delta\left(x = x_n\right)\right) &= \sum_{n=1}^{N}\frac{1}{N}e^{\langle y,x_n\rangle}\\ \mathcal{L}\left(\sum_{n=1}^{N}w_n\delta\left(x = x_n\right)\right) &= \sum_{n=1}^{N}w_n e^{\langle y,x_n\rangle}\end{aligned} \tag{8}$$

For sequential estimation, sequential models use bootstrapping of samples at every time step, to create a weighted empirical distribution approximating the posterior. This can be seen as equivalent to a mixture distribution.

$$\begin{aligned}p(z_{n+1}|X_n) &= \int p(z_n|X_n)p(z_{n+1}|z_n)dz_n\\ &\approx w_n^{(l)}p(z_{n+1}|z_n^{(l)})\end{aligned} \tag{9}$$

In Importance Weighted Auto-encoders, the weights are equivalent to sampling from the importance weighted posterior (Domke and Sheldon, 2019, Agakov and Barber, 2004), which is a marginalization over an auxiliary variable:

$$\log p(x) = E\left[\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{p(z_m,x)}{q(z_m)}\right)\right] + KL\left(q_{IWAE}(z_{1:M}\|p_{IWAE}(z_{1:M}\right)E_{s(\omega)}\left[R(\omega,x)a(z|\omega,x)\right] = p(z,x) \tag{10}$$

2

Using the same argument, we can add a factor potential to a PGM / add an energy function to an energy based model, and that would be equivalent to adding a weight:

$$
\begin{aligned}
\tilde{P}_{\Theta}(\mathcal{X}) &= \prod_i e^{\phi(C_i)} \\
\tilde{P}'_{\Theta}(\mathcal{X}) &= e^{\phi'(C_{i'})}\tilde{P}_{\Theta} \\
&= E_{c_{i'} \sim e^{\phi'(C_{i'})}}\left[\tilde{P}_{\Theta}|c_{i'}\right] \\
&\approx \sum_{m=1}^{M} w_m \tilde{P}_{\Theta}|c_{i'}
\end{aligned}
\tag{11}
$$

This can be seen as integrating or marginalizing over a (potentially infinite) ensemble of models.

Generalization error is studied in statistical learning theory, and it creates bounds due to finite sampling estimation. Interestingly enough, a derivation of Hoeffding's inequality, for a Bernoulli distribution for classification, involves calculating a continuous bound on the binomial distribution, and this likelihood has an exponentiated KL-divergence term:

$$
P_{X_1,\ldots,X_N \sim P_{data}(X)}\left(E_{x \sim P_{emp}(x)}\left[1(x) = q\right]\right) \approx (2\pi N q(1-q))^{-\frac{1}{2}} e^{-NKL(\mathrm{Binomial}(q)\|\mathrm{Binomial}(p))}
\tag{12}
$$