Magda's technical notes

This is the 2nd set of my notes on various ML topics. I started writing the 1st set when beginning my PhD and in retrospect keeping such a set of notes is a useful exercise. I've decided to begin a new set just because now that I finished the PhD it just feels I may want to give it a fresh new go. The 1st set has plenty of useful stuff and is still available from my Dropbox here.

The general purpose of the notes is to help me understand better the selected topics by re-explaining (*re-* because these have been explained elsewhere many times), and to have a reference and possibly reusable material for later.

This is a working document not meant to be polished. There may be typos and other editing errors. Technical errors mean that I didn't quite understand something which I unfortunately cannot rule out.

Last update: December 17, 2019, section 4 Some useful inequalities (or equalities) - in progress

Contents

1 Variational autoencoders 3

1.1 Maximum likelihood . . . 3

1.2 Latent variable models . . . 4

1.3 Decoder - reconstructions/generations . . . 4

1.4 Importance sampling . . . 5

1.5 Approximate posterior . . . 5

1.6 Encoder - inference . . . 6

1.6.1 Jensen's inequality . . . 6

1.6.2 Variational inference . . . 6

1.6.3 ELBO for optimisation . . . 7

1.7 Objective optimization . . . 7

1.7.1 The regularization term (KL divergence) . . . 8

1.7.2 The reconstruction term . . . 8

1.7.3 Final loss term . . . 9

1.8 Proofs . . . 9

2 Quick notes on logistic regression 12

2.1 Logistic sigmoid . . . 12

2.2 Cross-entropy loss . . . 13

2.3 Multi-class classification . . . 13

3 Evaluate data log likelihood by importance sampling 15

4 Some useful inequalities (or equalities) - in progress 17

4.1 Numerical inequalities 17

4.1.1 Triangle inequality 17

4.1.2 Holder's inequality 17

4.1.3 Minkowski's inequality 18

4.2 Optimization 18

4.2.1 Change max to min 18

4.2.2 Min of monotonic function 18

4.2.3 Min of exponential 19

4.2.4 Jensen's inequality 19

4.3 Basic probability 19

4.3.1 Union bound 19

4.4 Variable transformation 19

4.5 Concentration bounds 20

4.5.1 Markov's inequality 20

4.5.2 Chebyshev's inequality 21

4.5.3 Cramer-Chernoff method 22

4.6 Sub-Gaussian random variables 24

Index 26

1 Variational autoencoders

This is my take on variational autoencoders, mainly based on [1, 2, 3].

1.1 Maximum likelihood

We have got a data set $\mathcal{D}$ of n data points $x \in \mathcal{X}$ generated i.i.d. from some unknown probability distribution with pdf $p^*(x)$.

Note: *When talking about distributions here, the functions $p(), q()$ are the probability density (or mass) functions. We use only these two letters for densities of all random variables but it should not be assumed that these are the* same *pdfs. If x and y are two different random variables then $p(x)$ and $p(y)$ are not the same. We also stride away from the rigorous statistical notation and indicate both the random variable and its realization by lower-case letters. We hope that the true meaning will be clear from the context.*

What we want is to be able to generate data that are *close* to our or original data. We hence want to learn a $p(x)$ which will be close enough to the original unknown $p^*(x)$ so that sampling from $p(x)$ will result in samples that have high probability under $p^*(x)$.

Note: *This kind of assumes that we are fairly sure about the support $\mathcal{X}$ which I reckon should be something nice, without holes and such. Typically simply real vector.*

We will consider a family of distributions $p_\theta(x)$. The way to think about the parameters θ is not as of the standard distributional parameters (such as the mean and variance or the natural parameter of exponential family) but rather as parameters of the deterministic functions that specify these distributional parameters. For example, if the distributional family we consider were Gaussian we could write $p_\theta(x) = N(\mu = f(\theta), \sigma^2 = g(\theta))$.

A classical approach to learning the parameters of a distribution (of a fixed family) is via *maximizing the likelihood* of the parameters given the dataset

$$\widehat{\theta} = \arg\max_\theta p_\theta(\mathcal{D}) = \arg\max_\theta \prod_i^n p_\theta(x_i) \ , \tag{1.1}$$

where the decomposition into the product across the data points is possible due to the i.i.d. sampling assumption.

Often times it may be more convenient to maximize the log-likelihood instead

$$\log p_\theta(\mathcal{D}) = \sum_i^n \log p_\theta(x_i) \ . \tag{1.2}$$

Some people find it useful to think about maximizing the log-likelihood as about minimizing the Kullback-Leibler (KL) divergence between the true and the estimated distribution as follows:

First observe that

$$\frac{1}{n} \log p_\theta(\mathcal{D}) = \frac{1}{n} \sum_i^n \log p_\theta(x_i) \approx \mathrm{E}_{p^*(x)} \log p_\theta(x) \tag{1.3}$$

is a sample estimate of the expectation $\mathrm{E}_{p^*(x)} \log p_\theta(x)$ converging to it by the law of large numbers as $n \longrightarrow \infty$. This indeed is equivalent to minimising the KL divergence as can be seen from

$$D_{KL}(p^*(x) \| p_\theta(x)) = \mathrm{E}_{p^*(x)} \log \frac{p^*(x)}{p_\theta(x)} = \mathrm{E}_{p^*(x)} \log p^*(x) - \mathrm{E}_{p^*(x)} \log p_\theta(x) \ . \tag{1.4}$$

1.2 Latent variable models

For complex data distribution $p^*(x)$ fixing $p_\theta(x)$ to a chosen distribution family may be too much of a simplifying assumption.

Note: *People often say complex data. But in fact the data support $\mathcal{X}$ cannot be too complex, what has to be complex is the generative / sampling distribution?*

We therefore consider a latent variable model

$$p^*(x) = \int p^*(x,z)\,dz = \int p^*(z)\,p^*(x|z)\,dz \tag{1.5}$$

with latent random variables $z \in \mathcal{Z} \subseteq \mathbb{R}^k$. This is a mixture model which gives a better handle on approximating the possibly complex distribution $p^*(x)$.

1.3 Decoder - reconstructions/generations

There are two problems here: we do not observe z, and we do not know the joint and the marginal or conditional distributions. So what we do is take assumptions. First we take a *prior assumption* for the distribution $z \sim p(z)$ as $N(0,I)$.

Note: *The prior distribution should be something nice, simple, mathematically convenient. Standard normal was used in the original Kingma's work because of its convenience for the reparametrization trick. It has been later extend to multinomial with gumble-soft max trick - I don't have the references and don't know the math exactly - and probably to many more.*

Second, we take a an assumption for the distribution family of the conditional typically to be a Gaussian $p_\theta(x|z) = N(\mu_z = f(z,\theta), \sigma^2 I)$. Note here the fixed diagonal covariance matrix. What σ^2 shall be here is not quite clear to me but I believe it's usually simply fixed to 1.

Note: *Since the log-likelihood over the data set is just a sum of the log-likelihood for each data point x_i I will for simplicity drop the sum across the data speak about the likelihood for each data point x_i. This is fine to do because the derivative of a sum is the sum of the derivatives and therefore translates easily into the optimisation step.*

With these assumptions we could start maximizing the log-likelihood.

$$\log p_\theta(x_i) = \log \int p(z)\,p_\theta(x_i|z)\,dz = \log \int p_\theta(x_i,z)\,dz \ . \tag{1.6}$$

To achieve this, we could sample a large number of z's from $N(0,I)$ to approximate the expectation by a sample average

$$\log p_\theta(x_i) \approx \log \frac{1}{m} \sum_j^m p_\theta(x_i|z_j) \ . \tag{1.7}$$

The conditional distributions are $p_\theta(x_i|z_j) = N(\mu = f(z_j,\theta), \sigma^2 I)$, where $f(z_j,\theta)$ is a function approximator such as neural network parametrized by θ with z_j as the set of inputs. This is the reconstruction network, the ***decoder*** of the VAE. With the x's and z's now given, the form of the log-likelihood and the function form f fixed, we could in principle maximize the log-likelihood with respect to θ and find the solution by some form of gradient updates.

The problem of the above approach is that the number of samples of z we need to approximate the expectation in equation (1.6) by the sample average in (1.7) may be huge, especially for higher dimensional z, so not doable in practice.

1.4 Importance sampling

The problem of needing a huge z sample is also because by sampling from the prior we are very likely to sample z's that have nothing to do with the x samples in our data $\mathcal{D}$ and for which the conditional $p_\theta(x_i|z)$ is nearly zero even for the best possible θ. As a result, these contribute very little to the marginal likelihood $p_\theta(x)$.

It would therefore make complete sense to sample z from some distribution that takes into account our data so that we focus on the relevant region in the $\mathcal{Z} \subseteq \mathbb{R}^d$ space. Intuitively, the best such distribution is the posterior $p(z|x)$.

Had we known the posterior, we could plug it into the the log-likelihood

$$\log p_\theta(x_i) = \log \int p(z) p_\theta(x_i|z) \, dz = \log \int p(z|x_i) \frac{p(z)}{p(z|x_i)} p_\theta(x_i|z) \, dz \tag{1.8}$$

and use the *importance sampling* strategy [5] to approximate it

$$\log p_\theta(x_i) \approx \log \frac{1}{m} \sum_j^m \frac{p(z_j)}{p(z_j|x_i)} p_\theta(x_i|z_j) \quad z_j \sim p(z|x_i) \ . \tag{1.9}$$

This sampling strategy should need fewer z samples since instead of sampling from the prior $p(z)$ over the whole $\mathcal{Z} \subseteq \mathbb{R}^d$ we are sampling with higher probability samples relevant for our data (we then adjust for it by the likelihood ratio factor).

The problem here is that we don't know the posterior and cannot easily get it since it depends again on the marginal likelihood (evidence) $p_\theta(x)$.

$$p_\theta(z|x) = \frac{p_\theta(x,z)}{p_\theta(x)} \ . \tag{1.10}$$

Note: *Pretty much a chicken and egg problem.*

1.5 Approximate posterior

The strategy is therefore to replace the intractable posterior $p(z|x)$ by some other distribution $q_\phi(z|x)$. While in principle this could be any distribution we like, even one not depending on x, I make here the dependence on x explicit because that is what we want: use x to get more reasonable samples of z.

Similarly as above, we can plug this into the log-likelihood and use the importance sampling with it

$$\log p_\theta(x_i) = \log \int p(z) p_\theta(x_i|z) \, dz = \log \int q_\phi(z|x_i) \frac{p(z)}{q_\phi(z|x_i)} p_\theta(x_i|z) \, dz \tag{1.11}$$

and use the *importance sampling* strategy [5] to approximate it

$$\log p_\theta(x_i) \approx \log \frac{1}{m} \sum_j^m \frac{p(z_j)}{q_\phi(z_j|x_i)} p_\theta(x_i|z_j) \quad z_j \sim q_\phi(z|x_i) \ . \tag{1.12}$$

Note: *It is important to understand that at this step we consider the approximate posterior $q_\phi(z_j|x_i)$ to be fixed so that we can sample from it and we will not optimize with respect to its parameters ϕ. The parameters we optimize for are the θ parameter of the decoder network $f(z_j,\theta)$ in the distributions $p_\theta(x_i|z_j) = N(\mu = f(z_j,\theta), \sigma^2 I)$. While the z_j's are different and therefore the distributions are different the parameters θ of the decoder are shared across all x's. Is this sharing what people call amortised inference?*

1.6 Encoder - inference

So how do we get this approximate posterior distribution $q_\phi(z|x)$? Typically, we assume the distribution to be a Gaussian (should be compatible with the prior on z) so that $q_\phi(z|x) = N(g_\mu(x,\phi), g_\sigma(x,\phi)I)$, where g is the ***encoder*** (the *inference*) network with x as inputs and the respective means and variances as outputs.

There are multiple ways of thinking about how to optimize it.

1.6.1 Jensen's inequality

This is a fairly classical approach though I found it rather non-intuitive. From the Jensens's inequality for log being a concave function we have

$$\log \frac{\sum_i^n x_i}{n} \geq \frac{\sum_i^n \log x_i}{n} \quad \text{and} \quad \log \mathrm{E}(x) \geq \mathrm{E}(\log x) \ . \tag{1.13}$$

Hence

$$\log p_\theta(x_i) = \log \int q_\phi(z|x_i) \frac{p(z) p_\theta(x_i|z)}{q_\phi(z|x_i)} \, dz = \log \mathrm{E}_{q_\phi(z|x_i)} \frac{p_\theta(x_i,z)}{q_\phi(z|x_i)} \geq \underbrace{\mathrm{E}_{q_\phi(z|x_i)} \log \frac{p_\theta(x_i,z)}{q_\phi(z|x_i)}}_{ELBO} \ . \tag{1.14}$$

The last term is the so called *Evidence Lower BOund* (ELBO).

Note: (1.14) *is in the form of importance weighting as discussed in the previous section.*

$$\mathrm{E}_{q_\phi(z|x_i)} \log \frac{p_\theta(x_i,z)}{q_\phi(z|x_i)} = \mathrm{E}_{q_\phi(z|x_i)} \log \frac{p_\theta(z|x_i)}{q_\phi(z|x_i)} + \mathrm{E}_{q_\phi(z|x_i)} \log p_\theta(x_i) = \log p_\theta(x_i) - D_{KL}\left(q_\phi(z|x_i) \| p_\theta(z|x_i)\right) \ . \tag{1.15}$$

In the last step above the expectation for the evidence disappears as it does not depend on z.

In fact we get something very simple here:

$$\log p_\theta(x_i) \geq \log p_\theta(x_i) - D_{KL}\left(q_\phi(z|x_i) \| p_\theta(z|x_i)\right) \ , \tag{1.16}$$

which is almost obvious if we realize that the KL divergence is always positive.

Note: *Some people say that by maximizing the log-likelihood* $\log p_\theta(x_i)$*, we are actually minimising the KL divergence between the approximate and true posterior* $D_{KL}\left(q_\phi(z|x_i) \| p_\theta(z|x_i)\right)$*. From the above, I can't see anything to support this claim. Whatever the KL, the inequality will always hold and we don't touch it or the log-likelihood by minimizing the KL. However, I can see that by maximizing the ELBO we maximize the log-likelihood and minimize the KL divergence at the same time. By working with ELBO in my head we thus move from optimising a single objective of max-likelihood to optimising a composite objective. This links to regularization perspective where the KL can be seen as a regularization term. I'm sure I've seen it discussed somewhere.*

Note: *Actually, not really. The optimization here is with respect to* ϕ *of the approximate posterior* $q_\phi(z|x_i)$*. It thus has no influence on the log-likelihood, only on the ELBO and the KL divergence. So by maximizing ELBO with respect to* ϕ *we minimize the KL divergence and do not touch the log-likelihood.*

1.6.2 Variational inference

Another point of view is starting by the objective of finding $q_\phi(z|x_i)$ which approximates well the intractable posterior $p_\theta(z|x_i)$.

To achieve this, we will minimize the KL divergence between the two

$$D_{KL}\Big(q_\phi(z|x_i)\|p_\theta(z|x_i)\Big) = \underbrace{\mathrm{E}_{q_\phi(z|x_i)}\log q_\phi(z|x_i) - \mathrm{E}_{q_\phi(z|x_i)}\log p_\theta(x_i,z)}_{-ELBO} + \log p_\theta(x_i) \;. \tag{1.17}$$

Clearly

$$\mathrm{E}_{q_\phi(z|x_i)}\log q_\phi(z|x_i) - \mathrm{E}_{q_\phi(z|x_i)}\log p_\theta(x_i,z) = -\mathrm{E}_{q_\phi(z|x_i)}\log\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)} \;. \tag{1.18}$$

Note: *Minimising the KL divergence with respect to ϕ is equivalent to maximizing the ELBO and not touching the log-likelihood. It makes no sense to minimize the KL with respect to θ as this is the distribution we want to approximate, not change. In fact we cannot minimize the KL directly with respect to ϕ either because we do not know $p_\theta(z|x_i)$ so do not know what it is we want to approximate.*

1.6.3 ELBO for optimisation

From the two previous sections we see that maximizing ELBO with respect to ϕ has the minimizing effect on the KL divergence $D_{KL}\Big(q_\phi(z|x_i)\|p_\theta(z|x_i)\Big)$ which we otherwise cannot optimize directly due to the intractable $p_\theta(z|x_i)$.

Using the results above we rewrite the ELBO once again to get a form convenient for the VAE optimisation.

$$\begin{aligned}
ELBO &= \log p_\theta(x_i) - D_{KL}\Big(q_\phi(z|x_i)\|p_\theta(z|x_i)\Big) \\
&= \mathrm{E}_{q_\phi(z|x_i)}\log\frac{p_\theta(x_i,z)}{q_\phi(z|x_i)} \\
&= \mathrm{E}_{q_\phi(z|x_i)}\log p_\theta(x_i,z) - \mathrm{E}_{q_\phi(z|x_i)}\log q_\phi(z|x_i) \\
&= \mathrm{E}_{q_\phi(z|x_i)}\log p_\theta(x_i|z) + \mathrm{E}_{q_\phi(z|x_i)}\log p(z) - \mathrm{E}_{q_\phi(z|x_i)}\log q_\phi(z|x_i) \\
&= \underbrace{\mathrm{E}_{q_\phi(z|x_i)}\log p_\theta(x_i|z)}_{\text{reconstruction}} - \underbrace{D_{KL}\Big(q_\phi(z|x_i)\|\log p(z)\Big)}_{\text{regularization}}
\end{aligned} \tag{1.19}$$

Equation (1.19) is the loss function of the VAEs. In practice, the most convenient is the last line, where the term in the left is the reconstruction loss and the KL can be seen as a regularization term.

Note: *The reconstruction loss in the left is very similar to equation (1.11) but it misses the likelihood ratio. I think this is the point of [6, 7] though I haven't read those in detail to really understand how they link the ELBO back to the importance sampling.*

Idea: *If we want to re-examine the way the VAE shall generate then this is what we need to look at.*

Note: *There is a link between the expectation maximization(EM) algorithm. I haven't worked out the details but essentially optimising the loss with respect to θ while ϕ fixed should be equivalent to the M step and optimising with respect to ϕ with θ fixed should be equivalent to the E step. In the VAEs this split is not explicit as the model is optimised end-to-end and not in alternating steps.*

1.7 Objective optimization

The above may seem very abstract but one needs to realize that maximizing the ELBO in equation (1.19) boils down to finding the θ and ϕ of the $f(z,\theta)$ and $g(x,\phi)$ functions which specify the probability distributions we are learning. And these are all typically Gaussian for simplicity and mathematical convenience.

1.7.1 The regularization term (KL divergence)

KL divergence for two Gaussian distributions $q(z) = N(\mu_q, \Sigma_q)$ and $p(z) = N(\mu_p, \Sigma_p)$ with $z \in \mathbb{R}^k$ (proof in section 1.8)

$$D_{KL}(q(z) \| p(z)) = \frac{1}{2}\left(\log\frac{|\Sigma_p|}{|\Sigma_q|} - k + \operatorname{tr}(\Sigma_p^{-1}\Sigma_q) + (\mu_q - \mu_p)^T \Sigma_p^{-1}(\mu_q - \mu_p)\right) \tag{1.20}$$

Recall that we consider the Gaussian approximate posterior $q_\phi(z|x) = N(\mu_q = g_\mu(x,\phi), \Sigma_q = g_\sigma(x,\phi)I)$ and the Gaussian prior $p(z) = N(\mu_p = 0, \Sigma_p = I)$. We thus get for the KL divergence term (see section 1.8)

$$D_{KL}\left(q_\phi(z|x_i) \| p(z)\right) = \frac{1}{2}\sum_j^k \left(-2\log\sigma_{q_j} - 1 + \sigma_{q_j}^2 + \mu_{q_j}^2\right) \ , \tag{1.21}$$

where k is the dimensionality of the latent space $\mathcal{Z} \subseteq \mathbb{R}^k$, and the means and variances are outputs of the encoder network $\mu_q = g_\mu(x,\phi)$ and $\sigma_q = g_\sigma(x,\phi)$.

Remember from (1.19) that when we *maximize* ELBO we *minimize* the D_{KL}.

Note: *I have seen in some example code that it is more convenient (for numerical reasons?) to train the encoder to output* $\log\sigma_q$ *instead of* σ_q*. One reason I can see is that* $\log\sigma_q \in \mathbb{R}^k$*, while* $\sigma_q \in \mathbb{R}_+^k$*. I use this in my code as well with the following tweak to the KL:*

$$D_{KL}\left(q_\phi(z|x_i) \| p(z)\right) = \frac{1}{2}\sum_j^k \left(-2\,ls_{q_j} - 1 + \exp 2\,ls_{q_j} + \mu_{q_j}^2\right) \ , \tag{1.22}$$

where $ls_{q_j} = \log\sigma_{q_j} = g_\sigma(x,\phi)$ and $\mu_q = g_\mu(x,\phi)$ is the variance-related and mean output of the encoder network.

Note: *In my implementation g is a single network with two sets of outputs μ_q and ls_q .*

1.7.2 The reconstruction term

The reconstruction term depends on our assumptions for the form of the conditional distribution $p(x_i|z)$. So far, we have assumed it be a Gaussian $p(x_i|z) = N(\mu = f(z,\theta), \sigma^2 I)$

$$p(x_i|z) = (2\pi)^{-d/2}\sigma^{-d}\exp\left(-\frac{1}{2}(x_i - f(z,\theta))^T\Sigma^{-1}(x_i - f(z,\theta))\right) \ . \tag{1.23}$$

The corresponding log-likelihood is

$$\log p(x_i|z) = -\frac{d}{2}\log(2\pi\sigma^2) - \left(\frac{1}{2}(x_i - f(z,\theta))^T\Sigma^{-1}(x_i - f(z,\theta))\right) \ , \tag{1.24}$$

and hence maximising the reconstruction term (with respect to θ) is equivalent to minimising the negative expectation

$$\begin{aligned} -\mathrm{E}_{q_\phi(z|x_i)}\log p_\theta(x_i|z) &= \frac{1}{2\sigma^2}\mathrm{E}_{q_\phi(z|x_i)}(x_i - f(z,\theta))^T(x_i - f(z,\theta)) \\ &\approx \frac{1}{2m\sigma^2}\sum_j^m (x_i - f(z_j,\theta))^T(x_i - f(z_j,\theta)) \qquad z_j \sim q_\phi(z|x_i) \ , \end{aligned} \tag{1.25}$$

where in the last line we approximate the expectation by its Monte-Carlo sample mean.

In case we assume a Bernoulli distribution for our data $p(x_i|z) = \pi^{x_i}(1-\pi)^{(1-x_i)}, x_i \in \{0,1\}$ with $\pi = f(z,\theta)$. The log-likelihood is

$$\log p(x_i|z) = x_i \log f(z,\theta) + (1 - x_i)\log(1 - f(z,\theta)) \tag{1.26}$$

and the reconstruction term boils down to minimising

$$- \mathrm{E}_{q_\phi(z|x_i)} \log p_\theta(x_i|z) \approx \frac{-1}{m} \sum_j^m x_i \log f(z_j, \theta) + (1 - x_i) \log(1 - f(z_j, \theta)) \qquad z_j \sim q_\phi(z|x_i) \ . \tag{1.27}$$

1.7.3 Final loss term

The optimisation problem then should be a minimization of the empirical negative ELBO for the whole dataset

$$\begin{aligned} \arg\min_{\theta,\phi} \quad & \sum_i^n \Big(\frac{1}{2} \sum_j^k \Big(-2 g_{\sigma_j}(x_i, \phi) - 1 + \exp 2 g_{\sigma_j}(x_i, \phi) + g_{\mu_j}(x_i, \phi)^2 \Big) + \\ & \frac{1}{2m\sigma^2} \sum_j^m (x_i - f(z_j, \theta))^T (x_i - f(z_j, \theta)) \qquad z_j \sim q_\phi(z|x_i) \Big) \end{aligned} \tag{1.28}$$

In the above, σ^2 is typically fixed to 1 or can be used as a regularisation parameter which would probably lead onto something very similar to β-VAE. Optimisation of the decoder network with respect to θ should be rather straightforward as soon as we sample $z_j \sim q_\phi(z|x_i)$. What Kingma suggested is actually not to sample m times for each observation but only once. Since we use a stochastic gradient descent where each sample will be revisited multiple times, in the end we will have multiple samples of z's, though each from a somewhat different distribution $q_\phi(z|x_i)$ as the encoder network parameters ϕ got updated in the meantime.

The update for ϕ is a little more tricky. In addition calculating the gradients through the decoder network g, the ϕ parameters also play a role in the sampling of z.

For this Kingma suggested the **reparametrization trick**. Essentially, we rewrite z as a deterministic transformation of a random variable $\epsilon \sim N(0,1)$ so that $z_j = g_{\mu_j}(x_i, \phi) + \exp g_{\sigma_j}(x_i, \phi) \epsilon_j$, where I use exp because the output of my encoder network is $\log \sigma$.

1.8 Proofs

Proof: Multivariate Gaussian KL divergence.

$$q(z) = N(\mu_q, \Sigma_q) = (2\pi)^{-k/2} |\Sigma_q|^{-1/2} \exp\Big(-\frac{1}{2} (x - \mu_q)^T \Sigma_q^{-1} (x - \mu_q) \Big) \tag{1.29}$$

$$\log q(z) = -\frac{k}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_q| - \frac{1}{2} (x - \mu_q)^T \Sigma_q^{-1} (x - \mu_q) \tag{1.30}$$

$$\begin{aligned} D_{KL}(q(z) \| p(z)) &= \mathrm{E}_{q(z)} \log q(z) - \mathrm{E}_{q(z)} \log p(z) \\ &= \int q(z) \log q(z) \, dz - \int q(z) \log p(z) \, dz \\ &= \frac{1}{2} \Big(\log \frac{|\Sigma_p|}{|\Sigma_q|} - k + \mathrm{tr}\,(\Sigma_p^{-1} \Sigma_q) + (\mu_q - \mu_p)^T \Sigma_p^{-1} (\mu_q - \mu_p) \Big) \end{aligned} \tag{1.31}$$

$$
\begin{aligned}
\mathrm{E}_{q(z)} \log q(z) &= \mathrm{E}_{q(z)}\left(-\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}(x-\mu_q)^T \Sigma_q^{-1}(x-\mu_q)\right) \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \mathrm{E}_{q(z)}\left(\frac{1}{2}(x-\mu_q)^T \Sigma_q^{-1}(x-\mu_q)\right) \quad \text{(E of constants)} \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}\mathrm{E}_{q(z)}\,\mathrm{tr}\left((x-\mu_q)^T \Sigma_q^{-1}(x-\mu_q)\right) \quad \text{(trace of scalar)} \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}\mathrm{tr}\ \mathrm{E}_{q(z)}\left(\Sigma_q^{-1}(x-\mu_q)(x-\mu_q)^T\right) \quad \text{(linearity of E)} \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}\mathrm{tr}\ \Sigma_q^{-1}\mathrm{E}_{q(z)}\left((x-\mu_q)(x-\mu_q)^T\right) \quad \text{(linearity of E)} \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}\mathrm{tr}\ \Sigma_q^{-1}\Sigma_q \quad \text{(definition of } \Sigma_q\text{)} \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_q| - \frac{k}{2} \quad (= -H(x) \sim \text{entropy}) \qquad (1.32)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}_{q(z)} \log p(z) &= \mathrm{E}_{q(z)}\left(-\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_p| - \frac{1}{2}(x-\mu_p)^T \Sigma_p^{-1}(x-\mu_p)\right) \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_p| - \frac{1}{2}\mathrm{tr}\ \Sigma_p^{-1}\mathrm{E}_{q(z)}\left((x-\mu_p)(x-\mu_p)^T\right) \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_p| - \frac{1}{2}\mathrm{tr}\ \Sigma_p^{-1}\mathrm{E}_{q(z)}\left([(x-\mu_q)+(\mu_q-\mu_p)][(x-\mu_q)+(\mu_q-\mu_p)]^T\right) \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_p| - \\
&\quad \frac{1}{2}\mathrm{tr}\ \Sigma_p^{-1}\left(\mathrm{E}_{q(z)}(x-\mu_q)(x-\mu_q)^T + 2\mathrm{E}_{q(z)}(x-\mu_q)(\mu_q-\mu_p)^T + \mathrm{E}_{q(z)}(\mu_q-\mu_p)(\mu_q-\mu_p)^T\right) \\
&= -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_p| - \frac{1}{2}\mathrm{tr}\ \Sigma_p^{-1}\Sigma_q + 0 + (\mu_q-\mu_p)^T\Sigma_p^{-1}(\mu_q-\mu_p) \qquad (1.33)
\end{aligned}
$$

□

Proof: KL divergence between $q_\phi(z|x) = N(\mu_q = g_\mu(x,\phi), \Sigma_q = g_\sigma(x,\phi)I)$ and $p(z) = N(\mu_p = 0, \Sigma_p = I)$.

$$
\begin{aligned}
D_{KL}\left(q_\phi(z|x_i)\,\|\,p(z)\right) &= \frac{1}{2}\left(\log\frac{1}{\prod_j^k \sigma_{q_j}^2} - k + \sum_j^k \sigma_{q_j}^2 + \mu_q^T\mu_q\right) \\
&= \frac{1}{2}\left(-2\sum_j^k \log\sigma_{q_j} - k + \sum_j^k \sigma_{q_j}^2 + \mu_q^T\mu_q\right) \\
&= \frac{1}{2}\sum_j^k\left(-2\log\sigma_{q_j} - 1 + \sigma_{q_j}^2 + \mu_{q_j}^2\right) \qquad (1.34)
\end{aligned}
$$

□

References

[1] Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).

[2] Kingma, Diederik P. "Variational inference & deep learning: A new synthesis." (2017).

[3] https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

[4] https://ermongroup.github.io/cs228-notes/extras/vae/

[5] https://statweb.stanford.edu/ owen/mc/Ch-var-is.pdf

[6] Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders." arXiv preprint arXiv:1509.00519 (2015).

[7] Cremer, Chris, Quaid Morris, and David Duvenaud. "Reinterpreting importance-weighted autoencoders." arXiv preprint arXiv:1704.02916 (2017).

2 Quick notes on logistic regression

Based in parts on these CMU lectures, this book chapter and wikipedia on logistic and multinomial logistic regression.

2.1 Logistic sigmoid

In logistic regression the output variable $y \in \{0,1\}$. Still, we would like to use linear model in the form $\mathbf{w}^T\mathbf{x}$. However, the result of this linear model lives in $\mathbb{R}$ while $y \in \{0,1\}$.

We will therefore transform the y variable. First, instead treating it as discrete $\{0,1\}$ we will treat it as continuous in the interval $[0,1]$ essentially indicating the probability $P(y = 1) = \pi \in [0,1]$. Another, and perhaps better, way of thinking about this is that the y variable is generated from a Bernoulli distribution conditioned on $\mathbf{x}$ with the expectation $\mathrm{E}[y|\mathbf{x}] = \pi$. What the model shall predict is not y directly but rather $\mathrm{E}[y|\mathbf{x}] = \pi$.

Working with $\pi \in [0,1]$ is still not good enough since the outputs of the linear model are in $\mathbb{R}$. We need to come up with an invertible transformation g (*link function*) so that $g(\mathrm{E}(y|\mathbf{x})) = g(\pi) \in \mathbb{R}$.

We start from odds of $y = 1$ which is

$$\text{odds } = \frac{count(y=1)}{count(y=0)} = \frac{P(y=1)}{P(y=0)} = \frac{P(y=1)}{1-P(y=1)} = \frac{\pi}{1-\pi} \ .$$

The odds live in $[0,\infty)$. To finally get to something which lives in $\mathbb{R}$ we take the log of the odds $\log \text{odds} = \log \frac{\pi}{1-\pi} \in \mathbb{R}$ also called *logits*. The linear model outputs therefore shall be the logits. Good way to think about this is as the linear model for a transformation of the expected response. The output of the model is often indicated as z and called the *score*. Here it is simply the linear model, but it could be preceded by a whole network with a final linear layer.

$$\log \frac{\pi}{1-\pi} = \mathbf{w}^T\mathbf{x} = z \ . \tag{2.1}$$

It is rather easy to get the predictions in terms of the expectation of y, the probability $\mathrm{E}(y|\mathbf{x}) = \pi$, as the inverse of the logit *link function*

$$\begin{aligned}
\frac{\pi}{1-\pi} &= \exp(\mathbf{w}^T\mathbf{x}) \\
\pi &= \exp(\mathbf{w}^T\mathbf{x}) - \pi \exp(\mathbf{w}^T\mathbf{x}) \\
\pi(1+\exp(\mathbf{w}^T\mathbf{x})) &= \exp(\mathbf{w}^T\mathbf{x}) \\
\mathrm{E}(y|\mathbf{x}) = \pi &= \frac{\exp(\mathbf{w}^T\mathbf{x})}{1+\exp(\mathbf{w}^T\mathbf{x})} = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})} = \sigma(\mathbf{w}^T\mathbf{x}) \ , \qquad (2.2)
\end{aligned}$$

where $\sigma : \mathbb{R} \to [0,1]$ is the *logistic sigmoid* function.

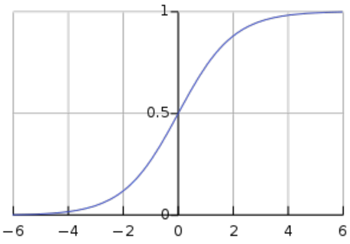

Figure 1: Logistic sigmoid; source Wikipedia

2.2 Cross-entropy loss

For the binary y the conditional probability distribution $p(y|\mathbf{x})$ is Bernouli with the conditional expectation $\pi = \mathrm{E}(y|\mathbf{x})$. We can formulate the logistic regression problem objective as maximizing the likelihood of $\mathbf{w}$ over the training set

$$\max_{\mathbf{w}} \prod_i^n \pi_i^{y_i}(1-\pi_i)^{1-y_i} \ , \tag{2.3}$$

where $\pi_i = \sigma(\mathbf{x}_i^T\mathbf{w})$ from equation (2.2).

We can instead minimize the negative log likelihood also called the **cross-entropy loss**

$$\min_{\mathbf{w}} \sum_i^n -y_i \log \pi_i - (1-y_i)\log(1-\pi_i) \ , \tag{2.4}$$

Recall the definition of cross-entropy

$$H_p(q) = -\sum_c^C p(y_c)\log q(y_c) \ , \tag{2.5}$$

where y is a random variable with C categories and p and q are two different distributions.

2.3 Multi-class classification

When we have more than 2 classes, we consider a linear model of the form $\mathbf{x}^T\mathbf{w}_c = z_c$ with specific parameters $\mathbf{w}_c$ for each class and z_c the per-class *scores*.

We can arrive at multinomial logistic regression following similar logic as in the binary case where we decompose the multiple categories $c = 1, \dots, C$ into a set of $C-1$ dummy variables with the last category as the default pivot.

For each class we have the odds against the pivot as

$$\text{odds}_c = \frac{count(y=c)}{count(y=C)} = \frac{P(y=c)}{P(y=C)} = \frac{\pi_c}{\pi_C} \ . \tag{2.6}$$

The scores z_c are the log-odds, the logits

$$z_c = \log\frac{\pi_c}{\pi_C} = \mathbf{x}^T\mathbf{w}_c \tag{2.7}$$

from which we get

$$\pi_c = \pi_C \exp(\mathbf{x}^T\mathbf{w}_c) \qquad \text{for all } c = 1, \dots, C-1 \tag{2.8}$$

Because probabilities some to one, we have

$$\pi_C = 1 - \sum_c^{C-1}\pi_c = 1 - \sum_c^{C-1}\pi_C \exp(\mathbf{x}^T\mathbf{w}_c) = 1 - \pi_C\sum_c^{C-1}\exp(\mathbf{x}^T\mathbf{w}_c) \tag{2.9}$$

and therefore

$$\pi_C = \frac{1}{1+\sum_c^{C-1}\exp(\mathbf{x}^T\mathbf{w}_c)} \qquad \pi_c = \frac{\exp(\mathbf{x}^T\mathbf{w}_c)}{1+\sum_c^{C-1}\exp(\mathbf{x}^T\mathbf{w}_c)} \tag{2.10}$$

This is obviously equal to the binary logistic sigmoid in case of just two classes where $\pi_c = P(y = 1)$ and $\pi_C = P(y = 0)$.

Soft-max To get the *soft-max* function instead of fixing one category as a pivot we treat all the probabilities evenly. This will in the end lead to overparametrization because we do not treat one class as the default.

As above, we have for the probabilities of each class

$$\pi_c = \frac{1}{Z}\exp(\mathbf{x}^T\mathbf{w}_c) \qquad \text{for all } c = 1,\ldots,C \ , \tag{2.11}$$

which is now valid for all classes and where Z is a common normalizing constant.

Since probability has to sum to 1 we have

$$\begin{aligned} 1 = \sum_c^C \pi_c &= \frac{1}{Z}\sum_c^C \exp(\mathbf{x}^T\mathbf{w}_c) \\ Z &= \sum_c^C \exp(\mathbf{x}^T\mathbf{w}_c) \\ \pi_c &= \frac{\exp(\mathbf{x}^T\mathbf{w}_c)}{\sum_c^C \exp(\mathbf{x}^T\mathbf{w}_c)} \ . \end{aligned} \tag{2.12}$$

The coefficients in the soft-max are redundant (not uniquely identifiable), because the values of the soft-max will not change if we add a constant vector α to each of the parameters vector

$$\frac{\exp(\mathbf{x}^T(\mathbf{w}_c+\alpha))}{\sum_c^C \exp(\mathbf{x}^T(\mathbf{w}_c+\alpha))} = \frac{\exp(\mathbf{x}^T\mathbf{w}_c)\exp(\mathbf{x}^T\alpha)}{\exp(\mathbf{x}^T\alpha)\sum_c^C \exp(\mathbf{x}^T\mathbf{w}_c)} = \frac{\exp(\mathbf{x}^T\mathbf{w}_c)}{\sum_c^C \exp(\mathbf{x}^T\mathbf{w}_c)} \ . \tag{2.13}$$

If we fix $\alpha = -\mathbf{w}_C$ to the parameters of the last class we get

$$\pi_C = \frac{\exp(\mathbf{x}^T(\mathbf{w}_C-\mathbf{w}_C))}{\sum_c^C \exp(\mathbf{x}^T(\mathbf{w}_c-\mathbf{w}_C))} = \frac{1}{1+\sum_c^{C-1}\exp(\mathbf{x}^T(\mathbf{w}_c-\mathbf{w}_C))} \tag{2.14}$$

and

$$\pi_c = \frac{\exp(\mathbf{x}^T(\mathbf{w}_c-\mathbf{w}_C))}{\sum_c^C \exp(\mathbf{x}^T(\mathbf{w}_c-\mathbf{w}_C))} = \frac{\exp(\mathbf{x}^T(\mathbf{w}_c-\mathbf{w}_C))}{1+\sum_c^{C-1}\exp(\mathbf{x}^T(\mathbf{w}_c-\mathbf{w}_C))} \ , \tag{2.15}$$

which is the same result as in equation (2.10) with the shifted weight vectors.

3 Evaluate data log likelihood by importance sampling

In addition to evaluting the ELBO across the test samples, it makes sense to evaluate the log-likelihood of the test data. The ELBO is formulated each time differently, depending on the graphical model etc., while the log-likelihood should be measuring always the same.

From the VAE basic assumptions we have the latent variable model

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz \ , \tag{3.1}$$

where $p(\mathbf{z})$ is the prior for the latent $\mathbf{z}$, and $p(\mathbf{x}|\mathbf{z})$ is the learned conditional likelihood - the decoder of the VAE.

Hence, the log-likelihood of observation $\mathbf{x}$ is

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz \ . \tag{3.2}$$

Empirically we could get this by sampling $\mathbf{z}$ from the prior $p(\mathbf{z})$

$$\log p(\mathbf{x}) \approx \log \sum_i^K p(\mathbf{x}|\mathbf{z}_i), \quad \mathbf{z}_i \sim p(\mathbf{z}) \ . \tag{3.3}$$

The problem with this one is that we may need a lot of samples to get a reasonable estimate of the log-likelihood - according to the prior we may be sampling $\mathbf{z}$ in an area very unlikely for any $\mathbf{x}$.

Instead, adopting the importance sampling principles we may sample from the learned posterior $q(\mathbf{z}|\mathbf{x})$ (the VAE decoder) so that

$$\begin{aligned} \log p(\mathbf{x}) &= \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})}dz \\ &= \log \int p(\mathbf{x}|\mathbf{z})q(\mathbf{z}|\mathbf{x})\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}dz \ , \end{aligned} \tag{3.4}$$

with the empirical estimate

$$\log p(\mathbf{x}) \approx \log \sum_i^K p(\mathbf{x}|\mathbf{z}_i)\frac{p(\mathbf{z}_i)}{q(\mathbf{z}_i|\mathbf{x})}, \quad \mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x}) \ . \tag{3.5}$$

What we have in the loss-function (maximization of the ELBO) is the log of the probabilities. Therefore I introduce these in to the importance sampled log-likelihood

$$\begin{aligned} \log p(\mathbf{x}) &= \log \int q(\mathbf{z}|\mathbf{x})p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}dz \\ &= \log \int q(\mathbf{z}|\mathbf{x}) \exp\left[\log p(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x})\right] dz \ , \end{aligned} \tag{3.6}$$

with the empirical estimate

$$\begin{aligned} \log p(\mathbf{x}) &\approx \log \sum_i^K \exp\left[\log p(\mathbf{x}|\mathbf{z}_i) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i|\mathbf{x})\right], \quad \mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x}) \\ &= \log \sum_i^K \exp\left[\log p(\mathbf{x}|\mathbf{z}_i) - \log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i)}\right], \quad \mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x}) \ . \end{aligned} \tag{3.7}$$

Here $\log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i)}$ is the log probability ratio and is better evaluated in the logs as

$$\log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i)} = \log q(\mathbf{z}_i|\mathbf{x}) - \log p(\mathbf{z}_i) \ . \tag{3.8}$$

Note that this is the single sample empirical estimate of the KL divergence which is the same as used in the ELBO calculation in the vampprior implementation of Tomczek (though in the ELBO there it is average across multiple samples).

The first term $\log p(\mathbf{x}|\mathbf{z}_i)$ is simply the log conditional likelihood that is the RE part of the ELBO for a single sample $\mathbf{z}$.

To evaluate the logsumexp it is safer for numerical stability to use the equivalent form

$$\log p(\mathbf{x}) \approx a + \log \sum_i^K \exp(a_i - a) \ , \tag{3.9}$$

where $a_i = \log p(\mathbf{x}|\mathbf{z}_i) - \log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i)}$ and $a = \max_i a_i$.

This is actually often already implemented in python for example in the *tf.reduce_logsumexp*.

4 Some useful inequalities (or equalities) - in progress

Some useful inequalities or equivalences found around and worth remembering.

4.1 Numerical inequalities

Based on [1].

4.1.1 Triangle inequality

$$|X+Y| \leq |X| + |Y| \tag{4.1}$$

4.1.2 Holder's inequality

Let $a, b > 0$ and $p, q > 1$ be any numbers satisfying

$$\frac{1}{p} + \frac{1}{q} = 1 \qquad p + q = pq \qquad (p-1)q = p \tag{4.2}$$

then

$$\frac{1}{p}a^p + \frac{1}{q}b^q \geq ab \tag{4.3}$$

with equality only if $a^p = b^q$.

Proof: Fix b and minimize the function $g(a) = \frac{1}{p}a^p + \frac{1}{q}b^q - ab$. To minimize, we set the derivative equal to zero $dg(a) = a^{p-1} - b = 0 \Rightarrow b = a^{p-1}$. The value of the function at minimum is $\frac{1}{p}a^p + \frac{1}{q}(a^{p-1})^q - aa^{p-1} = (\frac{1}{p} - 1)a^p + \frac{1}{q}a^p = 0$. So the minimum is 0 and (4.3) is established.

We use this to get the **Holder's inequality** Let X and Y be two r.v. and let p, q satisfy (4.2). Then

$$|\mathrm{E}XY| \leq \mathrm{E}|XY| \leq (\mathrm{E}|X|^p)^{1/p} (\mathrm{E}|Y|^q)^{1/q} \tag{4.4}$$

Proof: The first inequality follows from the Jensen's inequality (4.13). Define $a = \frac{|X|}{(\mathrm{E}|X|^p)^{1/p}}$ and $b = \frac{|Y|}{(\mathrm{E}|Y|^q)^{1/q}}$. Applying (4.3) we have

$$\begin{aligned}
\frac{1}{p}\frac{|X|^p}{\mathrm{E}|X|^p} + \frac{1}{q}\frac{|Y|^q}{\mathrm{E}|Y|^q} &\geq \frac{|XY|}{(\mathrm{E}|X|^p)^{1/p}(\mathrm{E}|Y|^q)^{1/q}} \\
\frac{1}{p}\frac{\mathrm{E}|X|^p}{E|X|^p} + \frac{1}{q}\frac{\mathrm{E}|Y|^q}{\mathrm{E}|Y|^q} &\geq \frac{\mathrm{E}|XY|}{(\mathrm{E}|X|^p)^{1/p}(\mathrm{E}|Y|^q)^{1/q}} \qquad \text{(taking expectation on both sides)} \\
1 &\geq \frac{\mathrm{E}|XY|}{(\mathrm{E}|X|^p)^{1/p}(\mathrm{E}|Y|^q)^{1/q}} \\
(\mathrm{E}|X|^p)^{1/p}(\mathrm{E}|Y|^q)^{1/q} &\geq \mathrm{E}|XY| \qquad \text{QED}
\end{aligned}$$

Cauchy-Schwarz inequality is the special case of Hodler's for $p = q = 2$

$$|\mathrm{E}XY| \leq \mathrm{E}|XY| \leq (\mathrm{E}|X|^2)^{1/2} (\mathrm{E}|Y|^2)^{1/2} \tag{4.5}$$

An example of CS is the **covariance inequality**

$$\begin{gathered}
|\mathrm{E}(X - \mathrm{E}X)(Y - \mathrm{E}Y)| \leq \mathrm{E}|(X - \mathrm{E}X)(Y - \mathrm{E}Y)| \leq (\mathrm{E}(X - \mathrm{E}X)^2)^{1/2} (\mathrm{E}(Y - \mathrm{E}Y)^2)^{1/2} \\
Cov(X, Y)^2 \leq (\mathrm{E}(X - \mathrm{E}X)^2)(\mathrm{E}(Y - \mathrm{E}Y)^2) = \rho_X^2 \rho_Y^2
\end{gathered} \tag{4.6}$$

Another form of **covariance inequality** for two functions of a random variable states that for $g(X), h(X)$ **both non-decreasing or both non-increasing** and therefore having $Cov(g(X)h(X)) \geq 0$

$$\mathrm{E}(g(X)h(X)) \geq \mathrm{E}g(X)\mathrm{E}h(X) \ . \tag{4.7}$$

For $g(X), h(X)$ **one non-decreasing the other non-increasing** and therefore having $Cov(g(X)h(X)) \leq 0$

$$\mathrm{E}(g(X)h(X)) \leq \mathrm{E}g(X)\mathrm{E}h(X) \ . \tag{4.8}$$

Other useful variants:

Let $Y = 1$, we get $\mathrm{E}|X| \leq (\mathrm{E}|X|^p)^{1/p}$ for $1 < p < \infty$.

Let $Y = 1$ and $1 < r < p < \infty$ we get $\mathrm{E}|X|^r \leq (\mathrm{E}|X|^{rp})^{1/p}$.

For **Liupanov's inequality** put $s = pr \Rightarrow 1/p = r/s$ and observe that $s > r$ so that $1 < r < s < \infty$. By rearranging the above we get $(\mathrm{E}|X|^r)^{1/r} \leq (\mathrm{E}|X|^s)^{1/s}$.

4.1.3 Minkowski's inequality

Let X, Y be r.v., then for $1 \leq p < \infty$

$$(\mathrm{E}|X+Y|^p)^{1/p} \leq (\mathrm{E}|X|^p)^{1/p} + (\mathrm{E}|Y|^p)^{1/p} \tag{4.9}$$

Proof: From the triangle inequality (4.1) we have

$$\mathrm{E}|X+Y|^p = \mathrm{E}\left(|X+Y||X+Y|^{p-1}\right) \leq \mathrm{E}\left(|X||X+Y|^{p-1}\right) + \mathrm{E}\left(|Y||X+Y|^{p-1}\right) \ .$$

Using Hodler's to the terms on the right side we have

$$\mathrm{E}\left(|X||X+Y|^{p-1}\right) \leq (\mathrm{E}|X|^p)^{1/p}(\mathrm{E}|X+Y|^{q(p-1)})^{1/q}$$

with p, q satisfying (4.2) and therefore

$$\begin{aligned}
\mathrm{E}|X+Y|^p &\leq (\mathrm{E}|X|^p)^{1/p}\left(\mathrm{E}|X+Y|^{q(p-1)}\right)^{1/q} + (\mathrm{E}|Y|^p)^{1/p}\left(\mathrm{E}|X+Y|^{q(p-1)}\right)^{1/q} \\
\frac{\mathrm{E}|X+Y|^p}{\left(\mathrm{E}|X+Y|^{q(p-1)}\right)^{1/q}} &\leq (\mathrm{E}|X|^p)^{1/p} + (\mathrm{E}|Y|^p)^{1/p} \\
(\mathrm{E}|X+Y|^p)^{1-1/q} &\leq (\mathrm{E}|X|^p)^{1/p} + (\mathrm{E}|Y|^p)^{1/p} \qquad q(p-1) = p \text{ from (4.2)} \\
(\mathrm{E}|X+Y|^p)^{1/p} &\leq (\mathrm{E}|X|^p)^{1/p} + (\mathrm{E}|Y|^p)^{1/p} \qquad \text{QED}
\end{aligned}$$

4.2 Optimization

4.2.1 Change max to min

$$\min f(x) = -\max -f(x) \tag{4.10}$$

Proof: $m \leq f(x)$ for any $f(x)$ is the minimum. We know that $-m \geq -f(x)$ which tells us that $-m$ is the maximum of $-f(x)$, that is $-m = \max -f(x)$. Multiplying both sides by -1 we get the result.

4.2.2 Min of monotonic function

For a non-decreasing function f we have $\max f(x) = f(\max x)$.

Proof: If f is non-decreasing then for all $x \geq x'$ we have $f(x) \geq f(x')$. For $\max f(x) = f^* = f(x^*)$ we know that $f(x^*) \geq f(x)$ and therefore $x^* \geq x$ for any x which gives $x^* = \max x$ and therefore $\max f(x) = f(\max x)$.

4.2.3 Min of exponential

Using the previous results we have (exp is monotonically increasing)

$$\min \exp(-a) = \exp(\min -a) = \exp(-\max a) \tag{4.11}$$

4.2.4 Jensen's inequality

A function f is convex if for all $x_1, x_2 \in \mathcal{X}$ and all $t \in [0,1]$ we have

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) \tag{4.12}$$

In words, convex function of an average of two points is below the average of the function evaluations at the two points.

This can be extended from averages to expectations for a convex function f (e.g. exp)

$$f(\mathrm{E}X) \leq \mathrm{E}f(X) \ . \tag{4.13}$$

For a concave function g (e.g. log) the inequality reverses into

$$g(\mathrm{E}X) \geq \mathrm{E}g(X) \ . \tag{4.14}$$

4.3 Basic probability

4.3.1 Union bound

Based on [1].

For a countable set of events $A_1, A_2, \ldots, A_n$ we have

$$\mathrm{P}(\cup_i^n A_i) \leq \sum_i^n \mathrm{P}(A_i) \tag{4.15}$$

Proof: From the basic laws of probability we have $\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \cap B) \leq \mathrm{P}(A) + \mathrm{P}(B)$. We indicate by $A = \cup_i^{n-1} A_i$ and $B = A_n$ to get

$$\mathrm{P}(\cup_i^n A_i) = \mathrm{P}(\cup_i^{n-1} A_i \cup A_n) \leq \mathrm{P}(\cup_i^{n-1} A_i) + \mathrm{P}(A_n) \tag{4.16}$$

from which by induction we get the result.

4.4 Variable transformation

Based on [3].

We have a r.v. X taking values in a set S with a known probability distribution P.

A function $g : S \to T$ is a new r.v. $Y = g(X)$ with values in T.

For $B \subseteq T$, $g^{-1}(B) = \{x \in S : g(x) \in B\}$ is the *inverse image* (preimage) of B under g and the probability is $P(Y \in B) = P(g(X) \in B) = P(X \in g^{-1}(B))$.

If the function $g(x) = y$ is **one-to-one** than it has an inverse *function* $g^{-1}(y) = x$ which is as well one-to-one.

If $g(x) = y$ is **strictly increasing** then the cumulative distribution function is

$$G(y) = P(Y \leq y) = P(g(X) \leq g(x)) = P(X \leq x) = F(x) \ , \tag{4.17}$$

where F is the cdf of X and $x = g^{-1}(y)$.

If $g(x) = y$ is **strictly decreasing** then

$$G(y) = P(Y \leq y) = P(g(X) \leq g(x)) = P(X \geq x) = 1 - F(x) \ . \tag{4.18}$$

However, if the function $g(x) = y$ is **many-to-one** than it does not have an inverse *function*, and $g^{-1}(y) = \{x_1, x_2, \ldots\}$ is the one-to-many inverse image. In the general case, we cannot make similar claims about cdf (or pdf) of Y as above.

Still, for a **non-decreasing** function $g(x) = y$ with an inverse image $g^{-1}(y) = (x_1, x_2)$, $x_1 \leq x_2$ we have

$$\begin{array}{rrcl}
\text{case 1} & G(y) = P(Y \leq y) = P(g(X) \leq g(x_1)) & > & P(X \leq x_1) = F(x_1) \\
\text{case 2} & G(y) = P(Y \leq y) = P(g(X) \leq g(x_2)) & = & P(X \leq x_2) = F(x_2) \\
\text{in general} & G(y) = P(Y \leq y) = P(g(X) \leq g(x)) & \geq & P(X \leq x) = F(x) \\
 & & \text{and} & \\
\text{case 1} & 1 - G(y) = P(Y \geq y) = P(g(X) \geq g(x_1)) & = & P(X \geq x_1) = 1 - F(x_1) \\
\text{case 2} & 1 - G(y) = P(Y \geq y) = P(g(X) \geq g(x_2)) & > & P(X \geq x_2) = 1 - F(x_2) \\
\text{in general} & 1 - G(y) = P(Y \geq y) = P(g(X) \geq g(x)) & \geq & P(X \geq x) = 1 - F(x)
\end{array} \tag{4.19}$$

Table 1: Example of probability distribution of a non-decreasing variable transformation $g(x)$.

x	$P(X = x)$	$P(X \leq x)$	$P(X \geq x)$	$g(x)$	$P(g(X) = g(x))$	$P(g(X) \leq g(x))$	$P(g(X) \geq g(x))$
1	1/6	1/6	1	2	1/3	1/3	1
2	1/6	1/3	5/6				
3	1/6	1/2	2/3	4	1/3	2/3	2/3
4	1/6	2/3	1/2				
5	1/6	5/6	1/3	6	1/3	1	1/3
6	1/6	1	1/6				

4.5 Concentration bounds

Based on [2].

By concentration inequality we usually mean an upper bound for the probability that a r.v. Z differs from its expected value by more than some amount $t > 0$. That is we seek upper bounds on the probabilities in the form

$$\mathrm{P}(Z - \mathrm{E}Z \geq t) \quad \text{and} \quad \mathrm{P}(Z - \mathrm{E}Z \leq -t) \tag{4.20}$$

or equivalently on the probability

$$\mathrm{P}(|Z - \mathrm{E}Z| \geq t) \ . \tag{4.21}$$

4.5.1 Markov's inequality

Based on [1].

For a non-negative random variable Y, i.e. $\mathrm{P}(Y < 0) = 0$, and all $t > 0$ we have

$$\mathrm{P}(Y \geq t) \leq \frac{\mathrm{E}Y}{t} \tag{4.22}$$

By fixing $t = r\mathrm{E}Y$ we get the equivalent

$$\mathrm{P}(Y \geq r\mathrm{E}Y) \leq \frac{1}{r} \tag{4.23}$$

Proof: Following [1]. For a given $t > 0$ it holds that

$$\begin{array}{rcll} t\,\mathbb{1}(Y \geq t) & \leq & Y & (\mathbb{1}(Y \geq t) = 1 \text{ if } Y \geq t \text{ and } 0 \text{ otherwise}) \\ \mathrm{E}(t\,\mathbb{1}(X \geq t)) & \leq & EY & \text{(by monotonicity of expectation)} \\ t\Big(1\,\mathrm{P}(Y \geq t) + 0\,\mathrm{P}(Y < t)\Big) & \leq & \mathrm{E}Y & \text{(expanding the expectation)} \\ \mathrm{P}(Y \geq t) & \leq & \frac{\mathrm{E}Y}{t} & \text{QED} \end{array}$$

We can apply (4.22) to a non-negative random function $Y = g(Z)$ of r.v. Z taking values in $I \subseteq \mathbb{R}$ so that

$$\mathrm{P}(g(Z) \geq t) \leq \frac{\mathrm{E}g(Z)}{t} \tag{4.24}$$

and for non-decreasing nonnegative g with $Z, t \in I \subseteq \mathbb{R}$ so that $g(t) > 0$

$$\mathrm{P}(Z \geq t) \leq \mathrm{P}(g(Z) \geq g(t)) \leq \frac{\mathrm{E}g(Z)}{g(t)} \tag{4.25}$$

4.5.2 Chebyshev's inequality

Based on [2].

Taking $g(t) = t^2$ and $Y = |Z - \mathrm{E}Z|$, we get from (4.25)

$$\begin{array}{rcl} \mathrm{P}(|Z - \mathrm{E}Z|^2 \geq t^2) & \leq & \frac{\mathrm{E}(|Z - \mathrm{E}Z|^2)}{t^2} \\ \mathrm{P}(|Z - \mathrm{E}Z| \geq t) & \leq & \frac{Var Z}{t^2} \end{array} \tag{4.26}$$

More generally, we may take $g(t) = t^q$ for some $q > 0$ and all $t > 0$ to get the general **moment bounds**

$$\mathrm{P}(|Z - \mathrm{E}Z| \geq t) \quad \leq \quad \frac{\mathrm{E}(|Z - \mathrm{E}Z|^q)}{t^q} \tag{4.27}$$

and we may choose q to optimize the upper bound.

Nevertheless, variance (that is $q = 2$) is probably the easiest to handle.

For a **sum of independent** r.v. $Z = \sum_i^n X_i$ we have for the expectation $\mathrm{E}Z = \sum_i^n \mathrm{E}X_i$, for the variance $VarZ = \sum_i^n VarX_i$ and therefore from (4.26)

$$\begin{array}{rcll} \mathrm{P}\left(\left|\sum_i^n X_i - \mathrm{E}\sum_i^n X_i\right| \geq t\right) & \leq & \frac{Var \sum_i^n X_i}{t^2} & \\ \mathrm{P}\left(\left|\sum_i^n (X_i - \mathrm{E}X_i)\right| \geq t\right) & \leq & \frac{\sum_i^n VarX_i}{t^2} & \\ \mathrm{P}\left(\frac{1}{n}\left|\sum_i^n (X_i - \mathrm{E}X_i)\right| \geq \frac{t}{n}\right) & \leq & \frac{\sum_i^n VarX_i}{t^2} & \\ \mathrm{P}\left(\frac{1}{n}\left|\sum_i^n (X_i - \mathrm{E}X_i)\right| \geq r\right) & \leq & \frac{n^{-1}\sum_i^n VarX_i}{nr^2} & \frac{t}{n} = r,\ t^2 = r^2 n^2 \end{array} \tag{4.28}$$

4.5.3 Cramer-Chernoff method

Based on [2].

For r.v. $Z \in \mathbb{R}$ and $g(t) = e^{\lambda t}$ where $\lambda > 0$ we get from Markov's (4.25)

$$\mathrm{P}(Z \geq t) \leq \frac{\mathrm{E}e^{\lambda Z}}{e^{\lambda t}} \quad , \tag{4.29}$$

where $F(\lambda) = \mathrm{E}e^{\lambda Z}$ is the *moment generating function* (in general defined for all $\lambda \in \mathbb{R}$). We will optimize for $\lambda \geq 0$ to get the best possible concentration bound.

Define $\psi_Z(\lambda) := \log \mathrm{E}e^{\lambda Z}$ for all $\lambda \geq 0$ and introduce *Cramer transform* of Z

$$\psi_Z^*(t) = \sup_{\lambda \geq 0}(\lambda t - \psi_Z(\lambda)) \tag{4.30}$$

From (4.29) we get the **Chernoff's inequality**

$$\mathrm{P}(Z \geq t) \leq \exp(-\psi_Z^*(t)) \tag{4.31}$$

Proof: Chernoff's inequality

$$\begin{aligned}
\mathrm{P}(Z \geq t) &\leq e^{-\lambda t}\mathrm{E}e^{\lambda Z} = \exp\log(e^{-\lambda t}\mathrm{E}e^{\lambda Z}) = \exp(\psi_Z(\lambda) - \lambda t) \\
\mathrm{P}(Z \geq t) &\leq \min_{\lambda} \exp(\psi_Z(\lambda) - \lambda t) \qquad \text{(true for any } \lambda \geq 0 \text{ so also for } \min) \\
\mathrm{P}(Z \geq t) &\leq \exp(-\max \lambda t - \psi_Z(\lambda)) = \exp(-\psi_Z^*(t)) \qquad \text{(use (4.11))} \qquad \text{QED}
\end{aligned}$$

For $\lambda = 0$ we have $\psi_Z(0) := \log \mathrm{E}e^{0Z} = \log 1 = 0$ and hence (as we can always opt for $\lambda = 0$ if there is no better λ)

$$\psi_Z^*(t) = \sup_{\lambda \geq 0}(\lambda t - \psi_Z(\lambda)) \geq 0 \tag{4.32}$$

By Jensen's inequality (4.13) we have $\psi_Z(\lambda) := \log \mathrm{E}e^{\lambda Z}$. Therefore for **negative** $\lambda < 0$ we get for all $t \geq \mathrm{E}Z$

$$\begin{aligned}
t &\geq \mathrm{E}Z \\
\lambda t &\leq \lambda \mathrm{E}Z \qquad (\lambda < 0) \\
\lambda t &\leq \psi_Z(\lambda) \qquad (\log \mathrm{E}e^{\lambda Z} \geq \mathrm{E}\log e^{\lambda Z} = \lambda \mathrm{E}Z) \\
\lambda t - \psi_Z(\lambda) &\leq 0
\end{aligned} \tag{4.33}$$

For all $t \geq \mathrm{E}Z$ we can therefore extend the supremum in (4.30) over $\lambda \in \mathbb{R}$ because none of the $\lambda < 0$ will be considered due to (4.32).

$$\psi_Z^*(t) = \sup_{\lambda \in \mathbb{R}}(\lambda t - \psi_Z(\lambda)) \tag{4.34}$$

which is known as the *Fenchel-Legendre transform* or as the *convex conjugate* of $\psi_Z(\lambda)$.

Chernoff's inequality is trivial if $\psi_Z^*(t) = 0$ (simply $\mathrm{P}(Z \geq t) \leq 1$). This happens whenever $\psi_Z(\lambda) = \infty$ for all positive $\lambda > 0$ or if $t \leq \mathrm{E}Z$.

Proof: If $t \leq \mathrm{E}Z$ and $\lambda \geq 0$ then

$$\begin{aligned}
t &\leq \mathrm{E}Z \\
\lambda t &\leq \lambda \mathrm{E}Z \leq \psi_Z(\lambda) \qquad (\lambda > 0) \\
\lambda t - \psi_Z(\lambda) &\leq 0 \qquad \text{QED}
\end{aligned}$$

To avoid the trivial situation, we assume that there exists $\lambda > 0$ such that $\mathrm{E}e^{\lambda Z} \leq \infty$. Denote by b the supremum of the interval of such λ so that $0 < b < \infty$. Then $\psi_Z(\lambda)$ is convex and infinitely many times differentiable on $I = (0, b)$ (strictly convex if Z is not almost surely constant).

Proof: of convexity using convex function definition (4.12) and Holder's inequality (4.4) with $X = e^{t\lambda_1 Z}$, $Y = e^{(1-t)\lambda_2 Z}$ and $p = 1/t$ and $q = 1/(1-t)$.

$$\begin{aligned} \mathrm{E}e^{t\lambda_1 Z} e^{(1-t)\lambda_2 Z} &\leq \left(\mathrm{E}e^{\lambda_1 Z}\right)^t \left(\mathrm{E}e^{\lambda_2 Z}\right)^{1-t} \\ \log \mathrm{E}e^{[t\lambda_1+(1-t)\lambda_2]Z} &\leq t \log \mathrm{E}e^{\lambda_1 Z} + (1-t)\log \mathrm{E}e^{\lambda_2 Z} \qquad \text{QED} \end{aligned}$$

The differentiability means that the Cramer transform $\psi^*_Z(t)$ can be obtained by differentiating$\lambda t - \psi_Z(\lambda)$ with respect to λ to get

$$\psi^*_Z(t) = \lambda_t t - \psi_Z(\lambda_t) \ , \tag{4.35}$$

where λ_t is such that $t = \psi'_Z(\lambda_t)$.

Because $\psi_Z(\lambda)$ is convex the derivative ψ' and its inverse $(\psi')^{-1}$ are increasing. We get $\lambda_t = (\psi')^{-1}(t)$.

Example: For a **centred normal random variable** Z with variance σ^2 and the mgf $\mathrm{E}e^{\lambda Z} = e^{\sigma^2\lambda^2/2}$ we have $\psi_Z(\lambda) = \sigma^2\lambda^2/2$, $\psi'_Z(\lambda) = \sigma^2\lambda$ and therefore $\lambda_t = t/\sigma^2$.

For every $t > 0$,

$$\psi^*_Z(t) = \lambda_t t - \psi_Z(\lambda_t) = \frac{t^2}{\sigma^2} - \frac{\sigma^2 t^2}{2\sigma^4} = \frac{t^2}{2\sigma^2} \tag{4.36}$$

so that the Chernoff's inequality (4.31) in this case gives

$$\mathrm{P}(Z \geq t) \leq e^{-t^2/(2\sigma^2)} \tag{4.37}$$

Sums of independent random variables Though Chernoff's inequality may not be as sharp as the moment bounds (4.27), it is particularly convenient for sums of independent random variables.

For a **sum of independent** r.v. $Z = \sum_i^n X_i$ we have for the moment generating function

$$\mathrm{E}e^{\lambda Z} = \mathrm{E}e^{\lambda \sum_i^n X_i} = \mathrm{E}\prod_i^n e^{\lambda X_i} = \prod_i^n \mathrm{E}e^{\lambda X_i} \tag{4.38}$$

and therefore

$$\psi_Z(\lambda) = \log \mathrm{E}e^{\lambda Z} = \log \prod_i^n \mathrm{E}e^{\lambda X_i} = \sum_i^n \log \mathrm{E}e^{\lambda X_i} = n\psi_X(\lambda) \tag{4.39}$$

$$\psi^*_X(t) = \lambda_t t - \psi_X(\lambda_t) \ , \tag{4.40}$$

where λ_t is such that $t = \psi'_X(\lambda_t)$.

$$\psi^*_Z(t) = \lambda_t t - n\psi_X(\lambda_t) \ , \tag{4.41}$$

where λ_t is such that $t = n\ \psi'_X(\lambda_t)$ so that

$$\psi^*_Z(t) = n\psi^*_X\left(\frac{t}{n}\right) \tag{4.42}$$

4.6 Sub-Gaussian random variables

Many r.v. have tail probabilities decreasing at least as rapidly as Gaussian r.v.

Definition sub-Gaussian r.v.: A centered r.v. X is said to be sub-Gaussian with variance factor ν (bound on the variance of X) if its log mgf is

$$\psi_X(\lambda) \leq \frac{\lambda^2 \nu}{2} \tag{4.43}$$

We denote the collection of such r.v. by $\mathcal{G}(\nu)$.

In other words, X belongs to $\mathcal{G}(\nu)$ if its mgf is dominated by that of a centered Gaussian r.v. with variance ν. If independent r.v. $X_1,\dots,X_n$ are all sub-Gaussian $X_i \leq \mathcal{G}(\nu_i)$ then $\sum_i X_i \in \mathcal{G}\left(\sum_i^n \nu_i\right)$.

Other properties: From Chernoff's inequality, if X belongs to $\mathcal{G}(\nu)$ then for every $t > 0$ (see the Gaussian example)

$$\mathrm{P}(X > t) \vee \mathrm{P}(-X > t) \leq e^{-t^2/(2\nu)} \quad , \tag{4.44}$$

where $a \vee b$ denotes the maximum of a and b.

Theorem 4.1 *Let X be r.f. with* $\mathrm{E}X = 0$. *If for some $\nu > 0$*

$$\mathrm{P}(X > x) \vee \mathrm{P}(-X > x) \leq e^{-t^2/(2\nu)} \qquad \textit{for all } x > 0$$

then for every integer $q \geq 1$

$$\mathrm{E}X^{2q} \leq 2q!(2\nu)^q \leq q!(4\nu)^q \quad .$$

Conversely, if for some constant $C > 0$

$$\mathrm{E}X^{2q} \leq q!(C)^q$$

then $X \in \mathcal{G}(4C)$ (sub-Gaussian with $\nu = 4C$).

Proof: Wlg we may assume $\nu = 1$ because otherwise we can apply a simple transformation $X/\sqrt{v}$. With $q \geq 1$, the random variable $Y = X^2 q = |X|^2 q \geq 0$ is **nonnegative**.

Theorem 4.2 (Fubini's): *For a product probability space $(S \times T, \mathcal{S} \otimes \mathcal{T}, \mu \otimes \nu)$ and $f : S \times T \to \mathbb{R}$ measurable, the integral with respect to the product measure $(\mu \otimes \nu)$ is equivalent to the iterated integrals*

$$\int_{S \times T} f(x,y) d(\mu \otimes \nu)(x,y) = \int_S \int_T f(x,y) d\nu(y) d\mu(x) = \int_T \int_S f(x,y) d\mu(x) d\nu(y) \tag{4.45}$$

For nonnegative Y we have

$$\begin{aligned}
\int_0^\infty \mathrm{P}(Y \geq y) dy &= \int_0^\infty \int_y^\infty p(t) dt\, dy && \text{(definition of density)} \\
&= \int_0^\infty \int_0^y p(t) dy\, dt && \text{(swap integrals }, \infty > y > 0, \infty > t > y, \Rightarrow \infty > t > 0, t > 0 > 0) \\
&= \int_0^\infty [y\, p(t)]_0^t\, dt && \text{(integral of constant)} \\
&= \int_0^\infty t\, p(t)\, dt = \mathrm{E}Y
\end{aligned}$$

Therefore

$$\mathrm{E}X^{2q} = \int_0^\infty \mathrm{P}(|X|^{2q} \geq x) dx \qquad\qquad = \int_0^\infty \mathrm{P}(|X| \geq x^{1/(2q)}) dx$$

References

[1] Casella, G., Berger, R. L.: Statistical Inference. Duxbury. 2002

[2] Boucheron, StÃľphane, GÃąbor Lugosi, and Pascal Massart. Concentration Inequalities: A Nonasymptotic Theory of Independence. Oxford University Press, 2013.

[3] Kyle Siegrist, Random website, https://www.randomservices.org/random/

[4] Mohri, M., Rostamizadeh, A., & Talwalkar, A.: Foundations of Machine Learning. MIT Press (2012)

Index

Cauchy-Schwarz inequality, 17
Chebyshev’s inequality, 21
Chernoff’s inequality, 22
convex conjugate, 22
convex function, 19
covariance inequality, 17
Cramer transform, 22
Cramer-Chernoff method, 22
cross-entropy, 13
cross-entropy loss, 13

ELBO, 6
evidence lower bound, 6
expectation maximization, 7

Fenchel-Legendre transform, 22
Fubini’s theorem, 24

Holder’s inequality, 17

importance sampling, 5, 15
inverse image, 19

Jensen’s inequality, 19
Jensens’s inequality, 6

KullbackâĂŞLeibler divergence, 3

link function, 12
Liupanov’s inequality, 18
logistic regression, 12
logistic sigmoid, 12
logits, 12

Markov’s inequality, 20
max to min, 18
maximum likelihood, 3
Minkowski’s inequality, 18
moment bounds, 21
moment generating function, 22

odds, 12

preimage, 19

reparametrization trick, 9

score, 12, 13
sigmoid, 12
soft-max, 14
Sub-Gaussian random variables, 24

Triangle inequality, 17

union bound, 19

variable transformation, 19