AI Manifesto: a Non-TLDR Approach

Batuhan Karaca

June 2, 2024

Contents

1 Introduction

1.1 TLDR But Binding

This book aims to give intuition for some ML (machine learning) concepts that I thought of significance. I started writing it simply because of the fact that I was bored of reading TLDR textbooks without understanding anything at all. That is why, I tried to give information as concise and as clear as possible. This compound thingy is solely my research and notetaking work piled up over years. I am trying to teach the topic using mostly mathematical proofs, though I also provide examples and/or explanations. I hope this book will be helpful for people wanting a career in the field.

I am not claiming to have 100% accurate statements, though have I not seen any conflictions between my findings and the publications. I do not think any theory is perfect as well. They are good only because they just work in practice. Furthermore, I am not claiming this book to be a textbook material on its own. I only wanted to share (hopefully) helpful information for deep understanding of the objects using math, as every scientific finding has a mathematical background in my opinion. Nevertheless, you will see some beliefs of my own as I could not come up with a clear answer (encountering conflicting statements in publications to check thereafter), so I warped it with my thoughts (science huh). For example, the fact that dependence is a superset of correlation in part 3.1 is such a gibberish. However, every other piece starts to fit nicely on each other with this statement (mutual coherency for now). I hope you will come up with your own brilliant ideas utilizing this knowledge and sharing them for all. Public science and open source forever!

1.2 Conventions, Notations and Definitions

It is not mandatory, but recommended to have some familiarity with the concepts (e.g. through courses) in linear algebra, calculus and probability theory.

Unless stated otherwise, these math notations apply.

- Capital letters represent matrices (or hypermatrices in general) and random variables.
- Lowercase symbols denote vectors and scalars.
- In some contexts, 0 and 1 may indicate a vector or matrix only composed of these scalars.
- Vectors are column vectors by default.
- Row vectors are denoted as their transpose with a superscript capital T (i.e. a^T).
- Scalar (or a row/column vector) of a matrix is denoted with corresponding lowercase symbol.
- Pre-superscript on a symbol of a matrix denotes the shape of the matrix (i.e. ${}^{M\times N}A$ is a matrix A with shape $M \times N$).
- Subscript of a lowercase variable symbol (i.e. $a_{ijk...}$) denotes the index of that variable in its corresponding matrix with capital letter. Furthermore, colon between indices indicates the ordered array (vector/matrix etc.) of variables whose indices are bounded between upper and lower bounds ordered (i.e. $a^T_{i:j} = [a^T_i, a^T_{i+1}, ..., a^T_{j-1}]$)
- Subscript of an integral or sum symbol denoting a variable (i.e. $\sum_X$, $\int_X$) indicates that the operation spans the entire domain of that variable (i.e. X).
- Subscript of a probabilitiy distribution denotes conditional probability (i.e. $p_\theta(x) = p(x|\theta)$
- Subscript of an expected value symbol (i.e. $E_{p(X)}$) denotes the probability weight of the random variable given in its formula (i.e. $p(X)$).
- Aggregation operators such as min support indexing (i.e. $\min_k x_k = x_{\arg_k \min x_k}$)

1.2.1 Mean (Expected Value)

$$E[X] = \mu_X = \int_X p(X) X dX \tag{1.2.1}$$

1.2.2 Variance

$$Var(X) = \sigma_X = \int_X p(X)(X - \mu_X)^2 dX \tag{1.2.2}$$

1.2.3 Covariance

$$Cov(X, Y) = \sigma_{XY} = \int_X \int_Y p(X, Y)(X - \mu_X)(Y - \mu_Y) dX dY \tag{1.2.3}$$

1.2.4 (Shannon's) Entropy

$$H(p(X)) = -E_{p(X)}[\log p(X)] = -\int_X p(X)\log p(X)dX \tag{1.2.4}$$

1.2.5 Cross Entropy

$$H(p(X), q(X)) = -E_{p(X)}[\log q(X)] = -\int_X p(X)\log q(X)dX \tag{1.2.5}$$

1.2.6 KL Divergence

$$KL(p(X)\|q(X)) = E_{p(X)}[\log\frac{p(X)}{q(X)}] = \int_X p(X)\log\frac{p(X)}{q(X)}dX \tag{1.2.6}$$

1.2.7 P-Norm

$$\|a\|_p = (\sum_i a_i^p)^{\frac{1}{p}} \tag{1.2.7}$$

$$\|a\|_1 \qquad \text{(Manhattan norm)} \tag{1.2.8}$$

$$\|a\|_2 = \|a\| \qquad \text{(Euclidean norm)} \tag{1.2.9}$$

1.2.8 Identity Matrix

Identity matrix is the matrix whose diagonal is all 1s and denoted I. It is multiplicative identity element of the matrices (i.e. $AI = IA = A$)

1.2.9 Multi-variate Normal Distribution

The general equation below holds only when the covariance matrix Σ is positive definite, where k is the number of features/variables in vectors x and μ. Note when $k = 1$, x and μ become scalar.

$$\mathcal{N}(x;\mu,\Sigma) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^k|\Sigma|}} \tag{1.2.10}$$

The specific case for when the variance is constant (same for all features, i.e. $\Sigma = \sigma^2 I$)

$$\mathcal{N}(x;\mu,\sigma^2 I) = \frac{1}{\sigma^k\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 I} = \frac{1}{\sigma^k\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 I\right) \tag{1.2.11}$$

For 1.2.11, we will use the notation $\mathcal{N}(x;\mu,\sigma^2) = \mathcal{N}(x;\mu,\sigma^2 I)$, omitting the identity to simplify calculations. For standard normal distribution we also omit the variable term (i.e. $\mathcal{N}(0,1) = \mathcal{N}(x;0,I)$). Note that both 1.2.10 and 1.2.11 are Gaussian functions, so the terms Normal and Gaussian are used interchangeably sometimes.

2 Linear Algebra Used In Learning and Optimization

2.1 Cosine Law

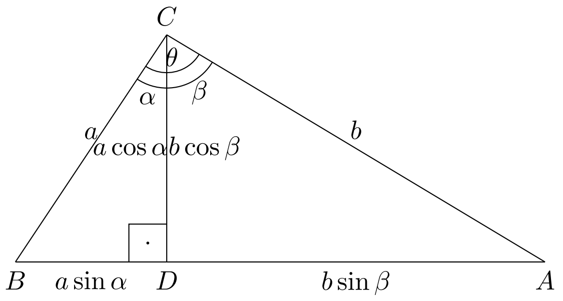

Figure 1:

Figure 1 shows a triangle with rules below

$$
\begin{aligned}
|BC| &= a \tag{2.1.1}\\
|AC| &= b \tag{2.1.2}\\
|BD| &= a\sin\alpha \tag{2.1.3}\\
|AD| &= b\sin\beta \tag{2.1.4}\\
|CD| &= a\cos\alpha = b\cos\beta \tag{2.1.5}\\
|AB| &= |BD| + |AD| \tag{2.1.6}\\
|CD|\perp|AB| & \tag{2.1.7}\\
\angle BCD &= \alpha \tag{2.1.8}\\
\angle DCA &= \beta \tag{2.1.9}\\
\angle BCA &= \theta = \angle BCD + \angle DCA \tag{2.1.10}
\end{aligned}
$$

$$
\begin{aligned}
c &= a\sin\alpha + b\sin\beta \tag{2.1.11}\\
c^2 &= a^2\sin^2\alpha + b^2\sin^2\beta + 2ab\sin\alpha\sin\beta \tag{2.1.12}\\
&= a^2\sin^2\alpha + b^2\sin^2\beta + 2ab[\cos\alpha\cos\beta - \cos(\alpha+\beta)] \text{ by trigonometric identities} \tag{2.1.13}\\
&= a^2\sin^2\alpha + b^2\sin^2\beta + ab\cos\alpha\cos\beta + ab\cos\alpha\cos\beta - 2ab\cos(\alpha+\beta) \tag{2.1.14}\\
&= a^2\sin^2\alpha + b^2\sin^2\beta + a^2\cos^2\alpha + b^2\cos^2\beta - 2ab\cos(\alpha+\beta) \tag{2.1.15}\\
&= a^2(\sin\alpha^2 + \cos\alpha^2) + b^2(\sin^2\beta + \cos^2\beta) - 2ab\cos(\alpha+\beta) \tag{2.1.16}\\
&= a^2 + b^2 - 2ab\cos(\alpha+\beta) \tag{2.1.17}\\
&= a^2 + b^2 - 2ab\cos\theta \tag{2.1.18}
\end{aligned}
$$

2.2 Relation Between Dot Product and Cosine

Assume

$$c = a - b \tag{2.2.1}$$

such that a, b and c are vectors. We define an operation (dot product) on vectors which has following properties

- commutativity (i.e. $a^Tb = b^Ta$)
- distributivity (i.e. $a^T(b+c) = a^Tb + a^Tc$))
- dot product of a vector with itself is equivalent to squared Euclidean norm (length) (i.e. $a^Ta = \|a\|^2$)

and many other properties of the scalar multiplication we know of (In fact, the well known dot product formula $a^Tb = \sum_i a_ib_i$ satisfies all of these properties). These properties in mind we find

$$
\begin{aligned}
c^2 &= (a-b)^T(a-b) \tag{2.2.2}\\
&= a^Ta + b^Tb - 2a^Tb \tag{2.2.3}\\
& \tag{2.2.4}
\end{aligned}
$$

Using 2.1.18 (note that the values shown there are scalar lengths)

$$
\begin{aligned}
a^Ta + b^Tb - 2a^Tb &= \|a\|^2 + \|b\|^2 - 2\|a\|\|b\|\cos\theta \tag{2.2.5}\\
a^Tb &= \|a\|\|b\|\cos\theta \tag{2.2.6}
\end{aligned}
$$

Now for a given point p and normal vector a, every point x making a and $(x-p)$ perpendicular to each other ($\theta = \frac{\pi}{2}$)

$$
\begin{aligned}
a^T(x-p) &= 0 \tag{2.2.7}\\
a^Tx - a^Tp &= 0 \tag{2.2.8}\\
a^Tx + b &= 0 \tag{2.2.9}
\end{aligned}
$$

indeed spans a hyperplane. Notice $b = -a^Tp$ is a constant scalar. We have proven the hyperplane equation as well

2.3 Jensen's Inequality

Continuing from our results in previous part, we extract the last feature of x with its corresponding coefficient, to bring an axis of reference to our analysis

$$a^T x + b = 0 \tag{2.3.1}$$

$$a_{1:N-1}^T x + a_N y + b = 0 \tag{2.3.2}$$

Then we bring up a hypersurface with

$$y = f(x) \tag{2.3.3}$$

which intersects the plane in a region Ω such that their intersecting region Ω_c is the boundary of Ω. f is either convex (i.e. a bowl placed on a table in 3D, y increasing toward the sky) or concave (i.e. the bowl upside down in 3D) within Ω. Within Ω_c, we can take any set of points as column vectors organized as a matrix

$$X = \begin{bmatrix} x_0 & x_1 & ... & x_M \end{bmatrix}$$

It is easy to see that X satisfies both 2.3.2 and 2.3.3. Let us multiply both sides with a vector of weights c whose Manhattan norm is 1 (sum of its items adds up to 1).

$$a_{1:N-1}^T X + (a_N y + b)1^T = 0 \tag{2.3.4}$$

$$a_{1:N-1}^T Xc + (a_N y + b)1^T c = 0 \tag{2.3.5}$$

$$a_{1:N-1}^T Xc + a_N y + b = 0 \tag{2.3.6}$$

Whenever only one element $c_i = 1$ (and trivially others are zero of course), we have

$$a_{1:N-1}^T x_i + a_N f(x_i) + b = 0 \tag{2.3.7}$$

$$a_{1:N-1}^T x_i c_i + a_N f(x_i) c_i + b c_i = 0 \tag{2.3.8}$$

$$\sum_i a_{1:N-1}^T x_i c_i + \sum_i a_N f(x_i) c_i + \sum_i b c_i = 0 \tag{2.3.9}$$

$$a_{1:N-1}^T \sum_i x_i c_i + a_N \sum_i f(x_i) c_i + b = 0 \tag{2.3.10}$$

$$\tag{2.3.11}$$

We have shown for a point $x_c = Xc$

$$y(Xc) = f(X)c = \sum_i c_i f(x_i) \tag{2.3.12}$$

$$\min_k x_{kj} \leq x_{ij} \leq \max_k x_{kj} \tag{2.3.13}$$

$$c_i \min_k x_{kj} \leq c_i x_{ij} \leq c_i \max_k x_{kj} \tag{2.3.14}$$

$$\sum_i c_i \min_k x_{kj} \leq \sum_i c_i x_{ij} \leq \sum_i c_i \max_k x_{kj} \tag{2.3.15}$$

$$\min_k x_{kj} \leq \sum_i c_i x_{ij} \leq \max_k x_{kj} \tag{2.3.16}$$

$$\tag{2.3.17}$$

We see that x_c lies within a subregion of Ω, hence by definition of convexity of surface,

$$f(Xc) = \begin{cases} \leq f(X)c & \text{if } f \text{ is convex} \\ \geq f(X)c & \text{if } f \text{ is concave} \end{cases} \tag{2.3.18}$$

2.4 Positive Semi-definiteness of a Matrix of the Form $X^T X$

Let A be a symmetric square matrix, and also $A = X^T X$. For any vector z,

$$z^T A z = z^T X^T X z = (Xz)^T (Xz) \geq 0 \tag{2.4.1}$$

Since squared norm is non-negative, $A = X^T X$ is positive semi-definite.

2.5 Existence of a (Trivially Unique) Orthonormal Basis of Eigenvectors for a Symmetric Diagonalizable Matrix

Let A be a diagonalizable symmetric square matrix. Let P be a square matrix such that it's ith column v_i is the ith eigenvector of A. Let Λ be the diagonal matrix whose ith diagonal λ_i is the eigenvalue corresponding to v_i. According to definition of eigenvalues/eigenvectors.

$$\begin{aligned}
AP &= P\Lambda && (2.5.1)\\
A &= P\Lambda P^{-1} && (2.5.2)\\
\Lambda &= P^{-1}AP && (2.5.3)\\
\lambda_i v_i &= Av_i && (2.5.4)\\
\lambda_i v_i^T &= v_i^T A^T && (2.5.5)\\
\lambda_i v_i^T v_j &= v_i^T A^T v_j && (2.5.6)\\
&= v_i^T A v_j && (2.5.7)\\
&= \lambda_j v_i^T v_j && (2.5.8)\\
(\lambda_i - \lambda_j) v_i^T v_j &= 0 && (2.5.9)
\end{aligned}$$

Since A is diagonalizable, $\lambda_i = \lambda_j$ if and only if $i = j$. Hence if $i \neq j$, $v_i^T v_j = 0$ and P is orthogonal, hence $PP^T = D$ such that

$$D_{ij} = \begin{cases} |v_i|^2 & i = j \\ 0 & \text{otherwise} \end{cases} \qquad (2.5.10)$$

We can have $P' = PD^{-\frac{1}{2}}$, and $\Lambda' = \Lambda D = D^{\frac{1}{2}}\Lambda D^{\frac{1}{2}}$. Then

$$A = P'\Lambda' P'^T \qquad (2.5.11)$$

such that P' is orthonormal. We are going to reference P' and Λ' as P and Λ respectively. We have shown that any symmetric diagonalizable matrix has orthonormal basis of eigenvectors.

2.6 Relation Between Definiteness and Sign of the Eigenvectors

$$\begin{aligned}
z^T A z &= z^T P \Lambda P^T z && (2.6.1)\\
&= (P^T z)^T \Lambda (P^T z) && (2.6.2)\\
&= \sum_i (P^T z)_i^2 \lambda_i && (2.6.3)\\
&= \sum_i z_i'^2 \lambda_i && (2.6.4)
\end{aligned}$$

The term is actually sum of eigenvalues weighted by values of z', which has the same norm as z due to the fact that P is orthonormal. For all z (hence z'), the weighted sum is non-negative if and only if A is semi-positive definite. Non-negativity in this case is guaranteed when all eigenvectors are non-negative. This approach also works for other types of definite matrices (we cannot say anything about indefinite matrices).

2.7 Relation Between Invertability and Definiteness

For any diagonalizable matrix A,

$$\begin{aligned}
det(A) &= det(P)det(\Lambda)det(P^{-1}) && (2.7.1)\\
&= det(P)det(\Lambda)\frac{1}{det(P)} && (2.7.2)\\
&= det(\Lambda) && (2.7.3)\\
&= \prod_i \lambda_i && (2.7.4)
\end{aligned}$$

We see that the semi-definite matrices are not invertible since they have at least one zero eigenvalue. We have shown that semi-definite matrices are not invertible and definite matrices are invertible.

2.8 Definiteness of Hessian Matrix to Determine Curvature Near Critical Points

2.8.1 Quadratic Function

Let $q(x)$ be a quadratic function with ${}^{d\times d}A, {}^{d\times 1}x, {}^{d\times 1}b$ and a scalar c

$$q(x) = x^TAx + x^Tb + c \tag{2.8.1}$$

of the form where A is symmetric.

$$\begin{aligned} \nabla q(x) &= (A + A^T)x + b && (2.8.2)\\ &= 2Ax + b && (2.8.3) \end{aligned}$$

The critical point $x^* = -\frac{1}{2}A^{-1}b$ where $\nabla q(x^*) = 0$.

$$\begin{aligned} q(x^* + \Delta x) &= q(x^*) + \Delta x^TA\Delta x + 2\Delta x^TAx^* + \Delta x^Tb && (2.8.4)\\ &= q(x^*) + \Delta x^TA\Delta x && (2.8.5) \end{aligned}$$

We can deduce from previous results,

- $\forall\Delta x[q(x^*) < q(x^* + \Delta x)]$ if A is positive definite (x^* is minimum, minimization problem)
- $\forall\Delta x[q(x^*) > q(x^* + \Delta x)]$ if A is negative definite (x^* is maximum, maximization problem)

If A is indefinite, then depending on Δx, $q(x^*)$ can be less or greater than $q(x^*+\Delta x)$, which means x^* is a saddle point. Since semi-definite matrices are not invertible we cannot use $x^* = -\frac{1}{2}A^{-1}b$. If A is semi-definite, for some x, we have $q(x) = x^Tb+c$ (a hyperplane with a $d+1$-dimensional normal having scalars of b and having 1 in the dimension of a reference axis, i.e. inclined-plane). If b is zero, then we have infinitely many critical points (a hyperplane with a normal along the reference axis, i.e. ground-plane).

Let d be an arbitrary vector.

$$\begin{aligned} \nabla q(x^* + Pd) &= 2A(Pd + x^*) + b && (2.8.6)\\ &= 2(P\Lambda P^T)(Pd + x^*) + b && (2.8.7)\\ &= 2P\Lambda P^TPd + 2Ax^* + b && (2.8.8)\\ &= 2P\Lambda d && (2.8.9)\\ &= \sum_i 2d_i\lambda_iv_i && (2.8.10)\\ \nabla q(x^* + Pd)^T\nabla q(x^* + Pd) &= (2P\Lambda d)^T2P\Lambda d && (2.8.11)\\ &= 4d^T\Lambda P^TP\Lambda d && (2.8.12)\\ &= 4d^T\Lambda^2d && (2.8.13)\\ &= \sum_i 4d_i^2\lambda_i^2 && (2.8.14)\\ &= \tilde{c}^2 \quad \text{For an arbitary scalar } \tilde{c} && (2.8.15)\\ \sum_i \frac{d_i^2}{\frac{\tilde{c}^2}{4\lambda_i^2}} &= 1 && (2.8.16) \end{aligned}$$

We have shown that contours (curves where gradient magnitudes/norms are equal) are ellipsoids.

$$\nabla q(x^* + \frac{\tilde{c}}{2\lambda_i}v_i) = \tilde{c}v_i \tag{2.8.17}$$

in a contour with multiplier $\tilde{c}$. The eigenvector v_i is a unit direction along a principal semi-axis of the ellipsoid. Note that in order to reach to a contour with multiplier $\tilde{c}$ from x^*, the magnitude/norm of the vector is $\frac{\tilde{c}}{2\lambda_i}$. As λ_i increases, the magnitude/norm decreases, and vice-versa.

For any function, $f(x)$ its second order Tailor expansion approximates it in the neighbor of x.

$$f(x + \Delta x) \sim f(x) + \Delta x^T\nabla f(x) + \Delta x^TH(x)\Delta x \tag{2.8.18}$$

where $H(x) = \nabla^2f(x)$ is Hessian of $f(x)$. Substituting A, b and c, this approximation can be used to determine the behavior of f near x in optimization algorithms such as gradient descent (where Δx is learning rate times the gradient). For example if $H(x)$ is semi-definite, this time we have a plateau, as plateau is a plane near x (ground or inclined).

2.8.2 Arbitrary Gaussian Function

Let $g(x) = he^{q(x)}$, with an arbitrary scalar h.

$$\nabla g(x) = \nabla q(x) g(x) \tag{2.8.19}$$

We see that the critical points of $g(x)$ are the same as of $q(x)$.

$$\begin{aligned} g(x^* + \Delta x) &= he^{q(x^* + \Delta x)} && (2.8.20) \\ &= he^{q(x^*) + \Delta x^T A \Delta x} && (2.8.21) \\ &= e^{\Delta x^T A \Delta x} g(x^*) && (2.8.22) \end{aligned}$$

Similarly deductions from part 2.8.1 can be made. Same as that part, If A is indefinite, then depending on Δx, $g(x^*)$ can be less or greater (saddle point). If A is semi-sefinite, for some infinitely many x, we have $g(x) = e^{x^T b + c}$ (inclined-plane). If b is zero, then we have infinitely many critical points (ground-plane). Otherwise, we have no critical points at all.

$$\begin{aligned} \nabla g(x^* + Pd) &= \nabla q(x^* + Pd) g(x^* + Pd) && (2.8.23) \\ &= (2P\Lambda d) e^{(Pd)^T A (Pd)} g(x^*) && (2.8.24) \\ &= (2P\Lambda d) e^{d^T P^T A P d} g(x^*) && (2.8.25) \\ &= (2P\Lambda d) e^{d^T \Lambda d} g(x^*) && (2.8.26) \\ \nabla g(x^* + Pd)^T \nabla g(x^* + Pd) &= (\sum_i 4d_i^2 \lambda_i^2) e^{2d^T \Lambda d} g(x^*)^2 && (2.8.27) \\ &= (\sum_i 4d_i^2 \lambda_i^2) e^{\sum_i 2d_i^2 \lambda_i} g(x^*)^2 && (2.8.28) \end{aligned}$$

$$(\sum_i 4d_i^2 \lambda_i^2) e^{\sum_i 2d_i^2 \lambda_i} g(x^*)^2 = \tilde{c}^2 \tag{2.8.29}$$

Note that after this point, I could not come with a rigorous proof, but used a computer. If we use density function of normal distribution which is a special case of $g(x)$ with $b = -2A\mu$ (x^* becomes μ), $c = \mu^T A \mu$, $h = \frac{1}{\sqrt{(2\pi)^n det(A)}} = \frac{1}{\sqrt{(2\pi)^n \prod_j \lambda_j}}$ (to have infinite integral equal 1 by definition of density functions), and inputting the values to a calculator (I used Desmos), the function behaves like an ellipsoid having eigenvectors along its principal semi-axes. please note that $A = -\frac{1}{2}\Sigma^{-1}$, where covariance matrix Σ needs to be positive definite (it can trivially be semi-positive definite in the limit for generalization). Therefore, A is negative definite, having all negative eigenvalues. As I increased the eigenvalues, the length of the principal semi-axis corresponding to the eigenvector has decreased. This analysis gives ellipsidicity information of the Gaussian that is fitted to a normally distributed sample.

3 Probability Theory Concepts Used In Learning

3.1 Distinction Between Correlation and Dependence

We have seen the relationship between eigenvectors/values and the covariance matrix. Assume that we fitted a continuous normal distribution to an elliptic sample of data in order to estimate the most probable regions of occurence, the properties of the continuous distribution will also approximate the properties of the sample. If we squish the sample more, the data will be better approximated with a more squished Gaussian. Remember from the previous parts along the axis of compression, the eigenvalue corresponding to that axis will become larger. At some point it will be the largest eigenvalue and the data will be approximated by a hyperplane with the normal being its eigenvector (1D hyperplane is a line). We say, such data has a *linear dependence*. However, dependence -relation between axes/dimensions/features in the data- is a more general concept and the covariance matrix only gives information about linear dependence. There can be infinitely many other types such as quadratic, cubic and so on. Correlation is a subset of dependence. When a number of random variables are independent of each other (i.e. mutual independence), they don't have any dependence whatsoever; hence they also become uncorrelated, but the reverse condition does not hold. In other words, independence implies uncorrelatedness but not vice-versa

3.2 Probabilistic Chain Rule Equations With Independent Random Variables

3.2.1 Conditional Mutual Independence

By definition, if

$$p(A|B, C) = p(A|C) \tag{3.2.1}$$

then A and B are conditionally independent given C. The relationship between joint and conditional probabilities is defined as

$$p(D|E)P(E) = p(D, E) \tag{3.2.2}$$

for any E, F. Then given 3.2.1 and 3.2.2

$$p(A|B, C)p(B|C)p(C) = p(A, B|C)p(C) \tag{3.2.3}$$
$$p(A|B, C)p(B|C) = p(A, B|C) \tag{3.2.4}$$
$$p(A|C)p(B|C) = p(A, B|C) \tag{3.2.5}$$

Now assume that for a vector of input x and vector output (label) y, for $i \neq j$, y_i, y_j are conditionally independent given x (1), and x_i, y_i are conditionally independent given x_j (2). You can think of the vector as the data and pairs (x, y) as points such that the point y_i is label of x_i.

$$p(y|x) = \prod_i p(y_i|x) \tag{3.2.6}$$
$$= \prod_i p(y_i|x_i) \tag{3.2.7}$$

If we use natural language, then we can say (as also proven above) y_i depends only on x_i. This is the foundational assumption that people build their learning models on. Note that the input x_i can be further divided into features x_{ij} that could be correlated among each other. Then we can further remove those features in the equation above, still having the same result $p(y|x)$.

People use dimension reduction algorithms such as PCA to find those correlated features for removal. PCA simply finds the line having eigenvector with the largest eigenvalue as its normal. Than merges the features affected by this eigenvector. One may iteratively drop the features until the desired dimension.

3.2.2 Markov Property and Markov Chain

Markov property is defined for a stochastic process when the future only depends on the present. Using the equation 3.2.1 along this definition gives

$$p(x_t|x_{t-1}, x_{t-2}, ..., x_0) = p(x_t|x_{t-1}) \tag{3.2.8}$$

Using the equation 3.2.2, we can recursively expand any joint probability as follows

$$p(x_{0:T}) = p(x_0, x_1, ..., x_T) \tag{3.2.9}$$
$$= p_\theta(x_0) \prod_{t=1}^{T} p(x_t|x_{t-1}, x_{t-2}, ..., x_0) \tag{3.2.10}$$
$$p(x_{1:T}|x_0) = \prod_{t=1}^{T} p(x_t|x_{t-1}, x_{t-2}, ..., x_0) \tag{3.2.11}$$

If we further assume the process involving p has markov property, then

$$p(x_{0:T}) = p_\theta(x_0) \prod_{t=1}^{T} p(x_t|x_{t-1}) \tag{3.2.12}$$
$$p(x_{1:T}|x_0) = \prod_{t=1}^{T} p(x_t|x_{t-1}) \tag{3.2.13}$$

3.3 Cross-Entropy

When training with given parameters θ and input x, learning models choose the combination of parameters that yields the maximum output y_{max}, the process known as the maximum likelihood estimation. We come up with the most generalized discrete generalized distribution, categorical distribution. It is actually generalized Bernoulli distribution including non-binary random variables. Mathematically

$$p(y_i) = \prod_j p(y_i = j)^{1\{y_i=j\}} \tag{3.3.1}$$

$$1\{y_i = j\} = \begin{cases} 1 & y_i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.3.2}$$

$$\begin{aligned}\hat{\theta}_{MLE} &= \arg_\theta \max p(y|x,\theta) && (3.3.3)\\ &= \arg_\theta \max \prod_i p(y_i|x_i,\theta) && (3.3.4)\\ &= \arg_\theta \max \prod_i \prod_j p(y_i = j|x_i,\theta)^{1\{y_i=j\}} && (3.3.5)\\ -\log \hat{\theta}_{MLE} &= \arg_\theta \min \sum_i \sum_j -\log p(y_i = j|x_i,\theta)^{1\{y_i=j\}} && (3.3.6)\\ &= \arg_\theta \min \sum_i \sum_j -(1\{y_i = j\}) \log p(y_i = j|x_i,\theta) && (3.3.7)\end{aligned}$$

3.4 KL-Divergence

We assume there is an underlying data distribution that we try to approximate with a model distribution. Continuing from 3.3.7, substituting dummy probability distributions gives

$$\begin{aligned}-\log \hat{\theta}_{MLE} &= \arg_\theta \min \sum_i \sum_j -p_{data}(y_i = j|x_i) \log p_{model}(y_i = j|x_i,\theta) && (3.4.1)\\ &= \arg_\theta \min[H(p_{data}(y|x), p_{model}(y|x,\theta))] && (3.4.2)\\ &= \arg_\theta \min[\sum_i \sum_j p_{data}(y_i = j|x_i) \log \frac{p_{data}(y_i = j|x_i)}{p_{model}(y_i = j|x_i,\theta)} + \sum_i \sum_j -p_{data}(y_i = j|x_i) \log p_{data}(y_i = j|x_i)] && (3.4.3)\\ &= \arg_\theta \min[KL(p_{data}(y_i = j|x_i)\|p_{model}(y_i = j|x_i,\theta)) + H(p_{data}(y_i = j|x_i))] && (3.4.4)\\ &= \arg_\theta \min[KL(p_{data}(y_i = j|x_i)\|p_{model}(y_i = j|x_i,\theta))] && (3.4.5)\end{aligned}$$

Shannon's entropy term can be omitted as it does not depend on parameters. We have shown that maximizing likelihood will minimize the cross-entropy and KL-divergence between data and model distributions. In this regard, the model distribution tries to approximate the data distribution when training.

In very specific case of supervised classification tasks, it is assumed that j denotes the class of the variable. This general approach can be applied to any type of model as we will see some other(s) along the way.

3.5 Non-negativity of KL-Divergence

Negative of any function switches its convexity property (i.e. reversing the bowl on the table). Using the definition of KL-Divergence, the Jensen's inequality and the fact that logarithm is a concave function

$$\begin{aligned}\int_X p(X) \log \frac{p(X)}{q(X)} dX &= \int_X p(X)(-\log) \frac{q(X)}{p(X)} dX && (3.5.1)\\ &\geq -\log \int_X p(X) \frac{q(X)}{p(X)} dX && (3.5.2)\\ &\geq -\log \int_X q(X) dX && (3.5.3)\\ &\geq -\log \int_X q(X) dX && (3.5.4)\\ &\geq -\log 1 && (3.5.5)\\ &\geq 0 && (3.5.6)\end{aligned}$$

3.6 Mean Squared Error

We again use maximum likelihood. However, this time we assume y_i is a real number with noise $\epsilon_i \sim \mathcal{N}(0,1)$ around the real output intended (by the nature :D).

$$\begin{align}
y_i &= f(x_i,\theta) + \epsilon_i \tag{3.6.1}\\
\hat{\theta}_{MLE} &= \arg_\theta \max \prod_i p(y_i|x_i,\theta) \tag{3.6.2}\\
&= \arg_\theta \max \prod_i \frac{1}{2\pi} e^{-\frac{\epsilon_i^2}{2}} \tag{3.6.3}\\
&= \arg_\theta \max \prod_i \frac{1}{2\pi} e^{-\frac{1}{2}(f(x_i,\theta)-y_i)^2} \tag{3.6.4}\\
-\log \hat{\theta}_{MLE} &= \arg_\theta \min[\sum_i \log 2\pi + \sum_i \frac{1}{2}(f(x_i,\theta)-y_i)^2] \tag{3.6.5}\\
&= \arg_\theta \min \sum_i \frac{1}{2}(f(x_i,\theta)-y_i)^2 \tag{3.6.6}
\end{align}$$

In this case, maximizing likelihood will minimize the mean-squared error.

3.7 Preference of Mean over Summation in Loss Functions

Deriving the formulae 3.3.7 and 3.6.6, we see a pattern in which we try to minimize a loss function.

$$\hat{\theta}_{MLE} = \arg_\theta \min \sum_i \mathcal{L}(\hat{y_i}, y_i) \tag{3.7.1}$$

It is easy to prove that the mean (expected value) has the super-position property

$$\begin{align}
\mu_{aX+bY} &= \int_X \int_Y p(X,Y)(aX+bY)dXdY \tag{3.7.2}\\
&= a\int_X (\int_Y p(X,Y)dY)X)dX + b\int_Y (\int_X p(X,Y)dX)Y)dY \tag{3.7.3}\\
&= a\int_X p(X)XdX + b\int_Y p(Y)YdY \tag{3.7.4}\\
&= a\mu_X + b\mu_Y \tag{3.7.5}
\end{align}$$

$$\begin{align}
Cov(aX+bY, cZ+dT) &= \int_X \int_Y \int_Z \int_T p(X,Y,Z,T)(aX+bY-\mu_{aX+bY})(cZ+dT-\mu_{cZ+dT})dXdYdZdT \tag{3.7.6}\\
&= \int_X \int_Y \int_Z \int_T p(X,Y,Z,T)(a(X-\mu_X)+b(Y-\mu_Y))(c(Z-\mu_Z)+d(T-\mu_T))dXdYdZdT \tag{3.7.7}\\
&= \int_X \int_Y \int_Z \int_T [p(X,Y,Z,T)ac(X-\mu_X)(Z-\mu_Z) + ad(X-\mu_X)(T-\mu_T) + \nonumber\\
&\qquad bc(Y-\mu_Y)(Z-\mu_Z) + bd(Y-\mu_Y)(T-\mu_T)]dXdYdZdT \tag{3.7.8}\\
&= ac\int_X \int_Z [\int_Y \int_T p(X,Y,Z,T)dYdT](X-\mu_X)(Z-\mu_Z)dXdZ + \nonumber\\
&\quad ad\int_X \int_T [\int_Y \int_Z p(X,Y,Z,T)dYdZ](X-\mu_X)(T-\mu_T)dXdT + \nonumber\\
&\quad bc\int_Y \int_Z [\int_X \int_T p(X,Y,Z,T)dXdT](Y-\mu_Y)(Z-\mu_Z)dYdZ + \nonumber\\
&\quad bd\int_Y \int_T [\int_X \int_Z p(X,Y,Z,T)dXdZ](Y-\mu_Y)(T-\mu_T)dYdT \tag{3.7.9}
\end{align}$$

$$\begin{aligned} &= ac\int_X\int_Z p(X,Z)(X-\mu_X)(Z-\mu_Z)dXdZ+ \\ &ad\int_X\int_T p(X,T)(X-\mu_X)(T-\mu_T)dXdT+ \\ &bc\int_Y\int_Z p(Y,Z)(Y-\mu_Y)(Z-\mu_Z)dYdZ+ \\ &bd\int_Y\int_T p(Y,T)(Y-\mu_Y)(T-\mu_T)dYdT \end{aligned} \tag{3.7.10}$$

$$= acCov(X,Z) + adCov(X,T) + bcCov(Y,Z) + bdCov(Y,T) \tag{3.7.11}$$

We assume the equation

$$Cov(\sum_{i=1}^{m-1} a_iX_i, \sum_{i=1}^{n-1} b_iY_i) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} a_ib_jCov(X_i,Y_j) \tag{3.7.12}$$

holds.

$$Cov(\sum_{i=1}^{m} a_iX_i, \sum_{i=1}^{n} b_iY_i) = Cov(\sum_{i=1}^{m-1} a_iX_i + a_mX_m, \sum_{i=1}^{n-1} b_iY_i + b_nY_n) \tag{3.7.13}$$

$$= Cov(\sum_{i=1}^{m-1} a_iX_i, \sum_{i=1}^{n-1} b_iY_i) + b_nCov(\sum_{i=1}^{m-1} a_iX_i, Y_n) + a_mCov(X_m, \sum_{i=1}^{n-1} b_iY_i) + a_nb_mCov(X_m,Y_n) \tag{3.7.14}$$

$$= \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} a_ib_jCov(X_i,Y_j) + b_n\sum_{i=1}^{m-1} a_iCov(X_i,Y_n) + a_m\sum_{j=1}^{n-1} b_jCov(X_m,Y_j) + a_nb_mCov(X_m,Y_n) \tag{3.7.15}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} a_ib_jCov(X_i,Y_j) \tag{3.7.16}$$

If we assume all the values (including zero) that the coefficients can take, then using mathematical induction, we prove that the equation (3) holds.

$$Var(\sum_{i=1}^{n} a_iX_i) = Cov(\sum_{i=1}^{n} a_iX_i, \sum_{i=1}^{n} a_iX_i) \tag{3.7.17}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_ia_jCov(X_i,X_j) \tag{3.7.18}$$

$$= \sum_{i=1}^{n} a_i^2Cov(X_i,X_i) + \sum_{i=1}^{n}\sum_{j\neq i}^{n} a_ia_jCov(X_i,X_j) \tag{3.7.19}$$

$$= \sum_{i=1}^{n} a_i^2Var(X_i) + \sum_{i=1}^{n}\sum_{j\neq i}^{n} a_ia_jCov(X_i,X_j) \tag{3.7.20}$$

$$= \sum_{i=1}^{n} a_i^2Var(X_i) \quad \text{(Since we assumed } X \text{ is uncorrelated)} \tag{3.7.21}$$

Then for a sample with constant variance σ^2 the variance of the sample mean,

$$\begin{aligned}
Var(\overline{X}) &= Var(\frac{1}{n}\sum_{i=1}^{n} X_i) && (3.7.22)\\
&= \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) && (3.7.23)\\
&= \frac{1}{n^2}\sum_{i=1}^{n} \sigma^2 && (3.7.24)\\
&= \frac{1}{n^2} n\sigma^2 && (3.7.25)\\
&= \frac{\sigma^2}{n} && (3.7.26)\\
Std(\overline{X}) &= \sqrt{Var(\overline{X})} && (3.7.27)\\
&= \frac{\sigma}{\sqrt{n}} && (3.7.28)
\end{aligned}$$

We realize that as number of samples increases, the mean of the sample approximates the true mean of the distribution better. That is why people usually use mean instead of sum. Hence new expression for the loss function becomes

$$\hat{\theta}_{MLE} = \frac{1}{n} \arg_\theta \min \sum_{i=1}^{n} \mathcal{L}(\hat{y_i}, y_i) \tag{3.7.29}$$

As n is a constant relative to the parameters, we are still maximizing the likelihood.

3.8 Reparameterization Trick

Used in prominent network architectures such as the VAE (Variational Autoencoder) and the diffusion model this technique makes taking derivative of a sampling operation feasible and facilitates some calculations.

Assuming two variables being sampled

$$X \sim \mathcal{N}(x; \mu, \sigma^2) \text{ and } \epsilon \sim \mathcal{N}(0, 1) \tag{3.8.1}$$

Using 1.2.11

$$\mathcal{N}(x; \mu, \sigma^2) \propto \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \text{ and } \mathcal{N}(0, 1) \propto \exp\left(-\frac{1}{2}\epsilon^2\right) \tag{3.8.2}$$

Comparing both sides gives

$$x = \mu + \sigma\epsilon \tag{3.8.3}$$

The coefficients of the distributions are omitted as sampling operation is independent of them. Now it is easier to do some calculations on x such as derivatives that are required in backpropagation operations.

4 Gradients of Functions in Backpropagation

4.1 FC (Fully Connected) Layers

An FC layer implements the function on a set of matrices

$${}^{N\times M}Y = {}^{N\times D}X\, {}^{D\times M}W + {}^{N\times 1}1\, {}^{1\times M}b \tag{4.1.1}$$

Using the definition of matrix multiplication, we have

$$y_{nm} = \sum_{d=1}^{D} x_{nd} w_{dm} + b_m \tag{4.1.2}$$

Looking at this definition, we can find partial derivatives

$$\frac{\delta y_{nm}}{\delta x_{ij}} = \begin{cases} w_{jm} & if\ n = i \\ 0 & \text{otherwise} \end{cases} \tag{4.1.3}$$

$$\frac{\delta y_{nm}}{\delta w_{ij}} = \begin{cases} x_{ni} & if\ m = j \\ 0 & \text{otherwise} \end{cases} \tag{4.1.4}$$

$$\frac{\delta y_{nm}}{\delta b_j} = \begin{cases} 1 & if\ m = j \\ 0 & \text{otherwise} \end{cases} \tag{4.1.5}$$

Using the chain rule with a loss function gives

$$L_{x_{ij}} = \sum_{n=1}^{N} \sum_{m=1}^{M} L_{y_{nm}} \frac{\delta y_{nm}}{\delta x_{ij}} \tag{4.1.6}$$

$$= \sum_{m=1}^{M} L_{y_{nm}} w_{jm} \tag{4.1.7}$$

$${}^{N\times D}L_X = {}^{N\times M}L_Y\ {}^{M\times D}W^T \tag{4.1.8}$$

$$L_{w_{ij}} = \sum_{n=1}^{N} \sum_{m=1}^{M} L_{y_{nm}} \frac{\delta y_{nm}}{\delta w_{ij}} \tag{4.1.9}$$

$$= \sum_{n=1}^{N} L_{y_{nm}} x_{ni} \tag{4.1.10}$$

$${}^{D\times M}L_W = {}^{D\times N}X_T\ {}^{N\times M}L_Y \tag{4.1.11}$$

$$L_{b_j} = \sum_{n=1}^{N} \sum_{m=1}^{M} L_{y_{nm}} \frac{\delta y_{nm}}{\delta b_j} \tag{4.1.12}$$

$$= \sum_{n=1}^{N} L_{y_{nm}} \tag{4.1.13}$$

$${}^{1\times M}L_b = {}^{1\times N}1\ {}^{N\times M}L_Y \tag{4.1.14}$$

5 Diffusion Models

5.1 What is a Diffusion Model

Diffusion models require two processes that adds noise to data for a number of steps T, (*Sampling steps* parameter in AUTOMATIC1111/stable-diffusion-webui and alike). The first one, *forward (diffusion) process*, makes data noisier by adding Gaussian noise T times, whereas the second, *reverse process*, tries to recover the original image from the result by adding noise again T times.

5.2 Forward Process

Let *forward process* be defined as a Markov Chain (see 3.2.13):

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{5.2.1}$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t) \tag{5.2.2}$$

Using 3.8 and substituting $\alpha = 1 - \beta_t$ we have

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \tag{5.2.3}$$

where $\epsilon_\tau \sim \mathcal{N}(0,1)$ is sampled from the standard normal distribution. Then

$$x_t = \Big(\sqrt{\prod_{\tau=1}^{t} \alpha_\tau}\Big)x_0 + \sum_{\tau=1}^{t-1} \Big(\sqrt{\prod_{i=\tau+1}^{t} \alpha_i - \prod_{i=\tau}^{t} \alpha_i}\Big)\epsilon_\tau + \sqrt{1-\alpha_t}\epsilon_t \tag{5.2.4}$$

Let us prove this argument using mathematical induction. When $t = 1$, it is easy to see equations 5.2.3 and 5.2.4 are the same. For an arbitrary t

$$x_{t-1} = \Big(\sqrt{\prod_{\tau=1}^{t-1} \alpha_\tau}\Big)x_0 + \sum_{\tau=1}^{t-2}\Big(\sqrt{\prod_{i=\tau+1}^{t-1} \alpha_i - \prod_{i=\tau}^{t-1}\alpha_i}\Big)\epsilon_\tau + \sqrt{1-\alpha_{t-1}}\epsilon_{t-1} \tag{5.2.5}$$

$$\sqrt{\alpha_t}x_{t-1} = \Big(\sqrt{\prod_{\tau=1}^{t} \alpha_\tau}\Big)x_0 + \sum_{\tau=1}^{t-2}\Big(\sqrt{\prod_{i=\tau+1}^{t} \alpha_i - \prod_{i=\tau}^{t}\alpha_i}\Big)\epsilon_\tau + \sqrt{\alpha_t-\alpha_t\alpha_{t-1}}\epsilon_{t-1} \tag{5.2.6}$$

$$= \Big(\sqrt{\prod_{\tau=1}^{t} \alpha_\tau}\Big)x_0 + \sum_{\tau=1}^{t-1}\Big(\sqrt{\prod_{i=\tau+1}^{t} \alpha_i - \prod_{i=\tau}^{t}\alpha_i}\Big)\epsilon_\tau \tag{5.2.7}$$

$$\sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t = \Big(\sqrt{\prod_{\tau=1}^{t} \alpha_\tau}\Big)x_0 + \underbrace{\sum_{\tau=1}^{t-1}\Big(\sqrt{\prod_{i=\tau+1}^{t} \alpha_i - \prod_{i=\tau}^{t}\alpha_i}\Big)\epsilon_\tau + \sqrt{1-\alpha_t}\epsilon_t}_{\Sigma_t} = x_t \tag{5.2.8}$$

Therefore, by induction we have proven that equation 5.2.4 is true. We should have an equation of the form

$$x_t = \mu + \sigma \cdot \epsilon_t \tag{5.2.9}$$

We want to calculate Σ_t. The information in part 3.7 gives some clues on what to do next. Recursively expanding the general case using 3.7.5, we find

$$\mu_{\sum_{i=1}^N a_i X_i} = a_1\mu_{X_1} + \mu_{\sum_{i=2}^N a_i X_i} \tag{5.2.10}$$

$$= a_1\mu_{X_1} + a_2\mu_{X_2} + \mu_{\sum_{i=3}^N a_i X_i} \tag{5.2.11}$$

$$... \tag{5.2.12}$$

$$= \sum_{i=1}^N a_i \mu_{X_i} \tag{5.2.13}$$

We also found in the same part

$$Var(\sum_{i=1}^n a_i X_i) = \sum_{i=1}^n a_i^2 Var(X_i) \quad \text{(Assuming } X \text{ is uncorrelated)} \tag{5.2.14}$$

Narrowing down for the normal distribution gives

$$X_i \sim \mathcal{N}(x_i; \mu_i, \sigma_i^2) \ and \ Y = \sum_{i=1}^n c_i X_i \longleftrightarrow Y \sim N(y; \sum_{i=1}^n c_i\mu_i, \sum_{i=1}^n c_i^2\sigma_i^2) \tag{5.2.15}$$

ϵ_t is already given to be normally distributed. Substituting $X_i = \epsilon_i$ and the corresponding coefficients

$$\sum_{\tau=1}^{t-1}(\prod_{i=\tau+1}^{t} \alpha_i - \prod_{i=\tau}^{t}\alpha_i) + (1-\alpha_t) = \prod_{i=2}^{t}\alpha_i - \prod_{i=1}^{t}\alpha_i + \prod_{i=3}^{t}\alpha_i - \prod_{i=2}^{t}\alpha_i + ... + \prod_{i=t-1}^{t}\alpha_i - \prod_{i=t-2}^{t}\alpha_i + \prod_{i=t}^{t}\alpha_i - \prod_{i=t-1}^{t}\alpha_i + (1-\alpha_t) \tag{5.2.16}$$

$$= \prod_{i=2}^{t}\alpha_i - \prod_{i=1}^{t}\alpha_i + \prod_{i=3}^{t}\alpha_i - \prod_{i=2}^{t}\alpha_i + ... + \prod_{i=t-1}^{t}\alpha_i - \prod_{i=t-2}^{t}\alpha_i + \alpha_t - \prod_{i=t-1}^{t}\alpha_i + 1 - \alpha_t \tag{5.2.17}$$

Note that similar elements cancel each other and we are left with $1 - \prod_{i=1}^t \alpha_i$. Substituting $\prod_{\tau=1}^t \alpha_\tau = \tilde{\alpha}_t$.

$$x_t = \sqrt{\tilde{\alpha}_t}x_0 + \sqrt{1-\tilde{\alpha}_t}\epsilon_t \tag{5.2.18}$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\tilde{\alpha}_t}x_0, 1-\tilde{\alpha}_t) \tag{5.2.19}$$

5.3 Reverse Process and the Loss Function

We found a closed form solution for the forward process function. However, reverse process will be learnt by an ML (machine learning) algorithm. The authors decided to use a type of CNN (Convolutional Neural Network), U-Net.

Let *reverse process* be defined as another Markov Chain (backward in time, see 3.2.12):

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{5.3.1}$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{5.3.2}$$

Using the approach in 3.3,

$$\hat{\theta}_{MLE} = \arg_\theta \max p_\theta(x_0) = \arg_\theta \min \int_j -q(x_0 = j)\log p_\theta(x_0 = j)dj \tag{5.3.3}$$

Bear in mind that q is now the data distribution (generated by data) whereas p is the model distribution (generated by model). Our loss function becomes

$$\mathcal{L}(x_0) = H(q(x_0), p_\theta(x_0)) = E_{q(x_0)}[-\log p_\theta(x_0)] \tag{5.3.4}$$

which is intractable

$$\begin{aligned}
-\log p_\theta(x_0) &\leq -\log p_\theta(x_0) + KL(q(x_{1:T}|x_0)\|p_\theta(x_{1:T}|x_0)) \text{ due to 3.5} && (5.3.5)\\
&\leq -\log p_\theta(x_0) + E_{q(x_{1:T}|x_0)}[\frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)}] && (5.3.6)\\
&\leq -\log p_\theta(x_0) + E_{q(x_{1:T}|x_0)}[\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}] && (5.3.7)\\
&\leq -\log p_\theta(x_0) + E_{q(x_{1:T}|x_0)}[\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}] + \log p_\theta(x_0) && (5.3.8)\\
&\leq E_{q(x_{1:T}|x_0)}[\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}] && (5.3.9)\\
E_{q(x_0)}[-\log p_\theta(x_0)] &\leq E_{q(x_{0:T})}[\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}] \text{ (Integrated both sides)} && (5.3.10)\\
\arg_\theta \min E_{q(x_0)}[-\log p_\theta(x_0)] &\leq \arg_\theta \min E_{q(x_{0:T})}[\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}] && (5.3.11)
\end{aligned}$$

We found another loss function which is an upper bound; hence, minimizing this function will minimize the original. This function is tractable as shown below

$$\arg_\theta \min E_{q(x_{0:T})}[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}] = \arg_\theta \min E_{q(x_{0:T})}[\log \frac{\prod_{t=1}^{T} q(x_t|x_{t-1})}{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}] \tag{5.3.12}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[-\log p(x_T) + \sum_{t=1}^{T} \log \frac{q(x_t|x_{t-1})}{p_\theta(x_{t-1}|x_t)}] \tag{5.3.13}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[-\log p(x_T) + \sum_{t=2}^{T} \log \frac{q(x_t|x_{t-1})}{p_\theta(x_{t-1}|x_t)} + \log \frac{q(x_1|x_0)}{p_\theta(x_0|x_1)}] \tag{5.3.14}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[-\log p(x_T) + \sum_{t=2}^{T} \log(\frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} \cdot \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}) + \log \frac{q(x_1|x_0)}{p_\theta(x_0|x_1)}] \tag{5.3.15}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[-\log p(x_T) + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} + \sum_{t=2}^{T} \log \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)} + \log \frac{q(x_1|x_0)}{p_\theta(x_0|x_1)}] \tag{5.3.16}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[-\log p(x_T) + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} + \log \frac{q(x_T|x_0)}{q(x_1|x_0)} + \log \frac{q(x_1|x_0)}{p_\theta(x_0|x_1)}] \tag{5.3.17}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[\log \frac{q(x_T|x_0)}{p(x_T)} + \sum_{t=2}^{T} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} - \log p_\theta(x_0|x_1)] \tag{5.3.18}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[\underbrace{KL(q(x_T|x_0) \parallel p(x_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{KL(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \underbrace{- \log p_\theta(x_0|x_1)}_{L_0}] \tag{5.3.19}$$

$$= \arg_\theta \min E_{q(x_{0:T})}[\sum_{t=2}^{T} \underbrace{KL(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \underbrace{- \log p_\theta(x_0|x_1)}_{L_0}] \tag{5.3.20}$$

$$\tag{5.3.21}$$

We got rid of L_T term since it is constant with respect to parameters θ. As a side note, the authors assume $p(x_T) \sim \mathcal{N}(0, 1)$ and $\beta_T = 1$ making $L_T \sim 0$.

We conditioned on x_0 at 5.3.15 since $q(x_{t-1}|x_t, x_0)$ is tractable and the proof is as follows

$$q(x_{t-1}|x_t, x_0)q(x_t|x_0)q(x_0) = q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)q(x_0) \tag{5.3.22}$$

$$q(x_{t-1}|x_t, x_0)q(x_t|x_0) = q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0) \tag{5.3.23}$$

$$q(x_{t-1}|x_t, x_0) = q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \tag{5.3.24}$$

$$= q(x_t|x_{t-1})\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \text{ due to Markov property} \tag{5.3.25}$$

$$= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t)\frac{\mathcal{N}(x_{t-1}; \sqrt{\tilde{\alpha}_{t-1}}x_0, 1-\tilde{\alpha}_{t-1})}{\mathcal{N}(x_t; \sqrt{\tilde{\alpha}_t}x_0, 1-\tilde{\alpha}_t)} \tag{5.3.26}$$

5.3.25 is apparent in 5.3.15. Doing the crazy calculations gives

$$q(x_{t-1}|x_t, x_0) = q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \tag{5.3.27}$$

$$\propto \exp\Big(-\frac{1}{2}(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t})\Big) \tag{5.3.28}$$

$$= \exp\Big(-\frac{1}{2}(\frac{x_t^2 - 2\sqrt{\alpha_t}x_t x_{t-1} + \alpha_t x_{t-1}^2}{\beta_t} + \frac{x_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}x_0 x_{t-1} + \bar{\alpha}_{t-1}x_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t})\Big) \tag{5.3.29}$$

$$= \exp\Big(-\frac{1}{2}((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})x_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0)x_{t-1} + C(x_t, x_0))\Big) \tag{5.3.30}$$

From 1.2.11, the Normal distribution has the form

$$\mathcal{N}(x;\mu,\sigma^2) \propto \exp\left(-\frac{1}{2}(\frac{1}{\sigma^2}x^2 - \frac{2\mu}{\sigma^2}x + \frac{1}{\sigma^2}\mu^2)\right) \tag{5.3.31}$$

Substituting gives

$$\begin{aligned}
q(x_{t-1}|x_t,x_0) &= \mathcal{N}(x_{t-1};\tilde{\mu}_t(x_t,x_0),\tilde{\beta}_t) && (5.3.32)\\
\tilde{\beta}_t &= 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) && (5.3.33)\\
&= 1/(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}) && (5.3.34)\\
&= 1/(\frac{1-\bar{\alpha}_t}{\beta_t(1-\bar{\alpha}_{t-1})}) && (5.3.35)\\
&= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t && (5.3.36)\\
\tilde{\mu}_t(x_t,x_0) &= (\frac{\sqrt{\alpha_t}}{\beta_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t && (5.3.37)\\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 && (5.3.38)
\end{aligned}$$

Remembering 5.2.18, we have

$$x_0 = \frac{1}{\sqrt{\tilde{\alpha}_t}}(x_t - \sqrt{1-\tilde{\alpha}_t}\epsilon_t) \tag{5.3.39}$$

Substituting gives

$$\begin{aligned}
\tilde{\mu}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t) && (5.3.40)\\
&= \frac{\sqrt{\alpha_t\bar{\alpha}_t}(1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1-\bar{\alpha}_t}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}\epsilon_t && (5.3.41)\\
&= \frac{\sqrt{\bar{\alpha}_{t-1}}(\alpha_t - \bar{\alpha}_t) + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}x_t + \frac{\sqrt{1-\bar{\alpha}_t}(1-\alpha_t)}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\epsilon_t && (5.3.42)\\
&= \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}\epsilon_t && (5.3.43)\\
&= \frac{1-\bar{\alpha}_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}\epsilon_t && (5.3.44)\\
&= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) && (5.3.45)
\end{aligned}$$