Contents

1 Gaussian Discriminant Analaysis

When using this method, we assume each class comes from its own normal distribution. In other terms, each class C has mean μ_c and variance σ_c^2. Recall the PDF for normally distributed random variables:

$$X \sim N(\mu, \sigma^2) : P(X) = \frac{1}{(\sqrt{2\pi}\sigma)^d} e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}$$

Given a particular point x, the Bayes decision rule $r^*(x)$ will return the class C that maximizes the posterior probability $P(X = x|Y = C)\pi_c$, where π_c is the prior probability of drawing a sample from class c. This Bayes decision rule is not special for GDA, except that we model the posterior probability using a Gaussian.

In order to optimize the posterior probability, we want to take its derivative and find the critical points. In order to make this derivative easier, we take the logorithm of the likelihood function before attempting to derive it. Note that $ln(\omega)$ is monotomically increasing for $\omega > 0$, so maximizing $ln(\omega)$ is equivalent to maximizing ω. To further our goal of taking an easy derivative, let's also rewrite the posterior probability for each class in a slightly different form.

$$\begin{aligned} Q_C(x) =& ln((\sqrt{2\pi})^d P(X = x)\pi_C) \\ =& -\frac{\|x-\mu_c\|^2}{2\sigma_C^2} - dln(\sigma_C^2) + ln(\pi_C) \end{aligned}$$

Note that $P(X = x)$ in the above expression is shorthand for $P(X = x|Y = C)$, since we are talking about $Q_C(x)$. If we were to write the equation for $Q_D(x)$, it would mean $P(X = x|Y = D)$.

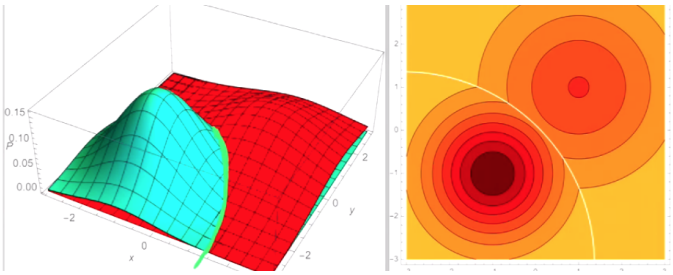
Our new function $Q_c(x)$ is useful, because it is quadratic in x (hence the name). Now that we have laid some groundwork, we can examine two methods of GDA. The first is called Quadratic Discriminant Analysis (QDA), and the second is called Linear Discriminant Analysis (LDA).

1.1 Quadratic Discriminant Analysis

Suppose there are only 2 classes C and D. The Bayes Optimal Decision Rule (BODR) $r^*(x)$ will be:

$$r^*(x) = \begin{cases} C & \text{if } Q_C(x) - Q_D(x) > 0 \\ D & \text{else} \end{cases}$$

We can think of $Q_C(x) - Q_D(x) > 0$ as the decision function, which is quadratic in x. The Bayes Optimal Decision Boundary (BODB) is the set of points where $\{x \mid Q_C(x) - Q_D(x) = 0\}$. In a single dimension, the BODB may have 0, 1, or 2 points. In d dimensions, the BODB will be a quadric (the higher-dimensional version of a conic section).



Sometimes, in addition to getting a prediction for the class of x, we may also want a probability that our prediction is correct.

$$P(Y = C|X) = \frac{P(X|Y = C)\pi_C}{P(X|Y = C)\pi_C + P(X|Y = D)\pi_D}$$

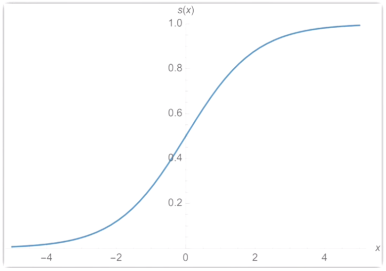
This calculation is no different from how we made our class prediction in the first place, because we had to calculate the probability of each class in order to choose the maximum probability. Right at the decision boundary, the probability of correctness will be at a minimum of $\frac{1}{2}$ (or for n intersecting

regions, $\frac{1}{n}$). Suppose we want to simplify the above expression. Recall that $e^{Q_C(x)} = ((\sqrt{2\pi})^d P(x)\pi_C$. We could rewrite the above expression as:

$$\begin{aligned} P(Y = C|X) =& \frac{e^{Q_c(x)}}{e^{Q_c(x)} + e^{Q_d(x)}} \\ =& \frac{1}{1 + e^{Q_D(x) - Q_C(x)}} \\ =& \frac{1}{1 + e^{-\gamma}} \qquad\qquad \text{where } \gamma = Q_C(x) - Q_D(x) \end{aligned}$$

In the above expression, the result is just some scalar function applied to the BODR. Actually, it happens to be a special and famous function called the logistic function or sigmoid function.

$$s(\gamma) = \frac{1}{1 + e^{-\gamma}}$$



In the image above, the horizontal axis is γ and the vertical axis is $s(\gamma)$. Essentially, the logistic function "squishes" the number line between zero and one. This behavior is useful for modeling probabilities, which fall within that range. Note that $s(0)$ is $\frac{1}{2}$, which is consistant with our interpretation of the probability of C and D both being $\frac{1}{2}$ at the decision boundary $\gamma = 0$. Also note that s is monotomically increasing. Therefore larger values of γ imply higher likelihood of being in class C.

1.2 Linear Discriminant Analysis

This method is very similar to QDA, but the decision boundaries generated by LDA will always be *linear*. This method is primarily useful to avoid overfitting that might occur in QDA. We make an extra assumption in LDA: all of the Gaussians have identical variance σ^2. This extra assumption greatly

simplifies our calculation. Now each class is fully identified by its mean μ_C and its prior π_C. If the priors are uniform, then the optimal decision rule simply chooses the class whose mean is closest to x.

$$\begin{aligned} Q_C(x) - Q_D(x) =& \frac{(\mu_C - \mu_D) \cdot x}{\sigma^2} - \frac{|\mu_C|^2 - |\mu_D|^2}{2\sigma^2} + ln(\pi_C) - ln(\pi_D) \\ =& w \cdot x + \alpha \end{aligned}$$

Where $w = \frac{\mu_C - \mu_d}{\sigma^2}$, and $\alpha = -\frac{|\mu_C|^2 - |\mu_D|^2}{2\sigma^2} + ln(\pi_C) - ln(\pi_D)$. Since we were able to write the decision rule in the standard form $w \cdot x + \alpha$, we have found a linear classifier.

Instead of looking at the sign of $Q_C(x) - Q_D(x)$, we can express the exact same linear classifier in a different way. We can choose the class C which maximizes the linear discriminant function (LDF).

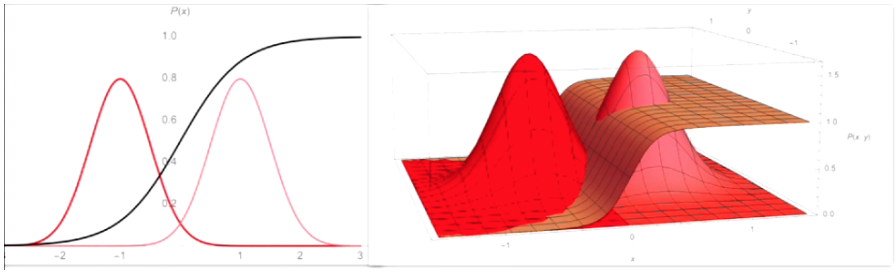
$$LDF_C(x) = \frac{\mu_C \cdot x}{\sigma^2} - \frac{|\mu_C|^2}{2\sigma^2} + ln(\pi_C)$$

If we subtract the analogous expression for class D from the one above, we get an expression that is equivalent to $Q_C(x) - Q_D(x)$. This slight variation on checking the sign of $Q_C(x) - Q_D(x)$ is useful, because we can easily generalize to several classes. We simply compute the LDF for each class, and our decision rule chooses the largest one for each x.

$$r^*(x) = \text{class } C \text{ which maximizes } LDF_C(x)$$

This rule is equivalent to computing all of the pairwise combinations of $Q_C(x) - Q_D(x)$ for each pair of classes, then running a simple algorithm to find the most likely class. Computing a single value for each class and taking the maximum is obviously a more efficient approach.

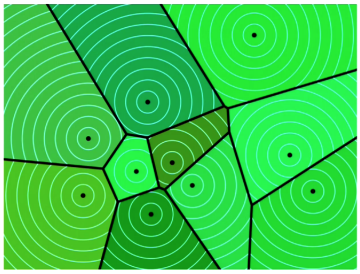
In the 2-class case, the decision boundary is $w \cdot x + \alpha = 0$ as always. Moreover, the Bayes posterior $P(Y = C|X = x)$ is equal to $s(w_C \cdot x + \alpha_D)$.



The curve in black (on the left) is the posterior probability, which is calculated by $\frac{\text{light curve}}{\text{light curve+dark curve}}$. Note that the posterior probability is a linearly transformed logistic function.

The curve in orange (on the right) is the posterior probability, calculated similarly to the previous case. Note that the posterior probability in this case is still a one-dimensional, linearly transformed logistic function. It has simply been extended through another dimension. In a higher-dimensional space, the posterior probability would behave the same way: extending "flatly" through each new dimension, remaining linear.

Below is an example of a multi-class LDA classifier, in the special case where each class has a uniform prior. As shown in the image below, the closest mean defines each region of classification. This classifier turns out to be equivalent to the centroid method, which seems naive yet works well for data under assumptions of Gaussian distributions, uniform variance, and uniform priors.



2 Maximum Likelihood Estimation

Consider an example problem where we are flipping a biased coin. After 10 flips, we observe 8 heads and 2 tails. What is the most likely value of p (the probability that the coin lands on heads for any given flip)?

We know that the number of heads that we observe follows a Binomial Distribution: $X \sim B(n, p)$.

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

In our example, we have

$$P(X = 8) = 45p^8(1-p)^2$$

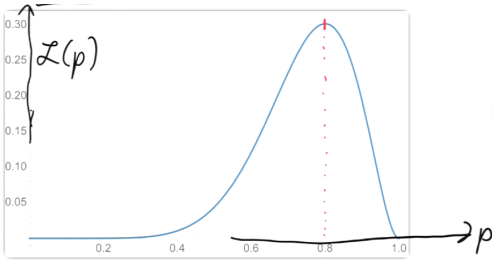
.

Call the above function $\mathcal{L}(p) = 45p^8(1-p)^2$ the likelihood function. Note that $\mathcal{L}$ is a function of distribution parameter(s).

The Maximum Likelihood Estimation (MLE) method allows us to estimate the parameters of a statistical model by picking the parameters which maximize $\mathcal{L}$. Note that MLE is one method among a larger group of density estimators, which are all methods of estimating a PDF.

For maximum formality, let's frame this as an optimization problem.

$$max_p\{\mathcal{L}(p)\}$$



To make sure we have the correct answer, let's use calculus and set the derivative equal $\frac{\delta\mathcal{L}}{\delta p}$ to zero.

$$\begin{aligned}
\frac{\delta\mathcal{L}}{\delta p} &= \frac{\delta}{\delta p} 45p^8(1-p)^2 \\
\frac{\delta}{\delta p} 45p^8(1-p)^2 &= 0 \\
4(1-p) - p &= 0 \\
p &= 0.8
\end{aligned}$$

Everything about the above coin example also applies to generic, prior probability problems. Say we had 10 samples and 8 of them were in class C. Then, for the same reasons as before, we know $\pi_C = 0.8$.

The coin example is simple, because we are estimating a single parameter (p). Let's look at a more interesting example, where we want to model a Gaussian distribution with multiple parameters (μ and σ^2).

2.1 MLE for Gaussians

Given sample points X_1, X_2, ..., X_n, find the best-fit Gaussian. In other words, find the Gaussian which would be *most likely* to produce X_1, X_2, ...,

X_n. This seems a bit strange at first, since the probability of producing that *exact* set of points will always be zero for any Gaussian; however, it turns out we are still able to perform meaningful calculations in this setting. For now, think of the problem statement as: "find the Gaussian mostly likely to produce points *similar* to X_1, X_2, ..., X_n". The probability of generating these n points is zero, but we will define the likelihood of generating these n points to be the product of their p values from the PDF.

$$\mathcal{L}(\mu, \sigma^2 | X_1, ..., X_n) = \prod_{i=1}^{n} X_i$$

Notice that L is a giant product, which can be very difficult to differentiate to find its maximum. Instead, we can differentiate the log likelihood. Since the logorithm is monotomically increasing, it will have the same maximum value. Define l to be the log likelihood: $l = ln(\mathcal{L})$.

$$\begin{aligned} l(\mu, \sigma^2 | X_1, ..., X_n) =& ln(\prod_{i=1}^{n} X_i) \\ =& \sum_{i=1}^{n} ln(P(X_i)) \\ =& \sum_{i=1}^{n} (-\frac{\|X_i - \mu\|^2}{2\sigma^2} - dln(\sqrt{2\pi}) - dln(\sigma)) \end{aligned}$$

We want to set $\nabla_\mu l = 0$ and $\frac{\delta l}{\delta \sigma} = 0$. The point which satisfies these equations will be a critical point of the log likelihood, allowing us to find its maximum with respect to μ and σ^2.

$$\begin{aligned} \nabla_\mu l =& \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma_2} \\ \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma_2} =& 0 \\ \hat{\mu} =& \frac{1}{n} \sum_{i=1}^{n} X_i \end{aligned}$$

Note that the value of $\hat{\mu}$ which maximizes l is simply the sample mean, which should not be surprising. Now, let's do the same thing for $\hat{\sigma^2}$.

$$\frac{\delta l}{\delta \sigma} = \sum_{i=1}^{n} \frac{\|X_i - \mu\|^2 - d\sigma^2}{\sigma^3}$$

$$\frac{\|X_i - \mu\|^2 - d\sigma^2}{\sigma^3} = 0$$

$$\hat{\sigma^2} = \frac{1}{dn} \sum_{i=1}^{n} \|X_i - \mu\|^2$$

We don't know μ in the above equation, so we substitute $\hat{\mu}$ instead. These calculations confirm something that may have seemed obvious. To estimate the mean and variance for an unknown Gaussian which describes class C, we use the mean and variance of the sample points from class C. Note that in the limit where the number of sample points goes to infinity, these values will converge on the correct mean and variance.

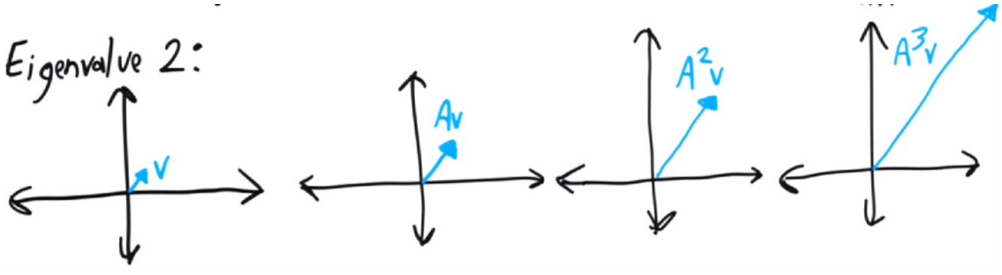
For QDA, we will estimate the conditional mean $\hat{\mu_C}$ and the conditional variance $\hat{\sigma_C^2}$ of each class C *separately*. We estimate the priors according to the "coin flip test" described at the beginning of this section. We call $\hat{\mu_C}$ and $\hat{\sigma_C^2}$ conditional, because they are conditioned on being in class C: $P(X = x|Y = C)$.

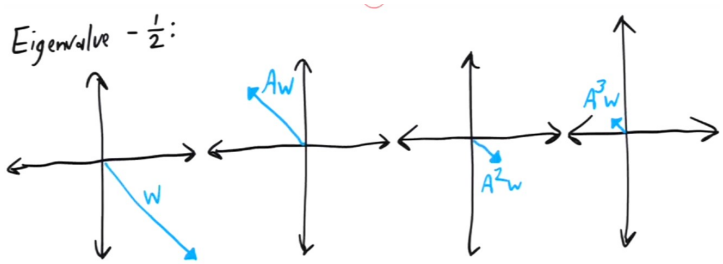
For LDA, we use the same methods as QDA for means and priors; however, we use a single variance (called the pooled variance) for all of the classes:

$$\hat{\sigma^2} = \frac{1}{dn} \sum_{C} \sum_{\{i|y_i=C\}} \|X_i - \hat{\mu_c}\|^2$$

3 Eigenvectors

Given square matrix A, if $Av = \lambda v$ for some vector $v \neq 0$ and some scalar λ, then v is an eigenvector of A, and λ is the eigenvalue of A associated with v.





THM1: If v and λ are an eigenvalue and eigenvector pair of A, then v is an eigenvector of A^k with eigenvector λ^k. The proof is simply: $A^2v = A(\lambda v) = \lambda^2 v$.

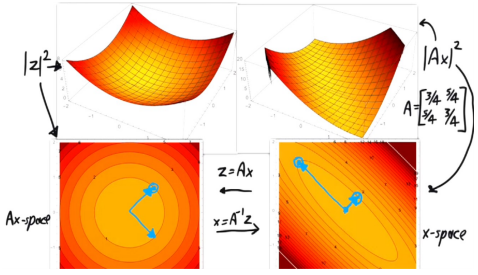
THM2: If A is invertible, then v is an eigenvector of A^{-1} with an eigenvalue of $\frac{1}{\lambda}$. The proof is simply: $A^{-1}v = \frac{1}{\lambda}A^{-1}Av = \frac{1}{\lambda}v$. Another view of the proof is shown below.

$$\begin{aligned} Av &= \lambda v \\ A^{-1}Av &= A^{-1}\lambda v \\ v &= A^{-1}\lambda v \\ \frac{1}{\lambda}v &= A^{-1}v \end{aligned}$$

Now, let's take a look at the Spectral Theorem, which states that every real, symmetric $n*n$ matrix has real eigenvalues and n eigenvectors which are all mutually orthogonal ($\forall i \neq j,\ v_i \cdot v_j = 0$). We can use these eigenvectors as a basis for R^n, since they are linearly independent.

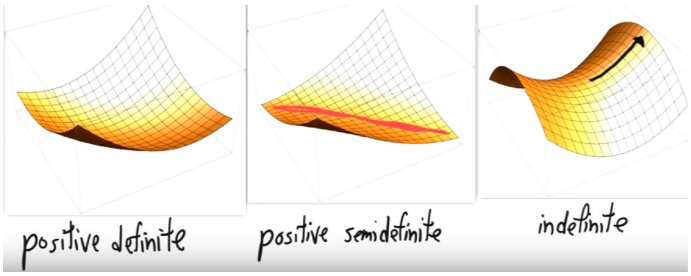
3.1 Quadratic Forms

Let's consider two functions: $\|z\|^2 = z \cdot z = z^T z$ and $\|Ax\|^2 = x \cdot (A^2 x) = x^T A^2 x$. The first function is quadratic, isotropic, and its isosurfaces are spheres. The second function is the quadratic form of the matrix A^2 (assuming A is symetric); it is anisotropic, and the isosurfaces are ellipsoids.



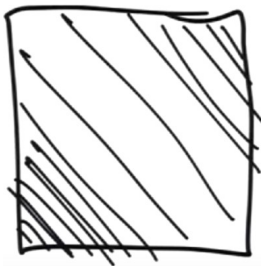
As we can see from the plots above, $\|Ax\|^2 = 1$ is an ellipsoid with axes $v_1, v_2, ..., v_n$ and radii $\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, ..., \frac{1}{\lambda_n}$. This is true, because if v_i has length $\frac{1}{\lambda_i}$, then $\|Av_i\|^2 = \|\lambda_i v_i\|^2 = 1 \implies v_i$ lies on the ellipsoid. One consequence of this interpretation is that bigger eigenvalue $\leftrightarrow$ steeper hill $\leftrightarrow$ shorter ellipsoid radius. In the special case where A is diagonal $\leftrightarrow$ eigenvectors are coordinate axis $\leftrightarrow$ ellipsoids are axis-aligned.

A square matrix B is positive definite if $w^T Bw > 0$ for all $w \neq 0 \leftrightarrow$ all eigenvalues are positive. It is positive semidefinite if $w^T Bw \geq 0$ for all $w \leftrightarrow$ all eigenvalues are non-negative. It is indefinite if it has at least one positive eigenvalue and at least one negative eigenvalue. It is invertible if it has no zero eigenvalues.



We want to know what this information tells us about the quadratic form $x^T A^2 x$. We know that A^2 is positive semi-definite (maybe positive definite) since all of the eigenvalues were squared.

Note that if our matrix has a zero eigenvalue, the radius of one of the ellipsoids in x space will go to infinity ($lim_{x \to 0} \frac{1}{x}$). Therefore, the isocontours of x space will form cylinders instead of ellipsoids. Another interpretation of a zero eigenvalue states that you have a direction of the transformation in which nothing changes. After projecting this missing direction out, the remaining directions (eigenvectors with non-zero eigenvalues) are still ellipsoids.



3.2 Building a Quadratic

When learning about eigenvalues and eigenvectors, it is common to learn how to find the eigenvalues and eigenvectors for a given matrix. However, it can often be more beneficial to learn how to work in the other direction: given a set of eigenvalues (assume orthogonality) and eigenvectors, how can we find a matrix that has them? In terms of the previous section, we know the ellipsoids, their directions, and their radii for the isocontours.

First, choose n mutually orthogonal unit vectors $v_1, ..., v_n$. Also define an $n * n$ matrix containing these unit vectors:

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

We know that $V^T V = I$, because the off-diagonal entries are orthogonal with zero-valued dot-products and the on-diagonal entries are unit vectors with one-valued dot-products. With this fact in mind, we can see that $V^T = V^{-1}$, since $V^{-1}V = 1$. One more useful permutation of these observations gives us $VV^T = 1$. Any matrix with these properties (summarized by $VV^T = 1$) is called an orthonormal matrix which acts like a rotation/reflection and does *not* change the length of the vectors it transforms.

Next, choose some inverse radii λ_i. Also define an $n*n$ matrix containing these inverse radii along the diagonal:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_n \end{bmatrix}$$

By the definition of eigenvectors, we can write:

$$\begin{aligned} AV &= V\Lambda \\ AVV^T &= V^T \\ A &= V\Lambda V^T && \text{because } V \text{ is orthonormal} \\ &= \sum_{i=1}^{n} \lambda_i v_i v_i^T && \text{where } v_i v_i^T \text{ is an outer product matrix with rank 1.} \end{aligned}$$

This method of rewriting A as the multiplication of other matricies is called matrix factorization, and this specific method is the eigendecomposition of A. You can think of Λ as the diagonalized version of A. You can also thing of V^T as the transformation that rotates the ellipsoid of isocontours in x space to be axis-aligned.

Observe that A^2 becomes $V\Lambda V^T V\Lambda V^T$ which is the same as $V\Lambda^2 V^T$. This quick proof shows that squaring a matrix keeps the same eigenvectors as the original matrix, but squares the eigenvalues (since squaring a diagonal matrix simply squares each of the values along the diagonal).

Given a symmetric Positive Semi-Definite (PSD) matrix M, we can find a symmetric square root $A = \sqrt{M}$. We simply take the square roots of M's eigenvalues, then construct A from M's eigenvectors and Λ with the square root of M's eigenvalues.

4 Anisotropic Gaussians

In the past, we have only considered Gaussians with the same variance in each direction. Now, we will take a look at Gaussians with different variances in different directions.

$$X \sim N(\mu, \Sigma)$$

$$P(X = x) = \frac{1}{(\sqrt{2\pi})^d \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Recall that X and μ are d dimensional vectors and Σ is a $d * d$ PSD matrix. Also note that $|\Sigma|$ refers to the determinant of Σ. We call Σ the covariance matrix, and its inverse Σ^{-1} is the equally important $d * d$ PSD precision matrix. This PDF can be a bit intimidating, so let's break it apart into sections.

$$P(X = x) = n(q(x))$$

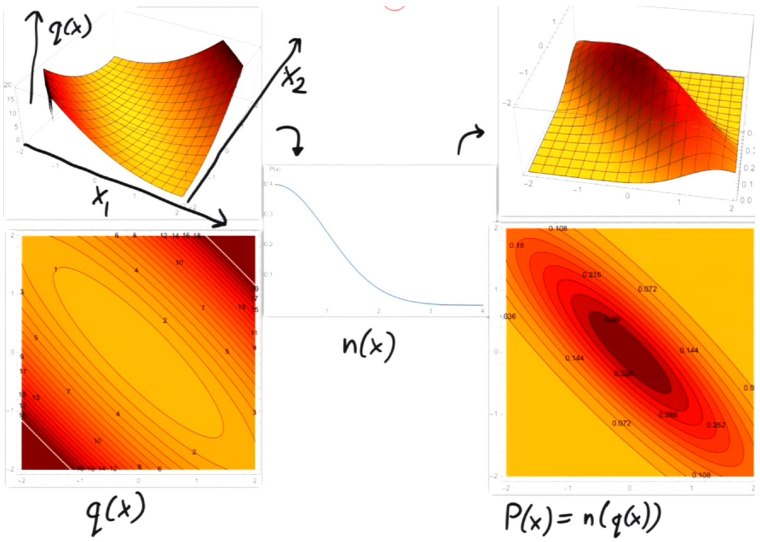
$$q(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$

$$n(x) = \text{univariate Gaussian PDF}$$

The quadratic function $q(x)$ maps $R^d \to R$. The exponential function n maps from $R \to R$. We can interpret $q(x)$ as the squared distance from $\frac{x}{\sqrt{\Sigma}}$ to $\frac{\mu}{\sqrt{\Sigma}}$. Consider the metric (a warped distance function) below:

$$
\begin{aligned}
d(x,\mu) =& |\frac{x}{\sqrt{\Sigma}} - \frac{\mu}{\sqrt{\Sigma}}| \\
=& \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \\
=& \sqrt{q(x)}
\end{aligned}
$$

There is a general principle that mapping reals to reals ($R \to R$), isosurfaces are maintained with potentially different isovalues and potentially some overlapping.



Define the covariance between random variables R and S (vectors or scalars) be

$$
\begin{aligned}
Cov(R,S) =& E[(R - E[R])(S - E[S])^T] \\
=& E[RS^T] - \mu_R \mu_S^T \\
Var(R) =& Cov(R,R)
\end{aligned}
$$

If R happens to be a vector, then $Cov(R,R)$ will be a matrix called the covariance matrix for R.

$$
Var(R) = \begin{bmatrix}
Var(R_1) & Cov(R_1,R_2) & \dots & Cov(R_1,R_d) \\
Cov(R_2,R_1) & Var(R_2) & \dots & Cov(R_2,R_d) \\
\vdots & \vdots & \ddots & \vdots \\
Cov(R_d,R_1) & Cov(R_d,R_2) & \dots & Var(R_d)
\end{bmatrix}
$$

For a Gaussian $R \sim N(\mu, \Sigma)$, one can prove that $Var(R) = \Sigma$. If R_i and R_j are independent $\implies Cov(R_i, R_j) = 0$. If R_i and R_j are two features of a multivariate normal distribution with $Cov(R_i, R_j) = 0 \implies R_i$ and R_j are independent. If all of the features are pairwise independent, then $Var(R)$ is a diagonal matrix. If $Var(R)$ is diagonal and R is a multivariate normal distribution, then you have an axis-aligned Gaussian, and the squared radii are on the diagonal of $\Sigma = Var(R)$, and $P(X = x) = \prod_{i=1}^{n} P(X_i)$.