What is the Covariance Matrix? (http://fouryears.eu/2016/11/23/what-is-the-covariance-matrix/)

Posted by Konstantin 23.11.2016

> *Basic linear algebra (http://fouryears.eu/wp-content/uploads/matrixalgebra-en.pdf) , introductory statistics and some familiarity with core machine learning concepts (such as PCA and linear models) are the prerequisites of this post. Otherwise it will probably make no sense. An abridged version of this text is also posted on Quora (https://www.quora.com/Principal-Component-Analysis-What-is-the-intuitive-meaning-of-a-covariance-matrix/answer/Konstantin-Tretyakov) .*

Most textbooks on statistics cover covariance (https://en.wikipedia.org/wiki/Covariance) right in their first chapters. It is defined as a useful "measure of dependency" between two random variables:

$$\mathrm{cov}(X,Y) = E[(X - E[X])(Y - E[Y])].$$

The textbook would usually provide some intuition on why it is defined as it is, prove a couple of properties, such as bilinearity, define the *covariance matrix* for multiple variables as $\mathbf{\Sigma}_{i,j} = \mathrm{cov}(X_i, X_j)$, and stop there. Later on the covariance matrix would pop up here and there in seeminly random ways. In one place you would have to take its inverse, in another - compute the eigenvectors, or multiply a vector by it, or do something else for no apparent reason apart from "that's the solution we came up with by solving an optimization task".

In reality, though, there are some very good and quite intuitive reasons for why the covariance matrix appears in various techniques in one or another way. This post aims to show that, illustrating some curious corners of linear algebra in the process.

Meet the Normal Distribution

The best way to truly understand the covariance matrix is to forget the textbook definitions completely and depart from a different point instead. Namely, from the the definition of the multivariate Gaussian distribution (https://en.wikipedia.org/wiki/Multivariate_normal_distribution) :

We say that the vector $\mathbf{x}$ has a *normal* (or *Gaussian*) distribution with mean μ and covariance $\mathbf{\Sigma}$ if:

$$\Pr(\mathbf{x}) = |2\pi\mathbf{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right).$$

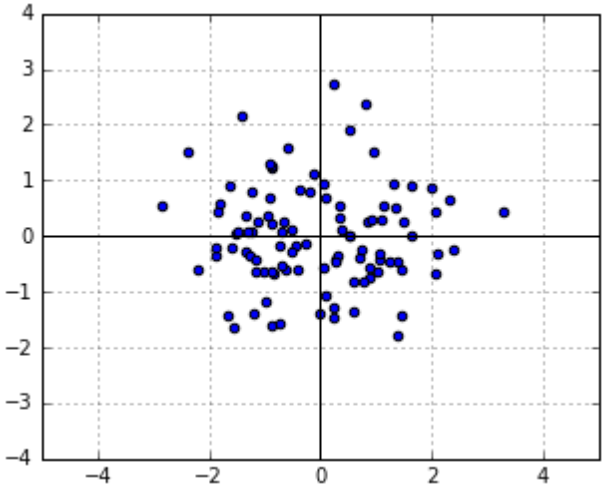
To simplify the math a bit, we will limit ourselves to the centered distribution (i.e. $\mu = \mathbf{0}$) and refrain from writing out the normalizing constant $|2\pi\mathbf{\Sigma}|^{-1/2}$. Now, the definition of the (centered) multivariate Gaussian looks as follows:

$$\Pr(\mathbf{x}) \propto \exp\left(-0.5\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}\right).$$

Much simpler, isn't it? Finally, let us define the covariance matrix as nothing else but *the parameter of the Gaussian distribution*. That's it. You will see where it will lead us in a moment.

Transforming the Symmetric Gaussian

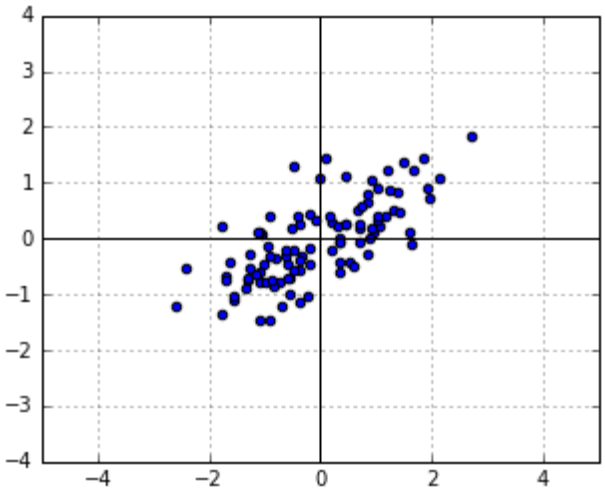
Consider a symmetric Gaussian distribution, i.e. the one with $\mathbf{\Sigma} = \mathbf{I}$ (the identity matrix). Let us take a sample from it, which will of course be a symmetric, round cloud of points:



We know from above that the likelihood of each point in this sample is

$$P(\mathbf{x}) \propto \exp(-0.5\mathbf{x}^T\mathbf{x}). \tag{1}$$

Now let us apply a linear transformation $\mathbf{A}$ to the points, i.e. let $\mathbf{y} = \mathbf{A}\mathbf{x}$. Suppose that, for the sake of this example, $\mathbf{A}$ scales the vertical axis by 0.5 and then rotates everything by 30 degrees. We will get the following new cloud of points $\mathbf{y}$:



What is the distribution of $\mathbf{y}$? Just substitute $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ into (1), to get:

$$\begin{aligned} P(\mathbf{y}) &\propto \exp(-0.5(\mathbf{A}^{-1}\mathbf{y})^T(\mathbf{A}^{-1}\mathbf{y})) \\ &= \exp(-0.5\mathbf{y}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y}) \end{aligned} \tag{2}$$

This is exactly the Gaussian distribution with covariance $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^T$. The logic works both ways: if we have a Gaussian distribution with covariance $\mathbf{\Sigma}$, we can regard it as a *distribution which was obtained by transforming the symmetric Gaussian by some* $\mathbf{A}$, and we are given $\mathbf{A}\mathbf{A}^T$.

More generally, if we have *any* data, then, when we compute its covariance to be $\mathbf{\Sigma}$, we can say that *if our data were Gaussian,* then *it could have been obtained* from a symmetric cloud using some transformation $\mathbf{A}$, and we just estimated the matrix $\mathbf{A}\mathbf{A}^T$, corresponding to this transformation.

Note that we do not know the actual $\mathbf{A}$, and it is mathematically totally fair. There can be many different transformations of the symmetric Gaussian which result in the same distribution shape. For example, if $\mathbf{A}$ is just a rotation by some angle, the transformation does not affect the shape of the distribution at all. Correspondingly, $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ for all rotation matrices. When we see a unit covariance matrix we really do not know, whether it is the "originally symmetric" distribution, or a "rotated symmetric distribution". And we should not really care - those two are identical.

There is a theorem in linear algebra, which says that any symmetric matrix $\mathbf{\Sigma}$ can be represented as:

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{3}$$

where $\mathbf{V}$ is orthogonal (i.e. a rotation) and $\mathbf{D}$ is diagonal (i.e. a coordinate-wise scaling). If we rewrite it slightly, we will get:

$$\mathbf{\Sigma} = (\mathbf{V}\mathbf{D}^{1/2})(\mathbf{V}\mathbf{D}^{1/2})^T = \mathbf{A}\mathbf{A}^T, \tag{4}$$

where $\mathbf{A} = \mathbf{V}\mathbf{D}^{1/2}$. This, in simple words, means that *any covariance matrix* $\mathbf{\Sigma}$ could have been the result of transforming the data using *a coordinate-wise scaling* $\mathbf{D}^{1/2}$ followed by *a rotation* $\mathbf{V}$. Just like in our example with $\mathbf{x}$ and $\mathbf{y}$ above.

Principal Component Analysis

Given the above intuition, PCA already becomes a very obvious technique. Suppose we are given some data. Let us *assume (or "pretend")* it came from a normal distribution, and let us ask the following questions:

1. What could have been the rotation $\mathbf{V}$ and scaling $\mathbf{D}^{1/2}$, which produced our data from a symmetric cloud?
2. What were the original, "symmetric-cloud" coordinates $\mathbf{x}$ before this transformation was applied.
3. Which original coordinates were scaled the most by $\mathbf{D}$ and thus contribute most to the spread of the data now. Can we only leave those and throw the rest out?

All of those questions can be answered in a straightforward manner if we just decompose $\mathbf{\Sigma}$ into $\mathbf{V}$ and $\mathbf{D}$ according to (3). But (3) is exactly the eigenvalue decomposition (https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix) of $\mathbf{\Sigma}$. I'll leave you to think for just a bit and you'll see how this observation lets you derive everything there is about PCA and more.

The Metric Tensor

Bear me for just a bit more. One way to summarize the observations above is to say that we can (and should) regard $\mathbf{\Sigma}^{-1}$ as a metric tensor (https://en.wikipedia.org/wiki/Metric_tensor) . A metric tensor is just a fancy formal name for a matrix, which summarizes the *deformation of space*. However, rather than claiming that it in some sense determines a particular transformation $\mathbf{A}$ (which it does not, as we saw), we shall say that it affects the way we compute *angles and distances* in our transformed space.

Namely, let us redefine, for any two vectors $\mathbf{v}$ and $\mathbf{w}$, their inner product (https://en.wikipedia.org/wiki/Inner_product_space) as:

$$\langle \mathbf{v}, \mathbf{w} \rangle_{\Sigma^{-1}} = \mathbf{v}^T \mathbf{\Sigma}^{-1} \mathbf{w}. \tag{5}$$

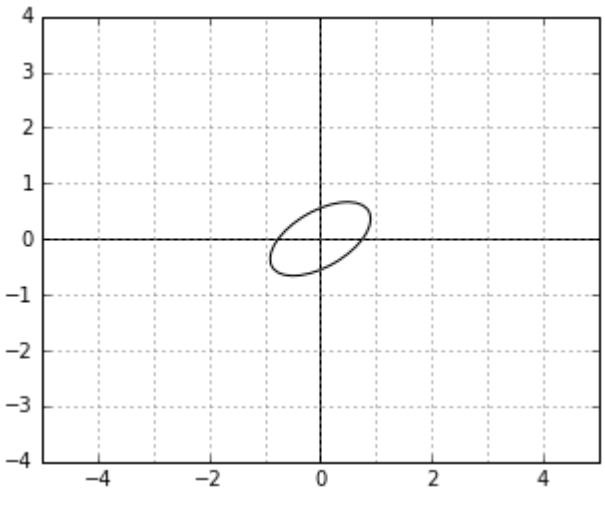
To stay consistent we will also need to redefine the *norm* of any vector as

$$|\mathbf{v}|_{\Sigma^{-1}} = \sqrt{\mathbf{v}^T \mathbf{\Sigma}^{-1} \mathbf{v}}, \tag{6}$$

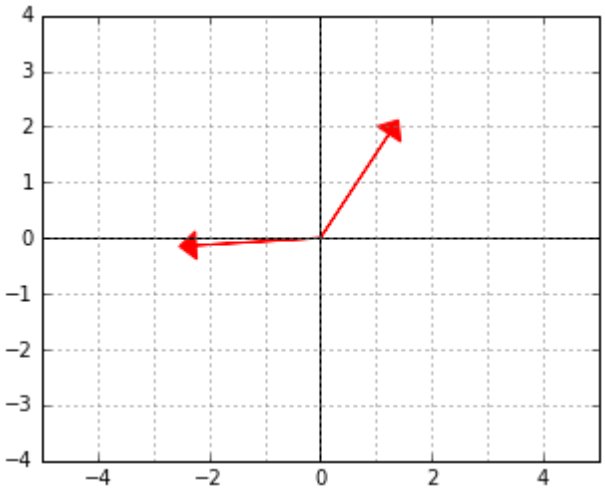
and the *distance* between any two vectors as

$$|\mathbf{v} - \mathbf{w}|_{\Sigma^{-1}} = \sqrt{(\mathbf{v} - \mathbf{w})^T \mathbf{\Sigma}^{-1} (\mathbf{v} - \mathbf{w})}. \tag{7}$$

Those definitions now describe a kind of a "skewed world" of points. For example, a unit circle (a set of points with "skewed distance" 1 to the center) in this world might look as follows:



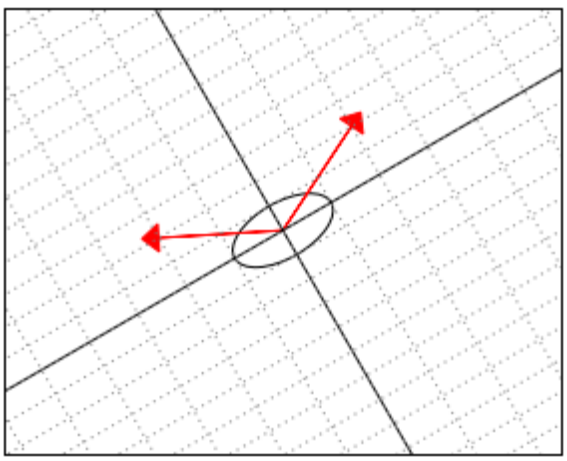
And here is an example of two vectors, which are considered “orthogonal”, a.k.a. “perpendicular” in this strange world:



Although it may look weird at first, note that the new inner product we defined is actually just the dot product of the “untransformed” originals of the vectors:

$$\mathbf{v}^T\mathbf{\Sigma}^{-1}\mathbf{w} = \mathbf{v}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{w} = (\mathbf{A}^{-1}\mathbf{v})^T(\mathbf{A}^{-1}\mathbf{w}), \tag{8}$$

The following illustration might shed light on what is actually happening in this Σ-“skewed” world. Somehow “deep down inside”, the ellipse thinks of itself as a circle and the two vectors behave as if they were (2,2) and (-2,2).



Getting back to our example with the transformed points, we could now say that the point-cloud $\mathbf{y}$ is actually a perfectly round and symmetric cloud “deep down inside”, it just happens to live in a *skewed space*. The deformation of this space is described by the tensor $\mathbf{\Sigma}^{-1}$ (which is, as we know, equal to $(\mathbf{A}\mathbf{A}^T)^{-1}$. The PCA now becomes a method for analyzing the *deformation of space*, how cool is that.

The Dual Space

We are not done yet. There's one interesting property of "skewed" spaces worth knowing about. Namely, the elements of their dual space (https://en.wikipedia.org/wiki/Dual_space) have a particular form. No worries, I'll explain in a second.

Let us forget the whole skewed space story for a moment, and get back to the usual inner product $\mathbf{w}^T\mathbf{v}$. Think of this inner product as a function $f_\mathbf{w}(\mathbf{v})$, which takes a vector $\mathbf{v}$ and maps it to a real number, the dot product of $\mathbf{v}$ and $\mathbf{w}$. Regard the $\mathbf{w}$ here as the *parameter ("weight vector")* of the function. If you have done any machine learning at all, you have certainly come across such *linear functionals* over and over, sometimes in disguise. Now, the set of *all possible linear functionals* $f_\mathbf{w}$ is known as the *dual space* to your "data space".

Note that each linear functional is determined uniquely by the parameter vector $\mathbf{w}$, which has the same dimensionality as $\mathbf{v}$, so apparently the dual space is in some sense equivalent to your data space - just the interpretation is different. An element $\mathbf{v}$ of your "data space" denotes, well, a data point. An element $\mathbf{w}$ of the dual space denotes a function $f_\mathbf{w}$, which *projects* your data points on the direction $\mathbf{w}$ (recall that if $\mathbf{w}$ is unit-length, $\mathbf{w}^T\mathbf{v}$ is exactly the length of the perpendicular projection of $\mathbf{v}$ upon the direction $\mathbf{w}$). So, in some sense, if $\mathbf{v}$-s are "vectors", $\mathbf{w}$-s are "directions, perpendicular to these vectors". Another way to understand the difference is to note that if, say, the elements of your data points numerically correspond to amounts in kilograms, the elements of $\mathbf{w}$ would have to correspond to "units per kilogram". Still with me?

Let us now get back to the skewed space. If $\mathbf{v}$ are elements of a skewed Euclidean space with the metric tensor $\mathbf{\Sigma}^{-1}$, is a function $f_\mathbf{w}(\mathbf{v}) = \mathbf{w}^T\mathbf{v}$ an element of a dual space? Yes, it is, because, after all, it is a linear functional. However, the *parameterization* of this function is inconvenient, because, due to the skewed tensor, we cannot interpret it as projecting vectors upon $\mathbf{w}$ nor can we say that $\mathbf{w}$ is an "orthogonal direction" (to a separating hyperplane of a classifier, for example). Because, remember, in the skewed space it is not true that orthogonal vectors satisfy $\mathbf{w}^T\mathbf{v} = 0$. Instead, they satisfy $\mathbf{w}^T\mathbf{\Sigma}^{-1}\mathbf{v} = 0$. Things would therefore look much better if we parameterized our dual space differently. Namely, by considering linear functionals of the form $f_\mathbf{z}^{\Sigma^{-1}}(\mathbf{v}) = \mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{v}$. The new parameters $\mathbf{z}$ could now indeed be interpreted as an "orthogonal direction" and things overall would make more sense.

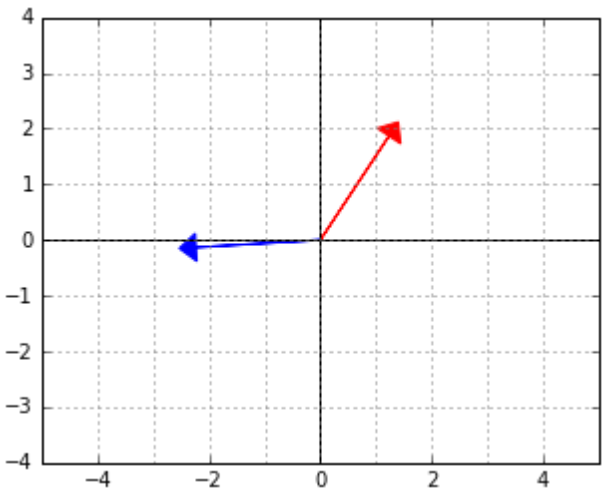
However when we work with actual machine learning models, we still prefer to have our functions in the simple form of a dot product, i.e. $f_\mathbf{w}$, without any ugly $\mathbf{\Sigma}$-s inside. What happens if we turn a "skewed space" linear functional from its natural representation into a simple inner product?

$$f_\mathbf{z}^{\Sigma^{-1}}(\mathbf{v}) = \mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{v} = (\mathbf{\Sigma}^{-1}\mathbf{z})^T\mathbf{v} = f_\mathbf{w}(\mathbf{v}), \tag{9}$$

where $\mathbf{w} = \mathbf{\Sigma}^{-1}\mathbf{z}$. (Note that we can lose the transpose because $\mathbf{\Sigma}$ is symmetric).

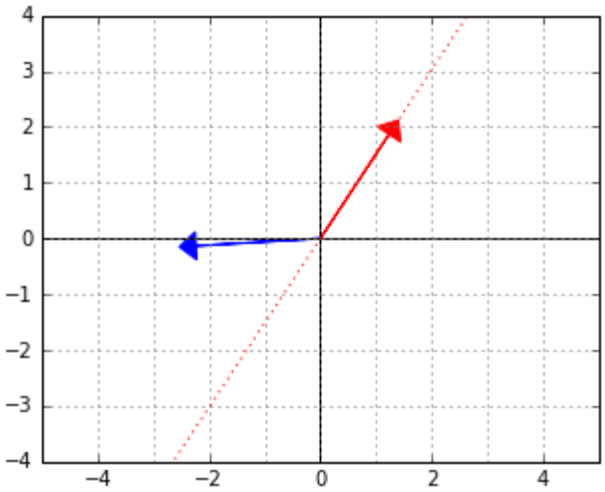
What it means, in simple terms, is that when you fit linear models in a skewed space, your resulting weight vectors will always be of the form $\mathbf{\Sigma}^{-1}\mathbf{z}$. Or, in other words, $\mathbf{\Sigma}^{-1}$ is a *transformation, which maps from "skewed perpendiculars" to "true perpendiculars"*. Let me show you what this means visually.

Consider again the two "orthogonal" vectors from the skewed world example above:



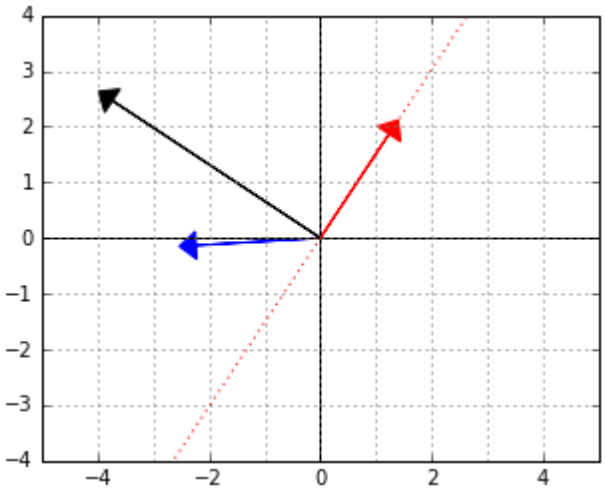
Let us interpret the blue vector as an element of the *dual space*. That is, it is the $\mathbf{z}$ vector in a linear functional $\mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{v}$. The red vector is an element of the "data space", which would be mapped to 0 by this functional (because the two vectors are "orthogonal", remember).

For example, if the blue vector was meant to be a linear classifier, it would have its separating line along the red vector, just like that:



If we now wanted to use this classifier, we could, of course, work in the "skewed space" and use the expression $\mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{v}$ to evaluate the functional. However, why don't we find the actual *normal* $\mathbf{w}$ to that red separating line so that we wouldn't need to do an extra matrix multiplication every time we use the function?

It is not too hard to see that $\mathbf{w} = \mathbf{\Sigma}^{-1}\mathbf{z}$ will give us that normal. Here it is, the black arrow:



Therefore, next time, whenever you see expressions like $\mathbf{w}^T\mathbf{\Sigma}^{-1}\mathbf{v}$ or $(\mathbf{v}-\mathbf{w})^T\mathbf{\Sigma}^{-1}(\mathbf{v}-\mathbf{w})$, remember that those are simply *inner products and (squared) distances* in a skewed space, while $\mathbf{\Sigma}^{-1}\mathbf{z}$ is a *conversion from a skewed normal to a true normal.* Also remember that the "skew" was estimated by pretending that the data were normally-distributed.

Once you see it, the role of the covariance matrix in some methods like the Fisher's discriminant (https://en.wikipedia.org/wiki/Linear_discriminant_analysis) or Canonical correlation analysis (https://en.wikipedia.org/wiki/Canonical_correlation) might become much more obvious.

The Dual Space Metric Tensor

"But wait", you should say here. "You have been talking about expressions like $\mathbf{w}^T\mathbf{\Sigma}^{-1}\mathbf{v}$ all the time, while things like $\mathbf{w}^T\mathbf{\Sigma}\mathbf{v}$ are also quite common in practice. What about those?"

Hopefully you know enough now to suspect that $\mathbf{w}^T\mathbf{\Sigma}\mathbf{v}$ is again an inner product or a squared norm in some deformed space, just not the "internal data metric space", that we considered so far. Which space is it? It turns out it is the "internal *dual* metric space". That is, whilst the expression $\mathbf{w}^T\mathbf{\Sigma}^{-1}\mathbf{v}$ denoted the "new inner product" between the *points*, the expression $\mathbf{w}^T\mathbf{\Sigma}\mathbf{v}$ denotes the "new inner product" between the *parameter vectors*. Let us see why it is so.

Consider an example again. Suppose that our space transformation $\mathbf{A}$ scaled all points by 2 along the x axis. The point (1,0) became (2,0), the point (3, 1) became (6, 1), etc. Think of it as changing the units of measurement - before we measured the x axis in kilograms, and now we measure it in pounds. Consequently, the norm of the point (2,0) according to the new metric, $|(2,0)|_{\Sigma^{-1}}$ will be 1, because 2 pounds is still just 1 kilogram "deep down inside".

What should happen to the *parameter ("direction")* vectors due to this transformation? Can we say that the parameter vector (1,0) also got scaled to (2,0) and that the norm of the parameter vector (2,0) is now therefore also 1? No! Recall that if our initial data denoted kilograms, our dual vectors must have denoted "units per kilogram". After the transformation they will be denoting "units per pound", correspondingly. To stay consistent we must therefore convert the parameter vector ("1 unit per kilogram", 0) to its equivalent ("0.5 units per pound",0). Consequently, the norm of the parameter vector (0.5,0) in the new metric will be 1 and, by the same logic, the norm of the dual vector (2,0) in the new metric must be 4. You see, the "importance of a parameter/direction" gets scaled inversely to the "importance of data" along that parameter or direction.

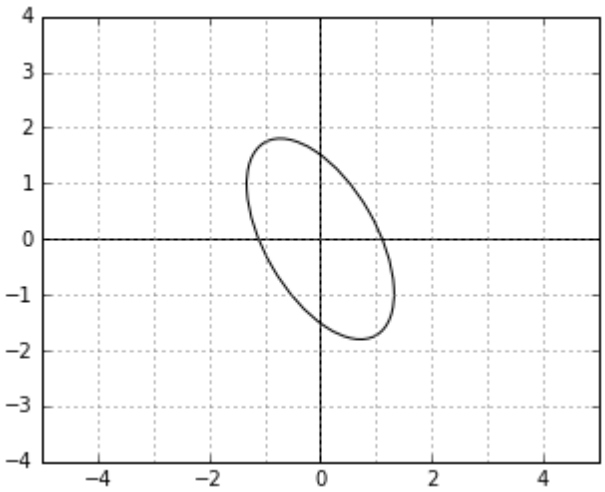
More formally, if $\mathbf{x}' = \mathbf{A}\mathbf{x}$, then

$$\begin{aligned} f_{\mathbf{w}}(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}^{-1}\mathbf{x}' \\ &= ((\mathbf{A}^{-1})^T\mathbf{w})^T\mathbf{x}' = f_{(\mathbf{A}^{-1})^T\mathbf{w}}(\mathbf{x}'). \end{aligned} \tag{10}$$

This means, that the transformation $\mathbf{A}$ of the data points implies the transformation $\mathbf{B} := (\mathbf{A}^{-1})^T$ of the dual vectors. The metric tensor for the dual space must thus be:

$$(\mathbf{B}\mathbf{B}^T)^{-1} = ((\mathbf{A}^{-1})^T\mathbf{A}^{-1})^{-1} = \mathbf{A}\mathbf{A}^T = \mathbf{\Sigma}. \tag{11}$$

Remember the illustration of the "unit circle" in the $\mathbf{\Sigma}^{-1}$ metric? This is how the unit circle looks in the corresponding $\mathbf{\Sigma}$ metric. It is rotated by the same angle, but it is stretched in the direction where it was squished before.



Intuitively, the norm ("importance") of the dual vectors along the directions in which the data was stretched by $\mathbf{A}$ becomes proportionally larger (note that the "unit circle" is, on the contrary, "squished" along those directions).

But the "stretch" of the space deformation in any direction can be measured by the variance of the data. It is therefore not a coincidence that $\mathbf{w}^T\mathbf{\Sigma}\mathbf{w}$ is exactly the variance of the data along the direction $\mathbf{w}$ (assuming $|\mathbf{w}| = 1$).

The Covariance Estimate

Once we start viewing the covariance matrix as a transformation-driven metric tensor, many things become clearer, but one thing becomes extremely puzzling: *why is the inverse covariance of the data a good estimate for that metric tensor*? After all, it is not obvious that $\mathbf{X}^T\mathbf{X}/n$ (where $\mathbf{X}$ is the data matrix) must be related to the $\mathbf{\Sigma}$ in the distribution equation $\exp(-0.5\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x})$.

Here is one possible way to see the connection. Firstly, let us take it for granted that if $\mathbf{X}$ is sampled from a symmetric Gaussian, then $\mathbf{X}^T\mathbf{X}/n$ is, on average, a unit matrix. This has nothing to do with

transformations, but just a consequence of pairwise independence of variables in the symmetric Gaussian.

Now, consider the transformed data, $\mathbf{Y} = \mathbf{X}\mathbf{A}^T$ (vectors in the data matrix are row-wise, hence the multiplication on the right with a transpose). What is the covariance estimate of $\mathbf{Y}$?

$$\mathbf{Y}^T\mathbf{Y}/n = (\mathbf{X}\mathbf{A}^T)^T\mathbf{X}\mathbf{A}^T/n = \mathbf{A}(\mathbf{X}^T\mathbf{X})\mathbf{A}^T/n \approx \mathbf{A}\mathbf{A}^T, \tag{12}$$

the familiar tensor.

This is a place where one could see that a covariance matrix may make sense outside the context of a Gaussian distribution, after all. Indeed, if you assume that your data was generated from *any* distribution P with uncorrelated variables of unit variance and then transformed using some matrix $\mathbf{A}$, the expression $\mathbf{X}^T\mathbf{X}/n$ will still be an estimate of $\mathbf{A}\mathbf{A}^T$, the metric tensor for the corresponding (dual) space deformation.

However, note that out of *all* possible initial distributions P, the normal distribution is exactly the one with the maximum entropy (https://en.wikipedia.org/wiki/Maximum_entropy_probability_distribution#Specified_variance:_the_normal_distribution) , i.e. the "most generic". Thus, if you base your analysis on the mean and the covariance matrix (which is what you do with PCA, for example), you could just as well assume your data to be normally distributed. In fact, a good rule of thumb is to remember, that whenever you even *mention* the word "covariance matrix", you are implicitly fitting a Gaussian distribution to your data.

Posted by Konstantin @ 11:04 pm

Tags: Algebra (http://fouryears.eu/tags/algebra/) , Explanation (http://fouryears.eu/tags/explanation/) , Machine learning (http://fouryears.eu/tags/machine-learning/) , Mathematics (http://fouryears.eu/tags/mathematics/) , Probability theory (http://fouryears.eu/tags/probability-theory/) , Statistics (http://fouryears.eu/tags/statistics/) , Visualization (http://fouryears.eu/tags/visualization/)