could lead to pruning too soon. If only the
two features is meaningful, pruning before building out
the children could result in never discovering that rule.

Decision trees aren't quite as powerful as neural networks in the hypotheses they can entertain, because they can't make functions of features that are more complex than ANDs and ORs. (No squares or powers for example,) But they do have the advantage of inspectability: it's easy for even a layperson to see exactly why a decision was made.

Covariance & Correlation

Our statistics so far has dealt with tests of single variables in controlled environments. This is good for many kinds of experiments, but doesn't really fully capture what's going on in many real situations, where we have multiple interacting variables. For example, we'd probably be remiss if we didn't explain what a correlation was, or how to find the strength of one.

A correlation is partly derived from the Covariance, a generalization of variance to the multivariable case. If we have observations of pairs (X, Y),

then the covariance is

$$E[(X-E[X])(Y-E[Y])].$$

We can notice a few things here:

- This is a generalization insofar as $cov(X,X)$ is the same as $Var(X)$.
- If we multiply out terms and realize $E[E[X]Y] = E[XE[Y]] = E[X]E[Y]$, this is equal to $E[XY]-E[X]E[Y]$
- Since $E[XY]=E[X]E[Y]$ for independent variables, it's zero if they're independent. (But the reverse isn't necessarily true.)
- If larger X tends to occur with larger Y, the covariance will be positive — their product in those cases will shift $E[XY]$ away from $E[X]E[Y]$. If they have the opposite relationship, the covariance is negative.

This has probably started to sound like a correlation. In fact, correlation is computed from the covariances, but scaled so that it doesn't depend on the magnitude of X and Y.

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

where σ_X and σ_Y are the standard deviations of X and Y.

<0.1 no corr
0.1–0.3 small corr
0.3–0.5 medium
0.5–1 strong

(We'll follow the textbook here and not try to correct for bias in the interest of keeping things simple.)

For example, suppose we have two six-sided die simulators that, thanks to a bug in initialization, always produce the same value. The covariance will then be the same as the variance for a single die, $\frac{1}{6}(-2.5)^2 + \frac{1}{6}(-1.5)^2 + \dots + \frac{1}{6}(2.5)^2$ since both differences are always the same. The bottom will be the same standard deviation squared - so, also just the variance. Top and bottom cancel, and the correlation is 1. This is bound to happen whenever the two variables are the same and perfectly correlated.

We can also determine what happens when the rolls are perfectly opposed: (1, 6), (2, 5) and so on. The covariance is $\frac{1}{6}(-2.5)(2.5) + \frac{1}{6}(-1.5)(1.5) + \dots + \frac{1}{6}(2.5)(-2.5)$ and we just have $-Var(X)$. The individual standard deviations are the same, and we're left with $\frac{-Var(X)}{Var(X)} = -1$. So -1 means perfect anticorrelation - there wasn't anything special about this example.

A caveat is that correlation using this coefficient only captures linear relationships between X and Y. If the plot of Y vs X doesn't look like a line, but some other kind of curve, the ρ statistic will only capture the goodness of fit of a line. It could well be 0 even if X and Y are not independent.

It's possible to get confidence intervals and do hypothesis testing with the correlation coefficient. For low values (<0.4) it's roughly normal and can

Another covariance example

X	X-E[X]	Y	Y-E[Y]	Z	Z-E[Z]
3	2.4	2	1.6	-2	-1.8
5	4.4	4	3.6	-4	-3.8
			-0.4	-1	-0.8
1	0.4	0			
-4	-4.6	-3	-3.4	4	4.2
-2	-2.6	-1	-1.4	2	2.2

$E[X] = \frac{3}{5} = 0.6 \quad E[Y] = \frac{2}{5} = 0.4 \quad E[Z] = -\frac{1}{5} = -0.2$

$$E[XY] = \frac{6+20+0+12+2}{5} = \frac{40}{5} = 8$$

$$E[YZ] = \frac{-4+-16+0+-12+-2}{5} = -\frac{34}{5} = -6.8$$

$$E[XZ] = \frac{-6+-20+-1+-16+-4}{5} = -\frac{47}{5} = -9.4$$

$Cov(X,Y) = E[XY] - E[X]E[Y] = 8 - 0.24 = 7.76$

$Cov(Y,Z) = E[YZ] - E[Y]E[Z] = -6.8 + 0.08 = -6.72$

$Cov(X,Z) = E[XZ] - E[X]E[Z] = -9.4 + 0.12 = -9.28$

$Var(X) = E[X^2] - E[X]^2 = \frac{9+25+1+16+4}{5} - (0.6)^2 = 11 - 0.36 = 10.64$

$Var(Y) = E[Y^2] - E[Y]^2 = \frac{4+16+0+9+1}{5} - (0.4)^2 = 6 - 0.16 = 5.84$

$Var(Z) = E[Z^2] - E[Z]^2 = \frac{4+16+1+16+4}{5} - (0.2)^2 = 8.2 - 0.04 = 8.16$

Covariance matrix

$$\begin{bmatrix} 10.64 & 7.76 & -9.28 \\ 7.76 & 5.84 & -6.72 \\ -9.28 & -6.72 & 8.16 \end{bmatrix}$$

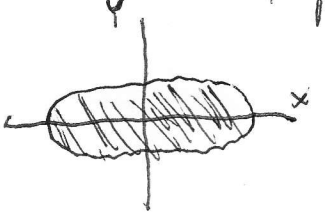
Sample ρ for X, Y

$$\frac{7.76}{\sqrt{10.64} \cdot \sqrt{5.84}} = 0.98$$

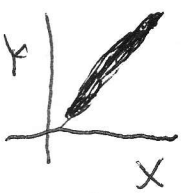
be treated that way (t-tests). For larger values, the distribution is skewed because it maxes out at 1, so you need to apply a transformation to make it normal again. We won't be too worried about that case here. (See section IX.8 of Lane.)

Multivariate normal

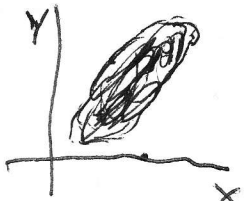
The multivariable version of the Gaussian is visualizable as an ellipse, or ellipsoid in 3D+. On each axis of the ellipse, the probability goes up and down again like a bell curve. If our variables are uncorrelated, the ellipse is nicely axis-aligned, with the PDF in each axis direction well-explained by a Gaussian.



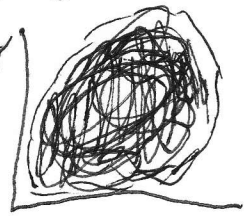
But the variables don't have to be uncorrelated for this general picture to hold - we can have multivariate Gaussians with correlated variables. The picture now becomes that of an ellipse that is rotated off-axis. If X and Y are almost perfectly correlated, we could have:



If there's a more mild correlation, then it might be more like:



or even:



Recall that we see Gaussians all the time in nature, so if we look at multiple variables at a time, we'll probably see multivariate Gaussians.

The parameters for a multivariate Gaussian are the vector of means for the variables, $\vec{\mu}$, and the covariance matrix Σ containing $\text{cov}(X_i, X_j)$ for entry i,j. (so the diag. is all var(X_i))

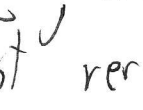
The actual equation is a bit of a beast, but I bring it up because we can see familiar ideas from the linear algebra portion of the course.

$$f(\vec{x}) = \frac{e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu})}}{\sqrt{2\pi^k \det(\Sigma)}}$$

where k is the number of dimensions (of $\vec{x}$).

In practice, you'll almost always interact with this distribution through a library, either to sample or get the PDF for some other purpose (like Bayesian reasoning).

Principal Component Analysis (PCA)

We may be interested in extracting the axes of our rotated Gaussian, for the purpose of "sketching" the data and identifying major trends. If two variables are well-correlated,  , we may just want to drop one of the variables as not very important,  , and then rethink our axes,  . This is more or less what PCA does: it identifies the directions of greatest variance and creates a new basis from those. The least important directions could be dropped entirely.

The way to do this is to find away to diagonalize the covariance matrix, so $\Sigma = PDP^{-1}$. If we can find a

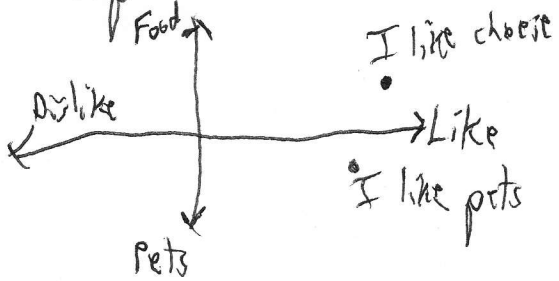
Change of basis so that no two [redacted] newly-defined variables have covariance, then we've found the basis where the directions best describe the changes in variation.

And that means... finding the eigenvalues and eigenvectors of the covariance matrix. Yes, recall that if we <u>can</u> diagonalize, those diagonal elements are actually eigenvalues, and P consists of eigenvectors, pointing in the directions of our new basis in diagonalized land. If we <u>can</u> diagonalize the covariance matrix, there is some way of thinking of the basis along axes where everything is independent. Those axes will be orthogonal, and further, if we rank by the size of the eigenvalue, that will give us the rank from direction of greatest importance (variance) to the one contributing the least variance. (We know that because our diagonalized matrix still can be interpreted as the covariance matrix — so its entries are also variances, in addition to being eigenvalues.)

<u>If</u> the data is centered at $\vec{0}$, then it turns out the covariance matrix can be estimated by just computing XX^T, where X is the data matrix (rows are observation vectors). If you use software to do PCA, check whether you're responsible for making the data "mean-centered."

Besides needing to be mean-centered, there's also a question of whether to scale in each direction before doing PCA, which is sensitive to the overall magnitude of the data in each direction. You could easily make distance measurements look huge to PCA by doing measurements in inches instead of miles, for example. Normalizing the data doesn't solve this issue, because it could magnify noise into big fluctuations. Ultimately, you may want to just pick measurement sizes that you feel are equally important somehow, or, just normalize and accept that some features will be unduly exaggerated or penalized.

PCA can be used for visualization of complex data, especially linguistic data. If I treat documents as binary vectors representing sets of words, [0...010...010...010...] (I like cheese) then performing PCA on the resulting matrix could reveal the top axes of variation for the documents.



But no automatic process can interpret the axes for you in this unsupervised method, so its success is in the eye of the beholder.