

PROF.

PATRICK WINSTON: Well that's the Kodo Drummers.

They're a group of about 30 or 40 Japanese people who live in a village on some island off the coast of Japan, and preserve traditional Japanese music.

It's an unusual semi communal group.

They generally run about 10 kilometers before breakfast, which is served at 5:00 AM.

Strange group.

Wouldn't miss a concert for the world, although they, alas, don't seem to be coming down to the Boston area very soon.

If you go to a concert from the Kodo Drummers--and you should-- and if you're no longer young, you'll want to bring earplugs.

Because, as we humans get older the dynamic range control in our inner ear tends to be less effective.

So that's why a person of my age might find some piece of music excruciatingly loud, whereas you'll think it's just fine.

Because you have better automatic gain control.

Just like in any kind of communication device there's a control on how intense the sound gets.

Ah, but I go off on a sidebar.

Many of you have looked at me in astonishment as I drink my coffee.

And you have undoubtedly have been saying to yourself, you know, Winston doesn't look like a professional athlete, but he seemed to have no trouble drinking his coffee.

So today's material is going to be pretty easy.

So I want to give you the side problem of thinking about how it's possible for somebody to do that.

How is it possible?

How would you make a computer program that could reach out and drink a cup of coffee, if it wanted a cup of coffee?

So that's one puzzle I'd like you to work on.

There's another puzzle, too.

And that puzzle concerns diet drinks.

This is a so-called Diet Coke.

Yeah, it's ripe.

If you take a Diet Coke and ask yourself, what would a dog think a Diet Coke is for?

That's another puzzle that you can work on while we go through the material of the day.

So this is our first lecture on learning, and I want to spend a minute or two in the beginning talking about the lay of the land.

And then we'll race through some material on nearest neighbor learning.

And then we'll finish up with the advertised discussion of sleep.

Because I know many of you think that because your MIT students you're pretty tough, and you don't need to sleep and stuff.

And we need to address that question before it's too late in the semester to get back on track.

All right.

So here's the story.

Now the way we're going to look at learning is there are two kinds.

There's this kind, and there's that kind.

And we're going to talk a little bit about both kinds.

The kind of the right is learning based on observations of regularity.

And computers are particularly good at this stuff.

And amongst the things that we'll talk about in connection with regularity based learning are today's topic, which is nearest neighbors.

Then a little bit downstream we'll talk about neural nets.

And then somewhere near the end of the segment, we'll talk about boosting.

And these ideas come from all over the place.

In particular, the stuff we're talking about today, nearest neighbors, is the stuff of which the field of pattern recognition-- it's the stuff of which pattern recognition journals are filled.

This stuff has been around a long time.

Does that mean it's not good?

I hope not, because that would mean that everything you learned in 1801 is not good, because the same course was taught 1910.

So it has been around a while, but it's extremely useful.

And it's the first thing to try when you have a learning problem, because it's the simplest thing.

And you always want to try the simplest thing before you try something more complex that you will be less likely to understand.

So that's nearest neighbors and pattern recognitions.

And the custodians of knowledge about neural nets, well this is sort of an attempt to mimic biology.

And I'll cast a lot of calumny on that when we get down there to talk about it.

And finally, this is the gift of the theoreticians.

So we in AI have invented some stuff, we've borrowed some stuff, we've stolen some stuff, we've championed some stuff, and we've improved some stuff.

That's why our discussion of learning will reach around all of these topics.

So that's regularity based learning.

And you can think of this as the branch of bulldozer computing.

Because, when doing these kinds of things, a computer's processing information like a bulldozer processes gravel.

Now that's not necessarily a good model for all the kinds of learning that humans do.

And after all, learning is one of the things that we think characterizes human intelligence.

So if we were to build models of it and understand that we have to go down this other branch, too.

And down this other branch we find learning ideas that are based on constraint.

And let's call this the human-like side of the picture.

And we'll talk about ideas that enable, for example, one-shot learning, where you learn something definite from each experience.

And we'll talk about explanation based learning.

By the way, do you learn by self explanation?

I think so.

I had an advisee once, who got nothing but A's and F's.

And I said, what are the subjects that you get A's in?

And why don't you get A's in all of your subjects?

And he said, oh, I get A's in the subjects when I convince myself the material is true.

So the learning was a byproduct of self explanation, an important kind of learning.

But alas, that's downstream.

And what we're going to talk about today is this path through the tree, nearest neighbor learning.

And here's how it works, in general.

Here's just a general picture of what we're talking about.

When you think of pattern recognition, or nearest neighbor based learning, you've got some sort of mechanism

that generates a vector of features.

So we'll call this the feature detector.

And out comes a vector of values.

And that vector of values goes into a comparator of some sort.

And that comparator compares the feature vector with feature vectors coming from a library of possibilities.

And by finding the closest match the comparator determines what some object is.

It does recognition.

So let me demonstrate that with these electrical covers.

Suppose they arrived on an assembly line and some robot wants to sort them.

How would it go about doing that?

Well it could easily use the nearest neighbor sorting mechanism.

So how would that work?

Well here's how it would work.

You would make some measurements.

And it we'll just make some measurements in two dimensions.

And one of those measurements might be the total area, including the area of the holes of these electrical covers.

Just so you can follow what I'm doing without craning your neck, let me see if I can find the electrical covers.

Yes, there they are.

So we've got one big blank one, and several others.

So we might also measure the hole area.

And this one here, this guy here, this big white one has no hole area, and its got the maximum amount of total area.

So it will find itself at that point in this space of features.

Then we've got the guy here, with room for four sockets in it.

That's got the maximum amount of hole area, as well as the maximum amount of area.

So it will be right straight up, maybe up here.

Then we have, in addition to those two, a blank cover, like this, that's got about 1/2 the total area that any cover can have, so we'll put it right here.

And finally, we've got one more of these guys.

Oh yes, this one.

1/2 the hole area, and 1/2 the total area.

So I don't know, let's see.

Where will that go?

Maybe about right here.

So now our robot is looking on the assembly line and it sees something coming along, and it measures the area.

And of course, there's noise.

There's manufacturing variability.

So it won't be precisely on top of anything.

But suppose it's right there.

Well it doesn't take any genius human, human or computer, to figure out that this must be one of those guys with maximum area and maximum hole area.

But now let's ask some other questions.

Where would [TAPPING ON CHALK BOARD], what would that be?

Or what would this be?

[TAPPING ON CHALK BOARD], and so on.

Well we have to figure out what those newly viewed objects are closest to in order to do an identification.

But that's easy.

We just calculate the distance to all of those standard, platonic, ideal descriptions of things, and we find out which is nearest.

But in general, it's a little easier to think about producing some boundaries between these various idealize places, so that we can just say, well which area is the object in?

And then we'll know instantaneously to what category it belongs.

So if we only had two, like the purple one and the yellow one, it would be easy.

Because, we would just construct a line between the two, with a line between the purple and yellow as a perpendicular bisector.

And so drawing it out instead of talking about it, if there were only two, that would be the boundary line.

Anything south of the dotted line would be purple, and anything north would be yellow.

And now we can do this with all the points, right?

So we can figure out-- oh could you, Pierre, could you just close the lap top please?

So if we want to do this with all these guys it would go something like this-- I better get rid of these dotted x's before they confuse me.

Let's see, if these were the only two points, then we would want to construct a perpendicular bisector between the line joining them.

And if these two were the only points, I would want to construct this perpendicular bisector.

And if these two were the only points, I would want to construct a perpendicular bisector.

And if these two points were the only ones involved I'd want to construct-- oh, you see what I'm doing?

I'm constructing perpendicular bisectors, and those are exactly the lines that I need in order to divide up this space.

And it's going to divide up like this.

And I won't say we'll give you a problem like this on an examination, but we have every year in the past ten.

To divide up a space and produce-- something we would like to give a name.

You know, Rumpelstiltskin effect, when you have a name you get power over it.

So we're going to call these decision boundaries.

OK so those are the simple decision boundaries, produced in a sample space, by a simple idea.

But there is a little bit more to say about this.

Because, I've talked about this as if we're trying to identify something.

There's another way of thinking about it that's extremely important.

And that is this.

Suppose I come in with a brand new cover, never before seen.

And I only measure, well let's say I only measure the hole area.

And the hole area has that value.

What is the most likely total area?

Well I don't know.

But there's a kind of weak principle of, if something is similar in some respects, it's likely to be similar in other respects.

So I'm going to guess, if you hold a knife to my throat and back me into a corner, that it's total area is going to be something like that orange cover whole, total area.

So this is a contrived example, and I don't make too much of it.

But I do want to make a lot of that first principal, over there.

And that is the idea that, if something is similar in some respects, it's likely to be similar in other respects.

Because that's what most of education is about.

Fairy tales, legal cases, medical cases, business cases-- if you can see that there are similar in some respects to a situation you've got now, then it's likely that they're going to be similar in other respects, as well.

So when we're learning, we're not just learning to recognize a category, we're learning because we're attempting to apply some kind of precedent.

That's the story on that.

Well that's a simple idea but does it have any application?

The answer is sure.

Here's an example.

My second example, the example of cell identification.

Suppose you have some white blood cells, what might you do?

You might measure the total area of the cell.

And not the hole area, but maybe the nucleus area.

And maybe you might measure four or five other things, and put this thing in a high dimensional space.

You can still measure the nearness in a high dimensional space.

So you can use the idea to do that.

It works pretty well.

A friend of mine once started a company based on this idea.

He got wiped out, of course, but it wasn't his fault.

What happened is that somebody invented a better stain and it became much easier to just do the recognition by brute force.

So let's see, that's two examples.

the introductory example of the holes of the electrical covers, and the example of cells.

And what I want to do now is show you how the idea can reappear in disguised forms in areas where you might

not expect to see it.

So consider the following problem.

You have a collection of articles from magazines.

And you're interested in learning something about how to address a particular question.

How do you go about finding the articles that are relevant to your question?

So this is a puzzle that has been studied for decades by people interested in information retrieval.

And here's the simple way to do it.

I'm going to illustrate, once again, in just two dimensions.

But it has to be applied in many, many dimensions.

The idea is you count up the words in the articles in your library, and you compare the word counts to the word counts in your probing question.

So you might be interested in 100 words.

I'm only going to write two on the board for illustration.

So we're going to think about articles from two magazines.

Well first of all, what words are we going to use?

One word is going to be hack, and that will include all derivatives of hack-- hacker, hacking, and so on.

And the other word is going to be computer.

And so it would not be surprising for you to see that articles from Wired Magazine might appear in places like this.

They would involve lots of uses of the word computer, and lots of uses of the word hack.

And now for the sake of illustration, the second magazine from which we are going to draw articles is Town and Country.

It's a very tony magazine, and the people who read out Town and Country tend to be social parasites.

And they still use the word hack.

Because you can talk about hacking, there's some sort of specialize term of art in dealing with horses.

So all the Town and Country articles would be likely to be down here somewhere.

And maybe they would be one like that when they talk about hiring some computer expert to keep track of the results so the weekly hunt, or something.

And now, in you come with your probe.

And of course your probe question is going to be relatively small.

It's not going to have a lot of words in it.

So here's your here's your probe question.

Here's your unknown.

Which article's going to be closest?

Which articles are going to be closest?

Well, alas, all those Town and Country articles are closest.

So you can't use the nearest neighbor idea, it would seem.

Anybody got a suggestion for how we might get out of this dilemma?

Yes, Christopher.

CHRISTOPHER: If you're looking for word counts and you want to include some terms of computer, then wouldn't you want to use that as a threshold, rather than the nearest neighbor?

PROF.

PATRICK WINSTON: I don't know, it's a good idea.

It might work, who knows.

Doug?

DOUG: Instead of using decision boundaries that are perpendicular bisectors, if you treated Wired and Town and Country as sort of this like, [INAUDIBLE] targets.

And they would look like some [? great radial ?], here.

I guess, some radius around curves.

If it's within a certain radius then-- PROF.

PATRICK WINSTON: Yes?

[? SPEAKER 1: Are we, ?] necessarily, have it done with some sort of a [? politidy distance ?] metric?

PROF.

PATRICK WINSTON: Oh, here we go.

We're not going to use any [? politidy distance ?] metric.

We're going to use some other metric.

SPEAKER 1: Like alogrithmic, or whatnot?

PROF.

PATRICK WINSTON: Well, algorithmic, gees, I don't know.

[LAUGHTER] PROF.

PATRICK WINSTON: Let me give you a hint.

Let me give you a hint.

There are all those articles up there, out there, and out there, just for example.

And here are the Town and Country articles.

They're out there, and out there, for example.

And now our unknown is out there.

Anybody got an idea now?

Hey Brett, what do you think?

BRETT: So you sort of want the ratio.

Or in this case, you can take the angle-- PROF.

PATRICK WINSTON: Let's be-- ah, there we go, we're getting a little more sophisticated.

The angle between what?

BRETT: The angle between the vectors.

PROF.

PATRICK WINSTON: The vectors.

Good.

So we're going to use a different metric.

What we're going to do is, we're going to forget including a distance, and we're going to measure the angle between the vectors.

So the angle between the vectors, well let's actually measure the cosine of the angle between the vectors.

Let's see how we can calculate that.

So we'll take the cosine of the angle between the vectors, we'll call it theta.

That's going to be equal to the sum of the unknown values times the article values.

Those are just the values in various dimensions.

And then we'll divide that by the magnitude of the other vectors.

So we'll divide by the magnitude of  $u$ , and we'll divide by the magnitude of the art vector to the article.

So that's just the dot product right?

That's a very fast computation.

So with a very fast computation you can see if these things are going to be in the same direction.

By the way, if this vector here is actually identical to one of those articles, what will the value be?

Well then a cosine will be 0 and we'll get the maximum die of the cosine, which is 1.

Yeah, that will do it.

So if we use any of the articles to probe the article space, they'll find themselves, which is a good thing to have a mechanism do.

OK.

So that's just the dot product of those two vectors.

And it works like a charm.

It's not the most sophisticated way of doing these things.

There are hairy ways.

You can get a Ph.D. by doing this sort of stuff in some new and sophisticated way.

But this is a simple way.

It works pretty well.

And you don't have to strain yourself, much, to implement it.

So that's cool.

That's an example where we have a very non-standard metric.

Now let's see, what else can we do?

How about a robotic arm control?

Here we go.

We're going to just have a simple arm.

And what we want to do is, we want to get this arm to move that ball along some trajectory at a speed, velocity, and acceleration that we have determined.

So we've got two problems here.

Well let's see, we've got two problems because, first of all, we've got angles,  $\theta_1$  and  $\theta_2$ .

It's a 2 degree of freedom of arm, so there are only two angles.

So the first problem we have is the kinematic problem of translating the (x,y)-coordinates of the ball, the desired ones, into the  $\theta_1$ ,  $\theta_2$  space.

That's simple kinematic problem.

No  $f = ma$  there.

It Doesn't involve forces, or time, or acceleration, anything.

Pretty simple.

But then we've got the problem of getting it to go along that trajectory with positions, speeds, and accelerations that we desire.

And now you say to me, well I've got 801, I can do that.

And that's true, you can.

Because, it's Newtonian mechanics.

All you have to do is solve the equations.

There are the equations.

Good luck.

Why are they so complicated?

Well because of the complicated geometry.

You notice we've got some products of  $\theta_1$  and  $\theta_2$  in there, somewhere, I think?

You've got  $\theta_2$ 's.

I see an acceleration squared.

And yeah, there's a  $\dot{\theta}_1$  times a  $\dot{\theta}_2$ .

A velocity times a velocity.

Where the hell did that come from?

I mean it's supposed to be  $f = ma$ , right?

Those are Coriolis forces, because of the complicated geometry.

OK.

So you hire Berthold Horn, or somebody, to work these equations out for you.

And he comes up with something like this.

And you try it out and it doesn't work.

Why doesn't it work?

It's Newtonian mechanics, I said.

It doesn't work because we forgot to tell Berthold that there's friction in all the joints.

And we forgot to tell him that they've worn a little bit since yesterday.

And we forgot that the measurements we make on the lab table are not quite precise.

So people try to do this.

It just doesn't work.

As soon as you get a ball of a different weight you have to start over.

It's gross.

So I don't know.

I can do this sort of thing effortlessly, and I couldn't begin to solve those equations.

So let's see.

What we're going to do is we're going to forget about the problem for a minute.

And we're going to talk about building ourselves a gigantic table.

And here's what's going to be on the table.

Theta 1, theta 2, theta 3, oops, there are only two.

So that's theta 1 again, but it's the velocity, angular velocity.

And then we have the accelerations.

So we're going to have a big table of these things.

And what we're going to do, is we're going to give this arm a childhood.

And we're going to write down all the combinations we ever see, every 100 milliseconds, or something.

And the arm is just going to wave around like a kid does in the cradle.

And then, we're not quite done.

Because there are two other things we're going to record.

Can you guess what they are?

There are going to be the torque on the first motor, and the torque on the second motor.

And so now, we've got a whole bunch of those records.

The question is, what do we got to do with it?

Well here's what we're going to do it.

We're going to divide this trajectory that we're hoping to achieve, up into little pieces.

And there's a little piece.

And in that little piece nothing is going to change much.

There's going to be an acceleration, velocity, position.

And so we can look those up in the table that we made in the childhood.

And we'll look around and find the closest match, and this will be the set of values for the positions, velocities, and accelerations that are associated with that particular movement.

And guess what we can do now?

We can say, in the past, the torques associated with that particular little piece of movement lie right there.

So we can just look it up.

Now this method was thought up and rejected, because computers weren't powerful enough.

And then, this is the age of recycling, right?

So the idea got recycled when computers got strong enough.

And it works pretty well, for things like this.

But you might say to me, well can it do the stuff that we humans can do, like this?

And the answer is, let's look.

So this is a training phase, it's going through its childhood.

You see what's happening is this.

The initial table won't be very good.

But that's OK.

Because there are only a small number of things that it's important for you to be able to do.

So when you try those things it's still writing into the table.

So the next time you try that particular motion, it's going to be better at it, because its got better stuff to interpolate [? amongst ?] in that table.

So that's why this thing is getting better and better as it goes on.

That's as good as I was doing.

Pretty good, don't you think?

There's just one thing I want to show at the end of this clip just for fun.

Maybe you've seen some old Zorro movies?

So here's a little set up where this thing has learned to use a lash.

So here's the lash, and there's a candle down there.

So watch this.

Pretty good, don't you think?

So how fast does the learning take place?

Let me go back to that other slides and show you.

So here's some graphs to show you how fast goes, boom.

That gives you the curves of how well the robot arm can go along a straight line, after no practice with just some stuff recorded in the memory.

And then with a couple of practice runs do give it better values amongst which to interpolate.

So I think that's pretty cool.

So simple, but yet so effective.

But you still might say, well, I don't know, it might be something that can be done in special cases.

I wonder if old Winston uses something like that when he drinks his coffee?

Well we' ought to do the numbers and see if it's possible.

But I don't want to use coffee, it's the baseball season.

We're approaching the World Series.

We might as well talk about professional athletes.

So let's suppose that this is a baseball pitcher.

And I want to know how much memory I'll need to record a whole lot of pitches.

Is there a good pitcher these days?

The Red Socks suck so I don't do Red Socks.

Clay Buchholz, I guess.

I don't know, some pitcher.

And what we're going to do, is we're going to say for each of these little segments were going to record 100 bytes per joint.

And we've got joints all over the place.

I don't know how many are involved in doing a baseball pitch, but let's just say we have had 100 joints.

And then we have to divide the pitch up into a bunch of segments.

So let's just say for sake of argument that there are 100 segments.

And how many pitches does a pitcher throw in a day?

What?

SPEAKER 2: In a day?

PROF.

PATRICK WINSTON: In a day, yeah.

This, we all know, is about 100.

Everybody knows that they take them out after about 100 pitches.

So what I want to know is how much memory we need to record all the pitches a pitcher pitches in his career.

So we still have to work on this little bit more.

How many days a year does a pitcher pitch?

Well, they've got winter ball, and that sort of thing, so let's just approximate it as 100.

I don't know, some of these may be a little high, some of the others may be a low.

And of course, the career-- just to make things easy-- is 100 years.

So that's one, two, three, four, five, six.

So we have 10 to the 12th bytes.

Is that the hopelessly big to store in here?

CHRISTOPHER: 10 to 100 [INAUDIBLE] or just 100 times throwing?

PROF.

PATRICK WINSTON: 100 pitches in a day-- Christopher's asking some detail-- and what we're going to do is we're going to record everything there is to know about one pitch, and then we're going to see how many pitches, he pitches in his lifetime.

And we're going to record all that.

Trust me.

Trust me.

OK. so we want to know if this is actually a practical scale.

And this, by the way, is cocktail conversation, who knows, right?

But it's useful to work out these numbers, and know some of these numbers.

So the question we have to ask is, how much computation is in there?

And the first question relevant to that is, how many neurons do we have in our brain?

Volunteer?

Neuroscience?

No one to volunteer?

All right.

Well this is a number you should know, because this is what you've got in there.

There are 10 to the 10th neurons in the brain, of which 10 to the 11th are in the cerebellum, alone.

What the devil do I mean by that?

I mean that your cerebellum is so full of neurons that it dwarfs the rest of the brain.

So if you exclude the cerebellum, you've got about 10 to 10th neurons.

And there about 10 to the 11th neurons in the cerebellum, alone.

What's the cerebellum for?

Motor control.

Interesting.

So we're a little short.

Oh, but we forget, that's just the number of neurons.

We have to count up the number of synapses.

Because conceivably, we might be able to adjust those synapses, right?

So how many synapses does a neuron have?

The answer is, it depends.

But the ones in the cerebellum-- I should be pointing back there, I guess-- 10 to the 5th.

So if we add all that up we have 10 to the 16th.

No problem.

It's just that existence proves that you don't have to worry too much about having storage.

So maybe our cerebellum functions, in some way, as a gigantic table.

And that's maybe how we learn motor skills, by filling up that table as we run around emerging from the cradle, learning how to manipulate ourselves as we go on.

So that's the story on arm control.

Now all this is pretty straightforward, easy to understand.

And of course, there are some problems.

Problem number one, what if the space of samples looks like this?

[TAPPING ON CHALK BOARD] What's going to happen in that case?

Well what's going to happen in that case is that the-- let's see, which values are going to be more important?

The x values, right?

The y values are spread out all over the place.

So you'd like the spread of the data to sort of be the same in all the dimensions.

So is there anything we can do to arrange for that to be true?

Sure, we can just normalize the data.

So we can borrow from our statistics course and say, well, let's see, we're interested in  $x$ .

And we know that the variance of  $x$  is equal to  $\frac{1}{n}$  times the sum of the values, minus the mean value squared.

That's a measure of how much the data spreads out.

So now, instead of using  $x$ , we can use  $x'$ , which is equal to  $x$  over  $\sigma$ .

What's the variance of that going to be?

$x'$  over  $\sigma$  sub  $x$ .

Anybody see, instantaneously, what the variance of that's going to be?

Or do we have to work it out?

It's going to be 1, Work out the algebra for me.

It's obvious, it's simple.

Just substitute  $x'$  into this formula for variance, and do the algebraic high school manipulation.

And you'll see that the variance turns out not to be of this new variable, this transformed variable you want.

So that problem, the non uniformity problem, the spread problem, is easy to handle.

What about that other problem?

No cake without flour?

What if it turns out that the data-- you have two dimensions and the answer, actually, doesn't depend on  $y$  at all.

What will happen?

Then you're often going to get screwy results, because it'll be measuring a distance that is merely confusing the answer.

So problem number two is the what matters problem.

Write it down, what matters.

Problem number three is, what if the answer doesn't depend on the data at all?

Then you've got the trying to build a cake without flour.

Once somebody asked me-- a classmate of mine, who went on to become an important executive in an important credit card company-- asked me if we could use artificial intelligence to determine when somebody was going to go bankrupt?

And the answer was, no.

Because the data available was data that was independent of that question.

So he was trying to make a cake without flour, and you can't do that.

So that concludes what I want to say about nearest neighbors.

No I want to talk a little bit about sleep.

Over there on that left-side branch, now disappeared, we talked about the human side of learning.

And I said something about one-shot, an escalation based learning.

And what that means is, you don't learn without problem solving.

And the question is, how is problem solving related to how much sleep you get?

And to answer questions like that, of course, you want to go to the people who are the custodians of the kind of knowledge you are interested in.

And so you would say, who are the custodians of knowledge about how much sleep you need?

And what happens if you don't get it?

And the answer is the United States Army.

Because they're extremely interested in what happens when you cross 10 or 12 time zones, and have no sleep, and have to perform.

So they're very interested in that question.

And they got even more interested after the first Gulf War, which was the most studied war in history, up to that time.

Because, there were after action reports they were full of examples like this.

The US Forces, in a certain part of the battlefield, and drawn up for the night.

And those are Bradley fighting vehicles, there, and back here Abrams tanks.

And they're all just kind of settling down for good night's sleep.

They've been up for about 36 hours straight, by the way.

When, much to their amazement, across their field-of-view came a column of Iraqi vehicles.

And both sides were enormously surprised.

A firefight broke out.

The lead vehicle, over here, on the Iraqi side caught on fire.

So these guys, in the Bradley fighting vehicles, went around to investigate, whereupon, these guys started blasting away, in acts of fratricidal fire.

And the interesting thing is that all these folks here swore in the after action reports that they were firing straight ahead.

And what happened was their ability to put ordnance on target was not impaired at all.

But their idea of where the target was, what the target was, whether it was a target, was all screwed up.

So this led to a lot of experiments in which people were sleep deprived.

And by the way, you think you're a tough MIT student, right?

These are Army Rangers.

It doesn't get any tougher than this, believe me.

So here's one of the experiments that was performed.

In those days they had what they called fire control teams.

And their job is to take information from an observer, over here, about a target, over here.

And tell the artillery, over here, where to fire.

So they kept some of these folks up for 36 hours straight.

And after 36 hours they all said, we're doing great.

And at that time they were bringing fire down on hospitals, mosques, churches, schools, and themselves.

Because, they couldn't do the calculations anymore, after 36 hours without sleep.

And now you say to me, well I'm a MIT student, I want to see the data.

So let's have a look at the data.

OK.

So there it goes.

That's what happens to you after 72 hours without sleep.

These are simple things to do.

Very simple calculations you have to do in your head, like adding numbers, spelling words, and things like that.

So after 72 hours without sleep, your performance relative to what you were at the beginning is about 30%.

So loss of sleep destroys ability.

[BELL RINGING] Sleep loss accumulates.

So you say, well I need eight hours of sleep-- and what you need, by the way, varies-- but I'm going to get by was

seven hours of sleep.

So after 20 days of one hour's worth of sleep deprivation, you're down about 25%.

If you say, well I need eight hours of sleep, but I'm going to have to get by with just six, after 20 days of that, you're down to about 25% of your original capability.

So you might say, well does caffeine help?

Or naps, naps in this case.

And the answer is, yes, a little bit.

Some people argue that you get the more affect out of the sleep that you do get if you divide it into two.

Winston Churchill always took a three hour nap in the afternoon.

He said that way he got a day and a half's worth of work out of every day.

He got the full amount of sleep.

But he divided it into two pieces.

Here's the caffeine one.

So caffeine does help.

And now you say, well, shoot, I think I'm going to take it kind of easy this semester.

And I'll just work hard during the week before finals.

Maybe I won't even bother sleeping for the 24 hours before the 6034 final.

That's OK.

Well let's see what will happen.

So let's work the numbers.

Here is 24 hours.

And that's where your effectiveness is after 24 hours.

Now let's go over to the same amount of effectiveness on the blood alcohol curve.

And it's about the level at which you would be legally drunk.

So I guess what we ought to do is to check everybody as they come in for the 6034 final, and arrest you if you've been 24 hours without sleep.

And not let you take any finals again, for a year.

So if you do all that, you might as well get drunk.

And now we have one thing left to do today.

And that is address the original question of, why it is that the dogs and cats in the world think that the diet drink makes people fat?

What's the answer?

It's because only fat guys like me drink this crap.

So since the dogs and cats don't have the ability to tell themselves stories, don't have that capacity to string together events into narratives, they don't have any way of saying, well this is a consequence of desiring not to be fat.

Not a consequence of being fat.

They don't have that story.

And so what they're doing is something you have to be very careful about.

And that thing you have to be very careful about is the confusion of correlation with cause.

They see the correlation, but they don't understand the cause, so that's why they make a mistake.