Learning with Prototypes

CS771: Introduction to Machine Learning

Purushottam Kar

What are Vectors

Consider a d-dimensional real vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d], \mathbf{x}_i \in \mathbb{R}$

For a physicist, a vector is a way to encode a magnitude and direction

For a mathematician, a vector is an object in a vector space

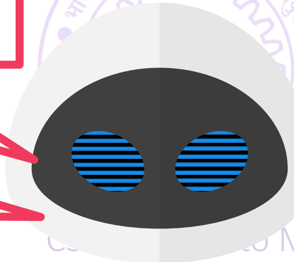
For an ML person, a vector is simply a list of numbers, each number representing a useful piece of information about an object

- **Example**: spam filter, a vector stored which words occurred in email
- **Example**: image recognition, a vector stored the pixel RGB values

Does this mean I will have to learn math to do ML?

A bit of math will be required but 1) it will be simple and 2) it will be totally worth it!

True, for me, vectors are just like arrays of numbers. However, sometimes I do indeed do math with them



SEE REFERENCE [DFO] SECTION 1.1 FOR A NICE INTRO

Operations on Vectors

Given two d-dimensional vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we can do the following

Sum of two vectors (is also a vector of same dimension)

$$a = \mathbf{x} + \mathbf{y} \in \mathbb{R}^d$$

Difference of two vectors (is also a vector of same dimension)

$$b = \mathbf{x} - \mathbf{y} \in \mathbb{R}^d$$

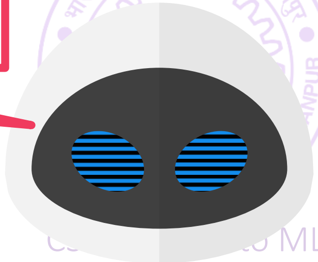
Dot product of two vectors (is a real number, possibly negative)

$$p = \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^d \mathbf{x}_i \mathbf{y}_i$$

Euclidean length of ... zero)

$$q = \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^d \mathbf{x}_i^2} = \sqrt{\mathbf{x}^\top \mathbf{x}}$$

Good point – just wait a second for some intuition. However, to be honest, sometimes it is good in ML not to hunt too much for intuition and just let the math be!



So I can take two feature vectors and add them. But what sense does it make to add two emails?

ML Operations on Feature Vectors

Given n feature vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n$

fea

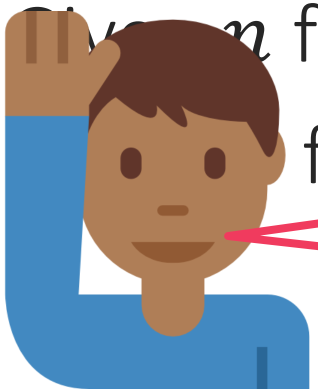
$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}^i$$

Euclidean distance between two fe

$$d_2(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|_2 = \sqrt{\sum_{j=1} (\mathbf{x}_j^1 - \mathbf{x}_j^2)}$$

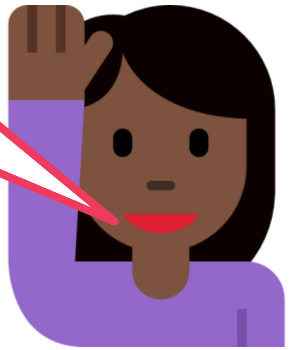
Distance of a feature vector from the average vector

$$d_2(\mathbf{x}^1, \boldsymbol{\mu}) = \|\mathbf{x}^1$$

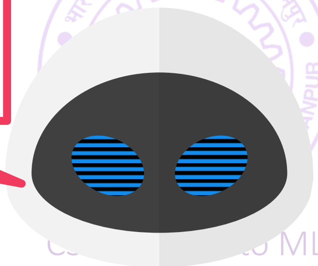


I can use this to find out if two emails are similar or very different from each other

I can use this to find out what an "average" email looks like ☺. So averaging vectors makes sense!



I can use this to find out if a new email is similar to an average spam email or an average regular email



Excellent! We are now ready for our first ML algorithm!!

Learning with Prototypes

- The basic mantra here is

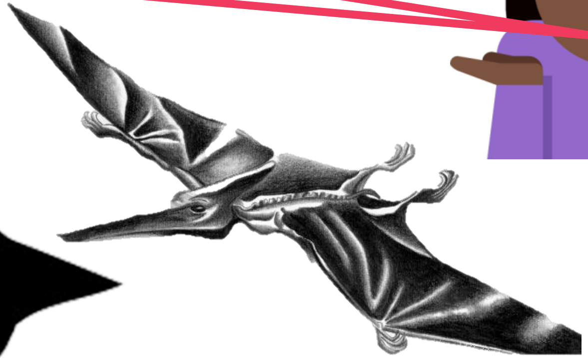
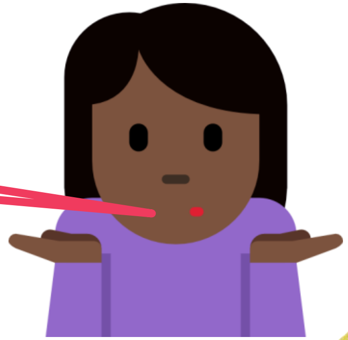
> *If a new email looks similar to an average spam email, it may be a spam email. On the other hand, if a new email looks similar to an average normal email, it should be normal*

- In ML, the word *prototype* is used to refer to something that is representative or captures the qualities of a class of objects?
- How can we do ML using prototypes?

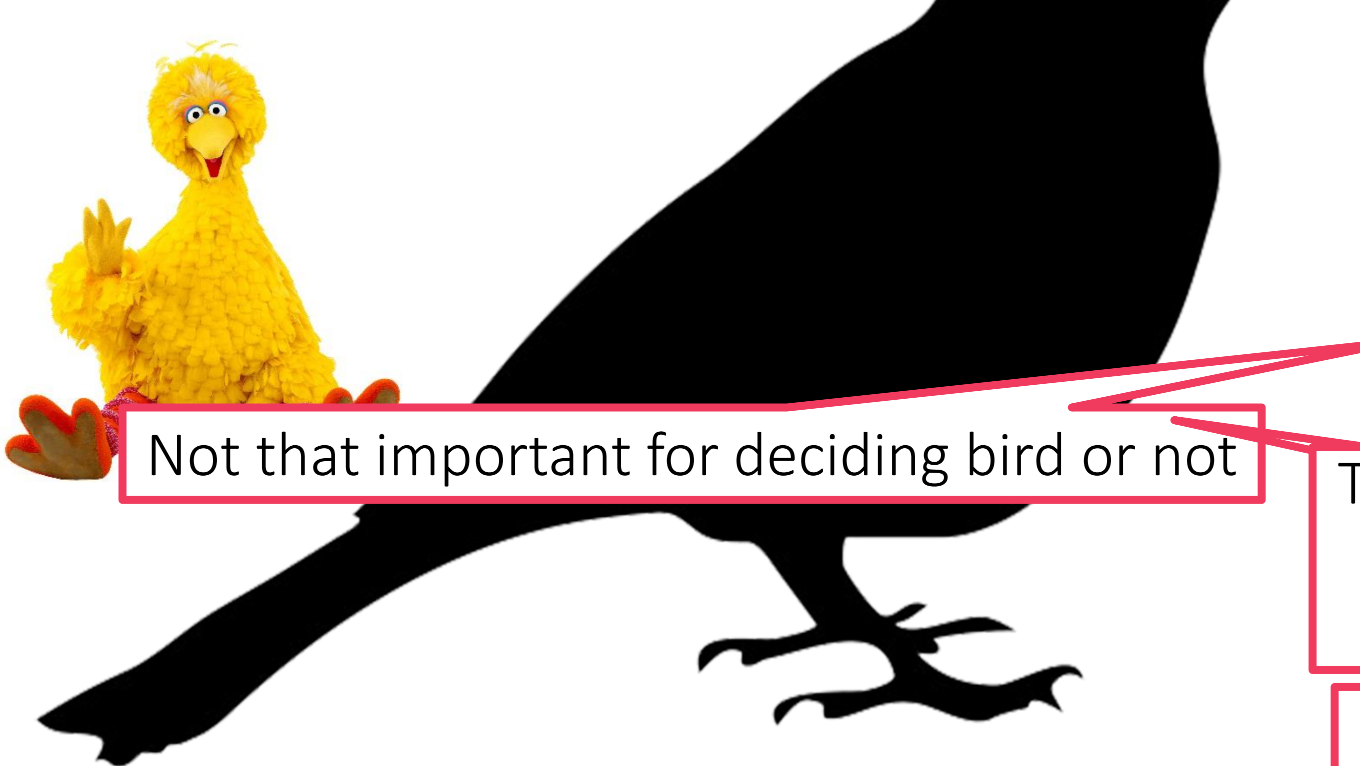
Bird or not

But the Pterosaur will still get classified as a bird – it has wings and a beak




Important for deciding bird or not



Has wings

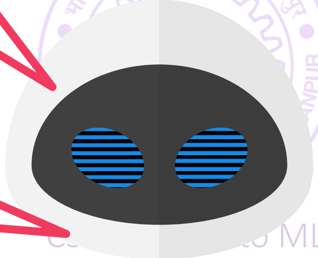
Can fly

Can sing

Not that important for deciding bird or not

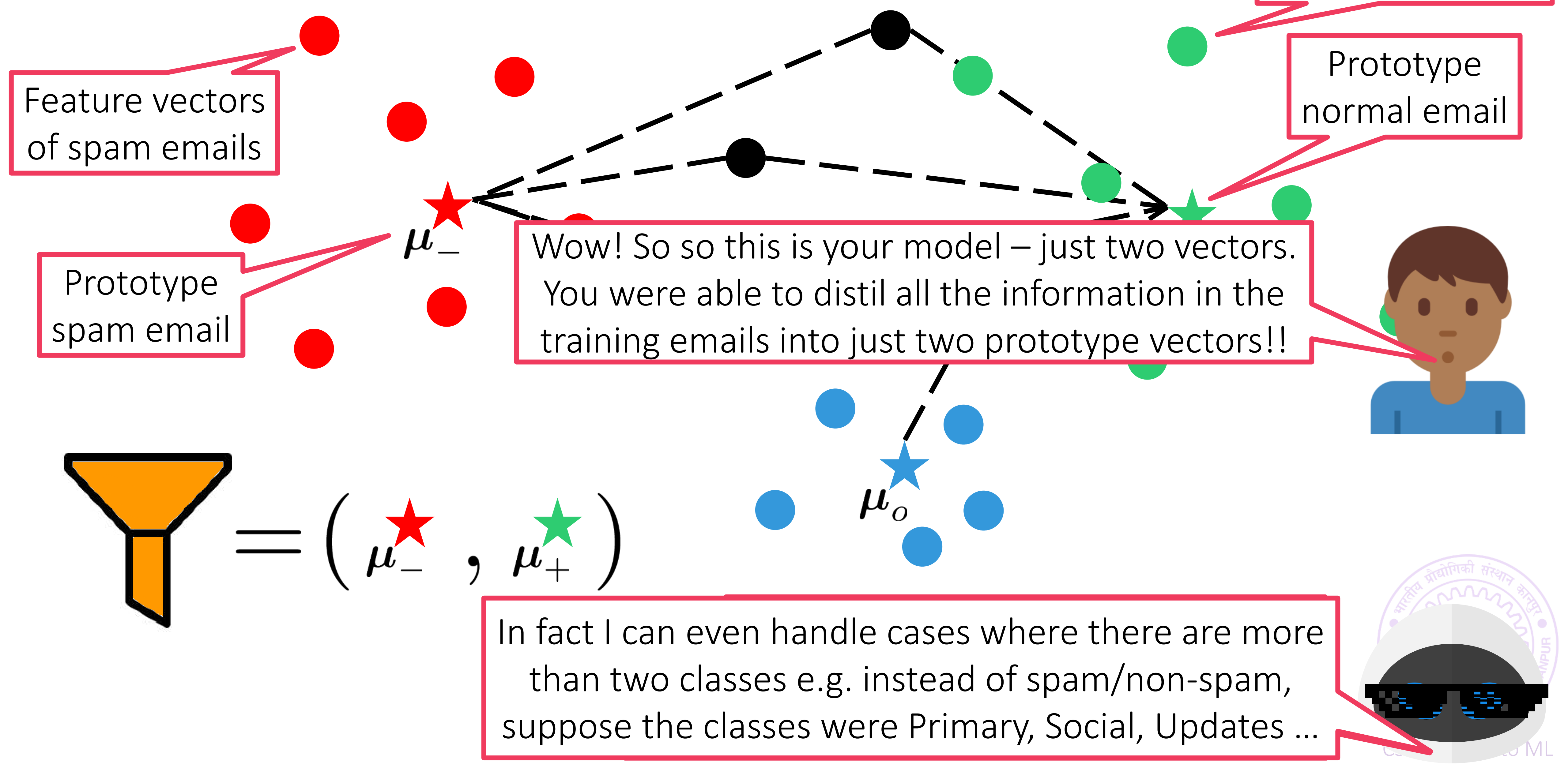
This means that our prototype is not good enough and we may need better features

What abstract qualities do we associate with “birdness”

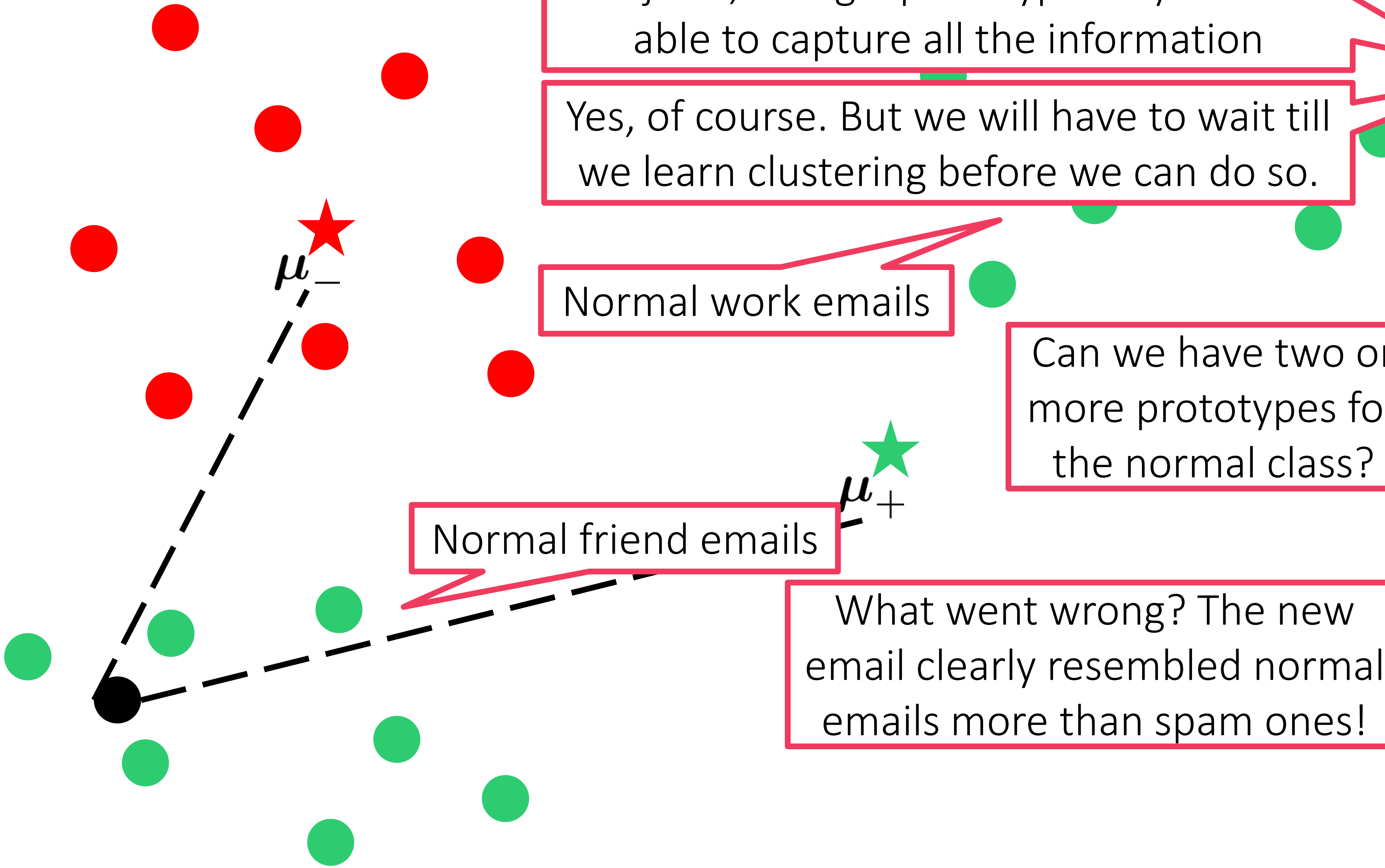


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Learning with Prototypes



Learning with Pro



If there is too much diversity in a class of objects, a single prototype may not be able to capture all the information

Yes, of course. But we will have to wait till we learn clustering before we can do so.

Normal work emails

Can we have two or more prototypes for the normal class?

Normal friend emails

What went wrong? The new email clearly resembled normal emails more than spam ones!

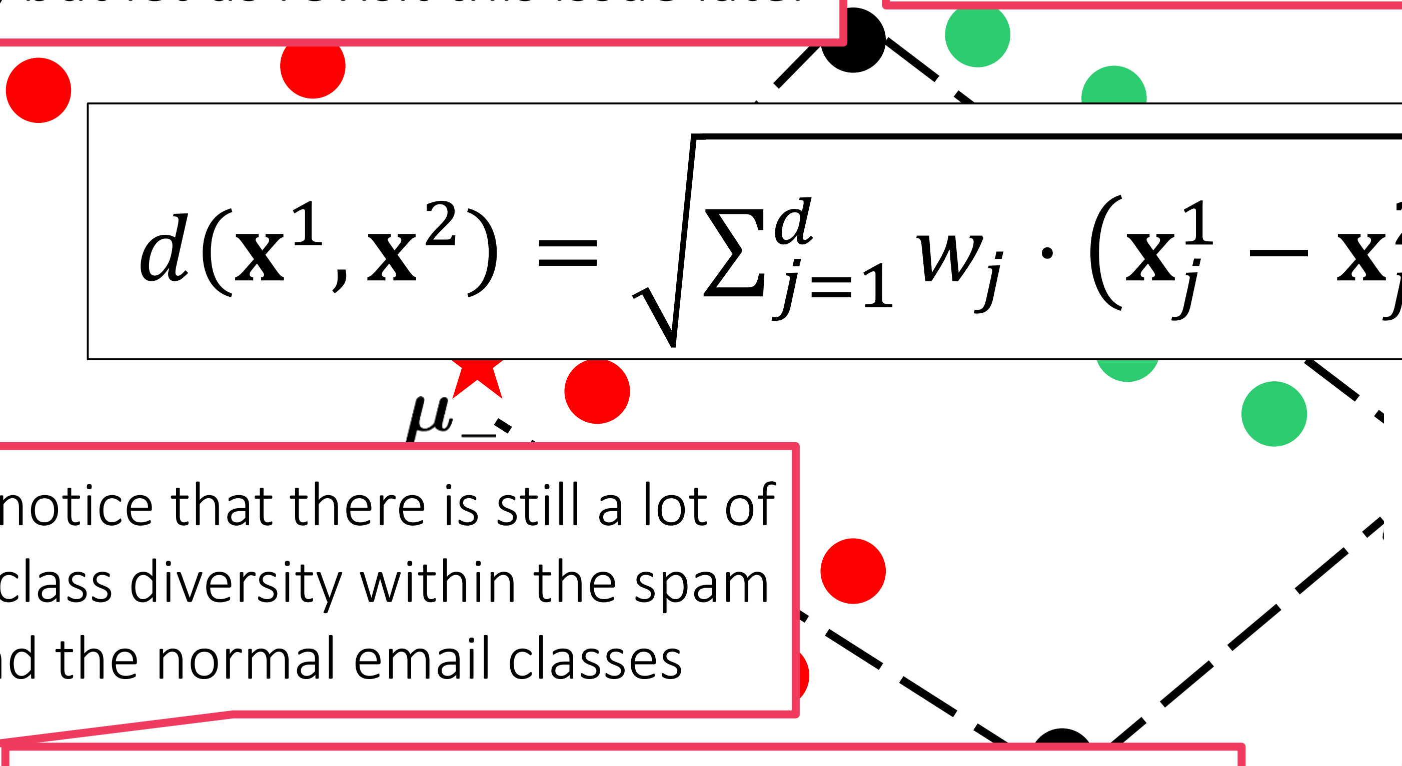
Learn… es - Issues

I think this time another issue is how we calculated distances. Can we use non-Euclidean distances?

Yes, but let us revisit this issue later

What went wrong this time?

Can we learn these weights too?



$$d(\mathbf{x}^1, \mathbf{x}^2) = \sqrt{\sum_{j=1}^{d} w_j \cdot \left(\mathbf{x}_j^1 - \mathbf{x}_j^2\right)^2}$$

Has wings

Can fly

Can sing

Has a beak

Well, notice that there is still a lot of intra-class diversity within the spam and the normal email classes

I think so too! E.g. we can give different weights to features while calculating distances

Learning with Prototypes - Lessons

An extremely simple technique to classify data

Can do binary classification (2-classes) as well as multi-classification

Very compact, light-weight model (one prototype per class)

Actually used in industrial applications like extreme classification

Actually state of the art if we have very few data points for a class

- For example, if we have very few spam emails in training data

Works well when class data points are packed closely – less diversity

Improvements possible using multiple prototypes, metric learning (using a non-Euclidean distance function)

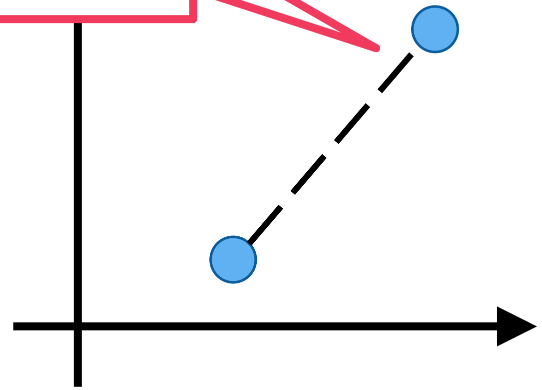
Making LwP more Powerful

The Euclidean distance is nice but ... eight

Also does not allow features to tal...

Euclidean distance does not change even if axes are rotated

E.g. $\|\mathbf{x}\|_2 = \sqrt{\sum_{j=1}^{d} \mathbf{x}_j^2}$ has no $\mathbf{x}_j\mathbf{x}_k$ term for $j \neq k$



Using a different distance function really helps

Metric learning: learn this distance function as well

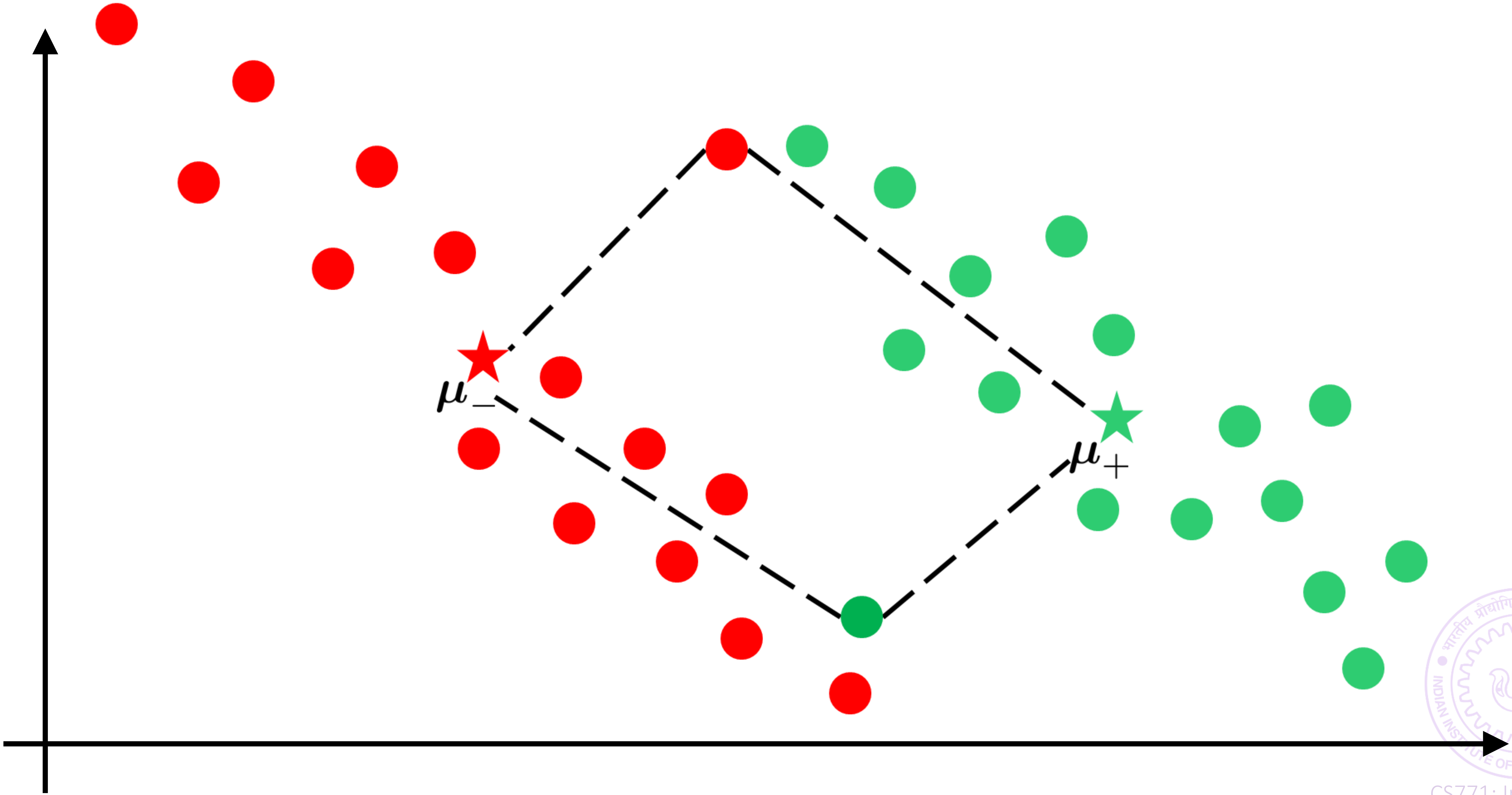
A very popular family of metrics – Mahalanobis metrics

Given a symmetric $d \times d$ matrix $A \in \mathbb{R}^{d \times d}$, we define a distance

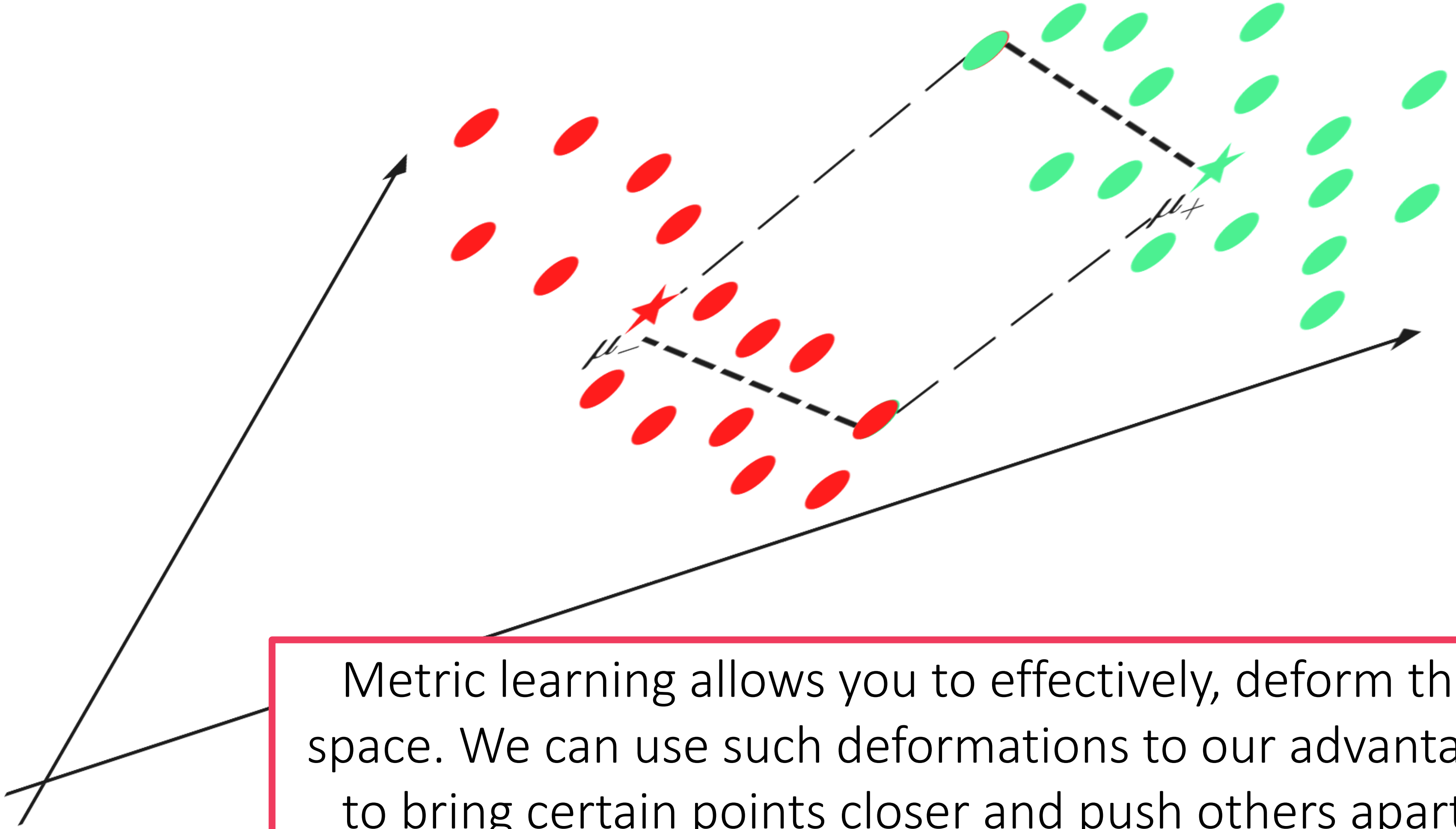
$$d_A(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top A (\mathbf{x} - \mathbf{y})}$$

Taking $A = I_d$ i.e. identity matrix, gives us the usual Euclidean distance

Why metric learning works



Why metric learning works



Metric learning allows you to effectively, deform the space. We can use such deformations to our advantage to bring certain points closer and push others apart.

LwP – behind the scenes

Let $\boldsymbol{\mu}^+, \boldsymbol{\mu}^-$ be the prototypes of the spa

Recall that we classify an email with feat

$$\|\mathbf{x} - \boldsymbol{\mu}^+\|_2 < \|\mathbf{x} - \boldsymbol{\mu}^-\|_2$$

$$\Leftrightarrow \|\mathbf{x} - \boldsymbol{\mu}^+\|_2^2 < \|\mathbf{x} - \boldsymbol{\mu}^-\|_2^2$$

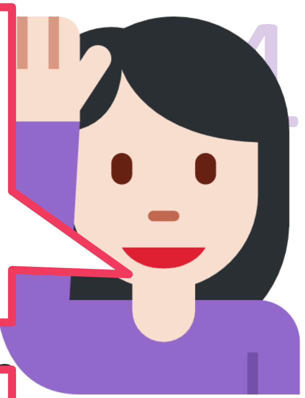
$$\Leftrightarrow \|\mathbf{x}\|_2^2 + \|\boldsymbol{\mu}^+\|_2^2 - 2\langle \mathbf{x}, \boldsymbol{\mu}^+ \rangle < \|\mathbf{x}\|_2^2 +$$

$$\Leftrightarrow \|\boldsymbol{\mu}^+\|_2^2 - 2\langle \mathbf{x}, \boldsymbol{\mu}^+ \rangle < \|\boldsymbol{\mu}^-\|_2^2 - 2\langle \mathbf{x}, \boldsymbol{\mu}^-\rangle$$

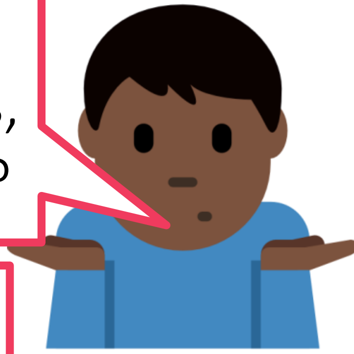
$$\Leftrightarrow \langle \mathbf{x}, 2(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-) \rangle + \|\boldsymbol{\mu}^-\|_2^2 - \|\boldsymbol{\mu}^+\|_2^2 >$$

$$\equiv \langle \mathbf{x}, \mathbf{w} \rangle + b > 0 \text{ with } \mathbf{w} = 2($$

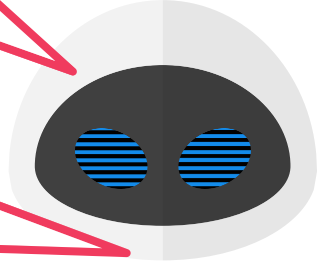
Classifiers with linear decision boundaries are called *linear classifiers*. Thus, LwP is a linear classifier



What happens if there are 2 or more prototypes per class, or else if there are 3 classes?

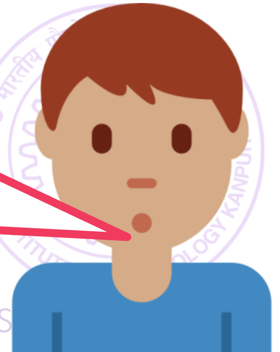


Think about this on your own – will discuss later



Yes, this is known as a *linear decision boundary*.

So the decision boundary of the LwP classifier is always a line or a hyperplane if the distance function is Euclidean!!



Linear/hyperplane Classifiers

The model is a single vector $\mathbf{w}$ of dimension d (features are also d-dim), and an optional scalar term (called *bias*) b

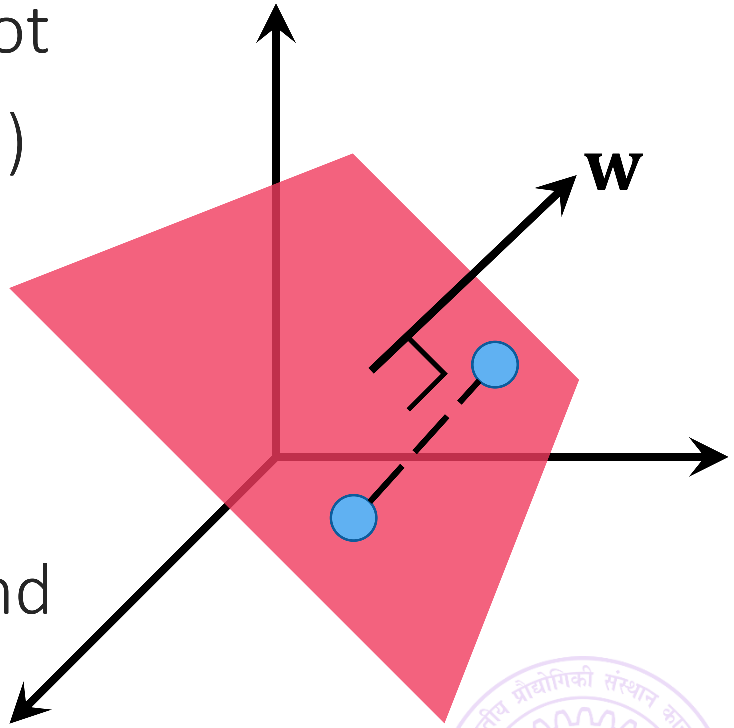
Predict on a test point $\mathbf{x}$ by checking if $\mathbf{w}^\top \mathbf{x} + b > 0$ or not

Decision boundary: line/hyperplane (where $\mathbf{w}^\top \mathbf{x} + b = 0$)

The vector $\mathbf{w}$ is called the *normal* or *perpendicular* vector of the hyperplane – why?

Consider any two vectors $\mathbf{x}, \mathbf{y}$ on the hyperplane i.e. $\mathbf{w}^\top \mathbf{x} + b = 0 = \mathbf{w}^\top \mathbf{y} + b$. This means $\mathbf{w}^\top (\mathbf{x} - \mathbf{y}) = 0$. Note that the vector $\mathbf{x} - \mathbf{y}$ is parallel to the hyperplane and $\mathbf{w}$ perpendicular to all such vectors

The bias term b if changed, shifts the plane – it can be thought of as a threshold as well – how large does $\mathbf{w}^\top \mathbf{x}$ have to be in order for us to classify $\mathbf{x}$ as spam etc!



To b or not to b – that is the question!

Trivia: the closest point (Euclidean distance) on the hyperplane to the origin is at a distance $|b|/\|\mathbf{w}\|_2$ from the origin – can you show why?

Sometimes, it is convenient to not have a separate bias term

Create another dim in feature vector and fill it with 1 i.e. $\tilde{\mathbf{x}} = [\mathbf{x}, 1]$

So now features (and model) are $d + 1$-dimensional

However, note that if we have a model $\tilde{\mathbf{w}} = [w_0, w_1, \dots, w_d] \in \mathbb{R}^{d+1}$ over the new features and if we denote $\mathbf{w} = [w_0, \dots w_{d-1}] \in \mathbb{R}^d$, then

$$\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} + w_d$$

Thus, w_d effectively acts as a bias term for us ☺

LwP with Mahalanobis metric still linear!!

A matrix A that satisfies a property called *positive semi-definiteness* (PSD has several other nice properties too

For all vectors $\mathbf{x}$, we must have $\mathbf{x}^\top A\mathbf{x} \geq 0$

$$d_A(\mathbf{x}, \boldsymbol{\mu}^+) < d_A(\mathbf{x}, \boldsymbol{\mu}^-) \Leftrightarrow 2\mathbf{x}^\top A(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-) + {\boldsymbol{\mu}^-}^\top A\boldsymbol{\mu}^- - {\boldsymbol{\mu}^+}^\top A\boldsymbol{\mu}^+ > 0$$

$$\equiv \langle \mathbf{x}, \mathbf{w} \rangle + b > 0 \text{ where } \mathbf{w} = 2A(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-), b = {\boldsymbol{\mu}^-}^\top A\boldsymbol{\mu}^- - {\boldsymbol{\mu}^+}^\top A\boldsymbol{\mu}^+$$

We can write $A = LL^\top$ where $L \in \mathbb{R}^{d \times d}$ (L need not be sym or PSD)

$$d_A(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top A(\mathbf{x} - \mathbf{y})} = \sqrt{(\mathbf{x} - \mathbf{y})^\top LL^\top(\mathbf{x} - \mathbf{y})}$$

$$= \|L^\top \mathbf{x} - L^\top \mathbf{y}\|$$

Nice! Thi
all $\mathbf{x}, \mathbf{y}$
dist

Oh! So the Mahalanobis distance is just Euclidean distance if we transform the vectors as $\mathbf{x} \mapsto L\mathbf{x}$