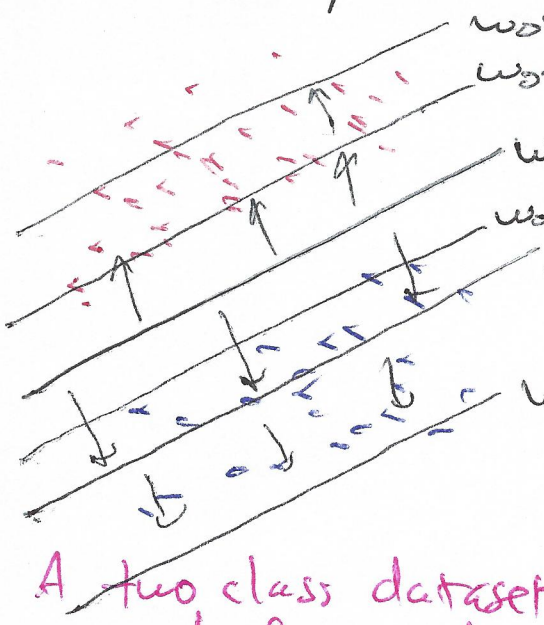
Linear multiclass classification

Some geometry with planes.



A two class dataset viewed from above

- If you slide a line/plane parallel like this one

Sliding the plane parallel up and down we can define the plane → where $\alpha > 0$

$$w_0+w_1x_1+w_2x_2=\alpha$$

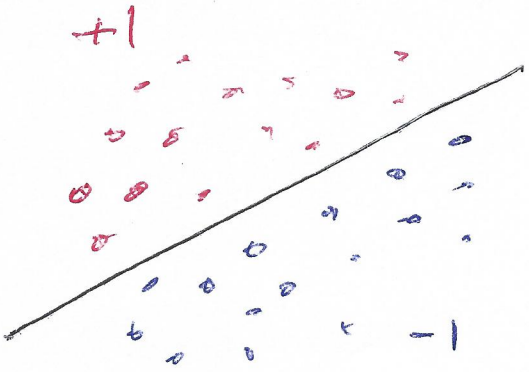
by shifting up and

$$w_0+w_1x_1+w_2x_2=-\alpha$$

by shifting down.

- the farther we shift, the larger (in absolute value) α becomes

Linear multi-class classification



$w_0 + w_1 x_1 + w_2 x_2 = 0$

or in compact notation

$\vec{w}^T \tilde{x} = 0$

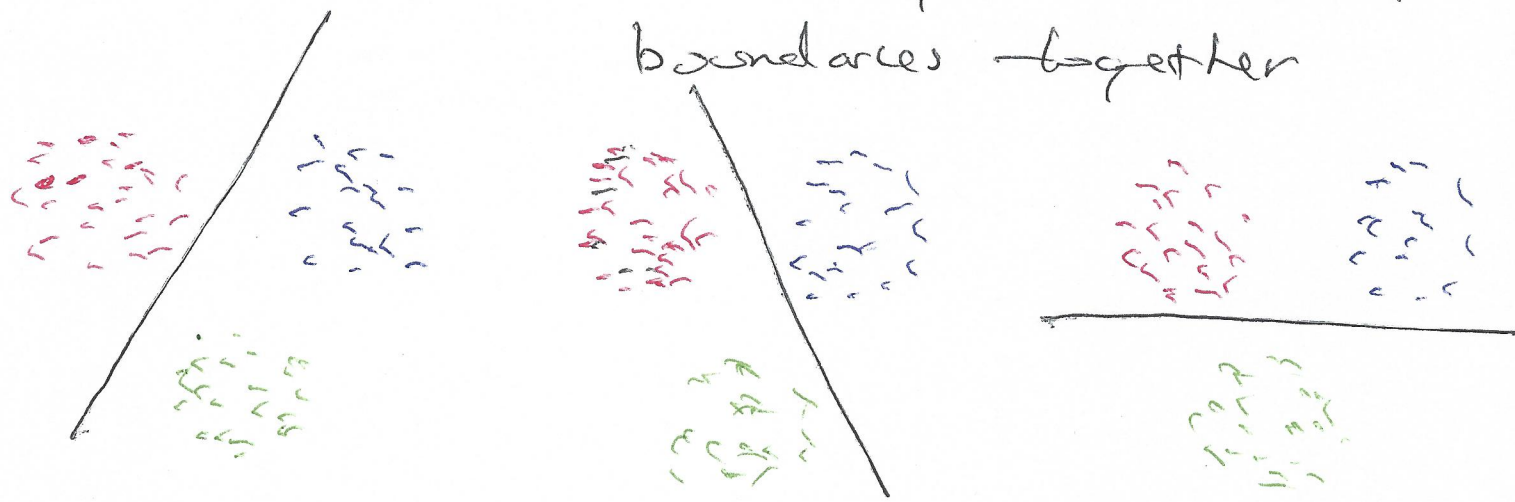
$\vec{w} = [w_0, w_1, w_2]^T$

2-class "from above"

$\tilde{x} = [1, x_1, x_2]^T$

- with > 2 class classification its more useful to use different labels

- how to generalize 2-class to multiclass?
- one of most popular / useful ways: reduce to several 2-class problems then fuse boundaries together

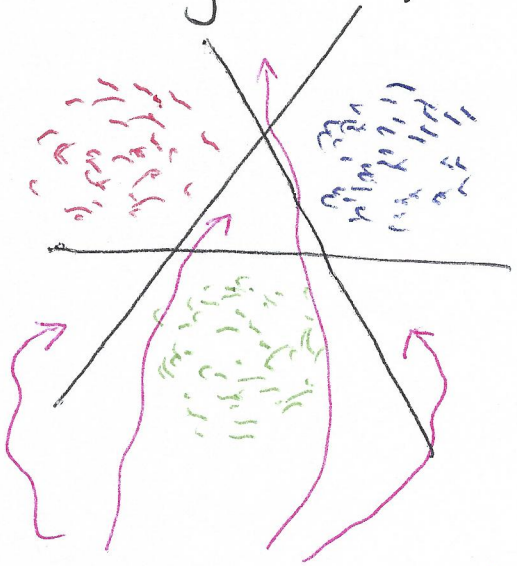


Subproblem 1
red vs all

subproblem 2
blue vs all

Subproblem 3
green vs all

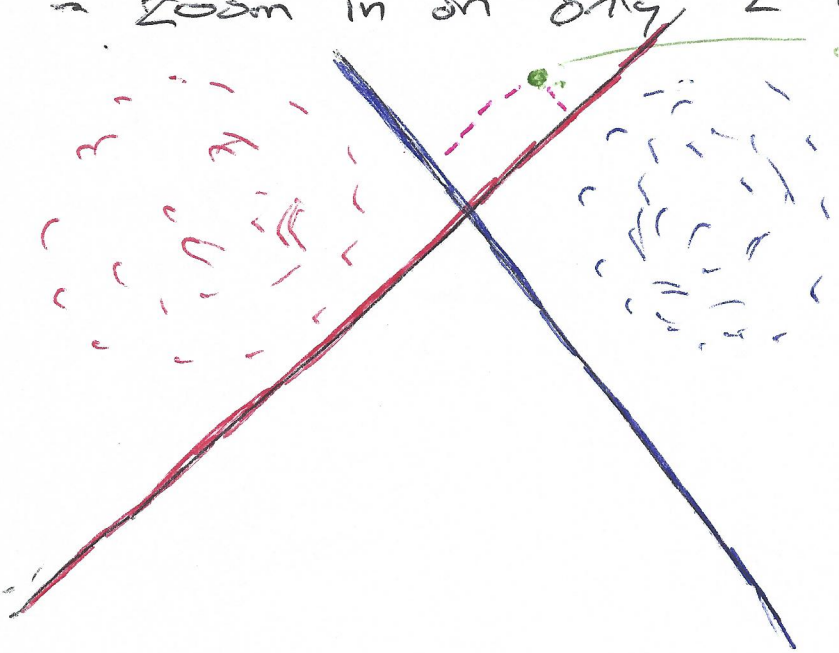
Each subproblem solved the usual way e.g. by logistic regression



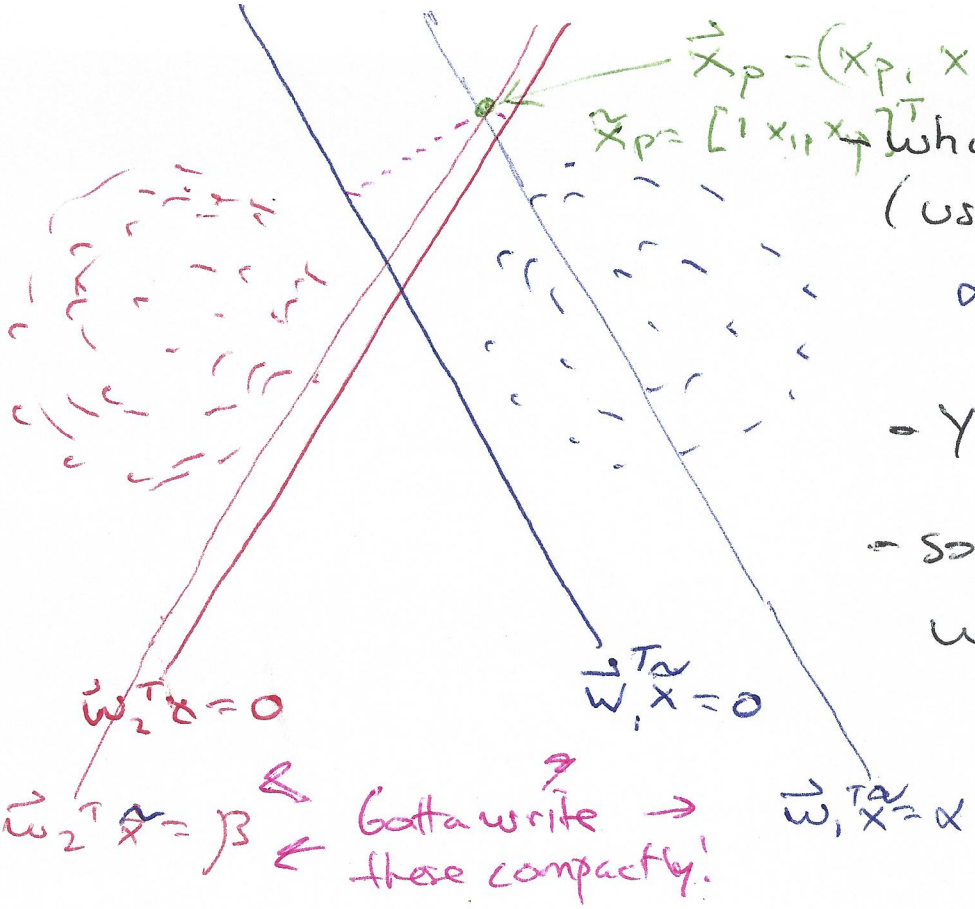
- we have C classes then C subproblems, C linear classifiers / linear boundaries
- but many regions have overlapping interest from more than 1 class

what do we do here?
which class to assign to?

- How to assign label to these points?
- Zoom in on only 2 classes + 2 boundaries



- which class should assign?
- Blue, because it's further into the blue territory than the red
- in other words, it's farther from the blue boundary than the red one.



Which is larger (using the picture) α or β?

- Yup, its α
- so which class did we assign too?

Gotta write these compactly!

- we said assign to $\underset{j=1,2}{\text{argmax}}\ \vec{w}_j^T \tilde{x}_p$

remember this notation

$\tilde{x}_p = [1\ x_{p,1}\ x_{p,2}]^T$

- this same logic works <u>every where</u> in the space, and regardless of # of classes. That is predicted label $\hat{y}_p$

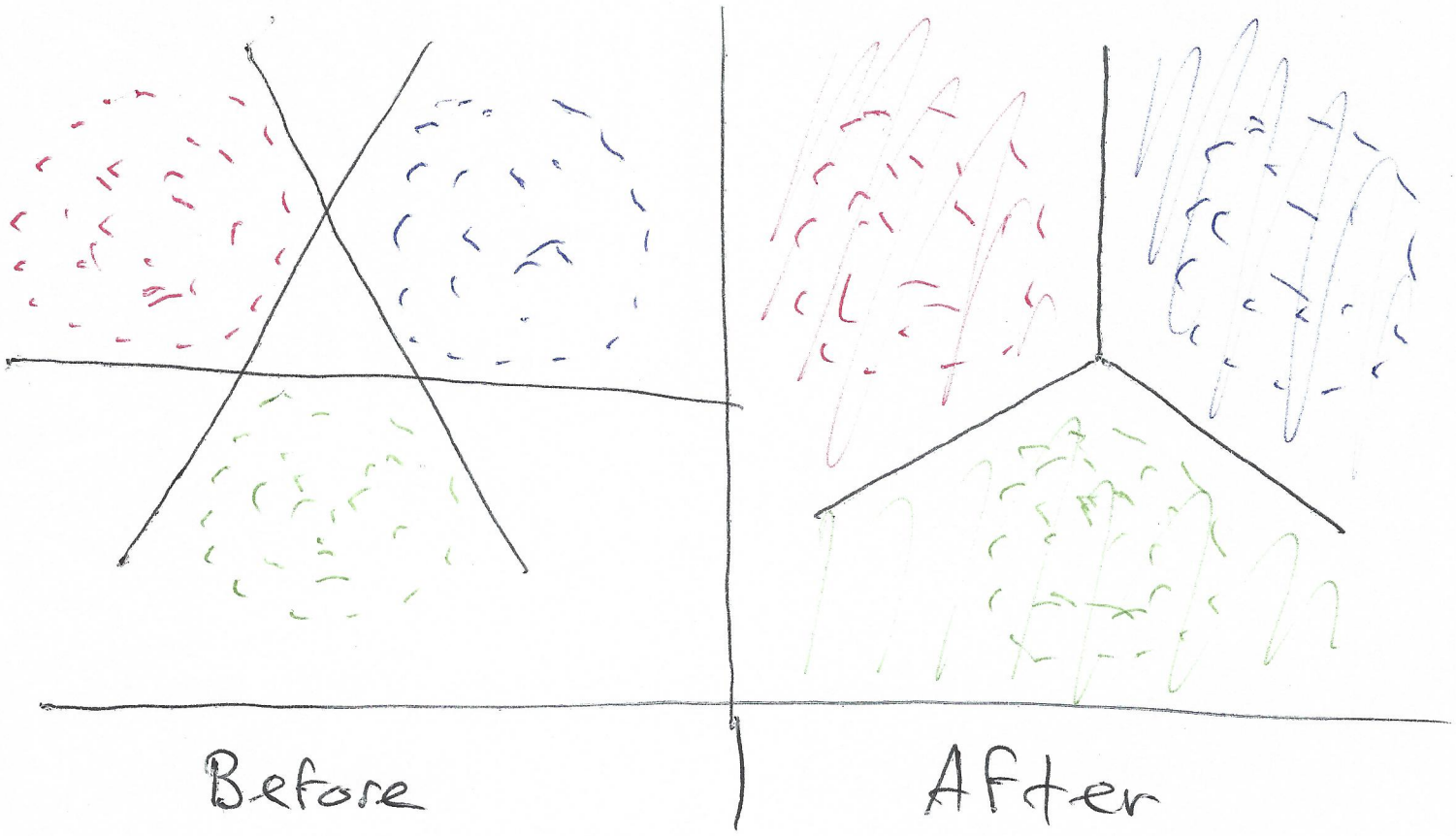
$$\hat{y}_p = \underset{c=1\ldots C}{\text{argmax}}\ \vec{w}_c^T \tilde{x}_p$$

← fusion rule

- this is called One-vs-all classification

- very closely related to "softmax multi-class" & "multiclass logistic regression" and gives similar results

- What happens to the rules after fusion?



- Dope!

So how do this algorithmically

- For each subproblem generate artificial labels specific to subproblem e.g. +1 for class, -1 for all else
- Solve all subproblems, giving weights $\vec{w}_1, \vec{w}_2 \ldots \vec{w}_C$
- to predict label of point $\vec{x}$

$$\hat{y} = \underset{c=1 \ldots C}{\operatorname{argmax}} \vec{w}_c^T \vec{x}$$