Machine Learning to build Intelligent Systems

Manas Dasgupta

Understanding Support Vector Machine (SVM)

Structure of this Module

Understanding Support Vector Machine

TOPICS
Introduction to Support Vector Machine
Linear SVM Classifications
Non-linear SVM Classification
Polynomial Kernel Trick

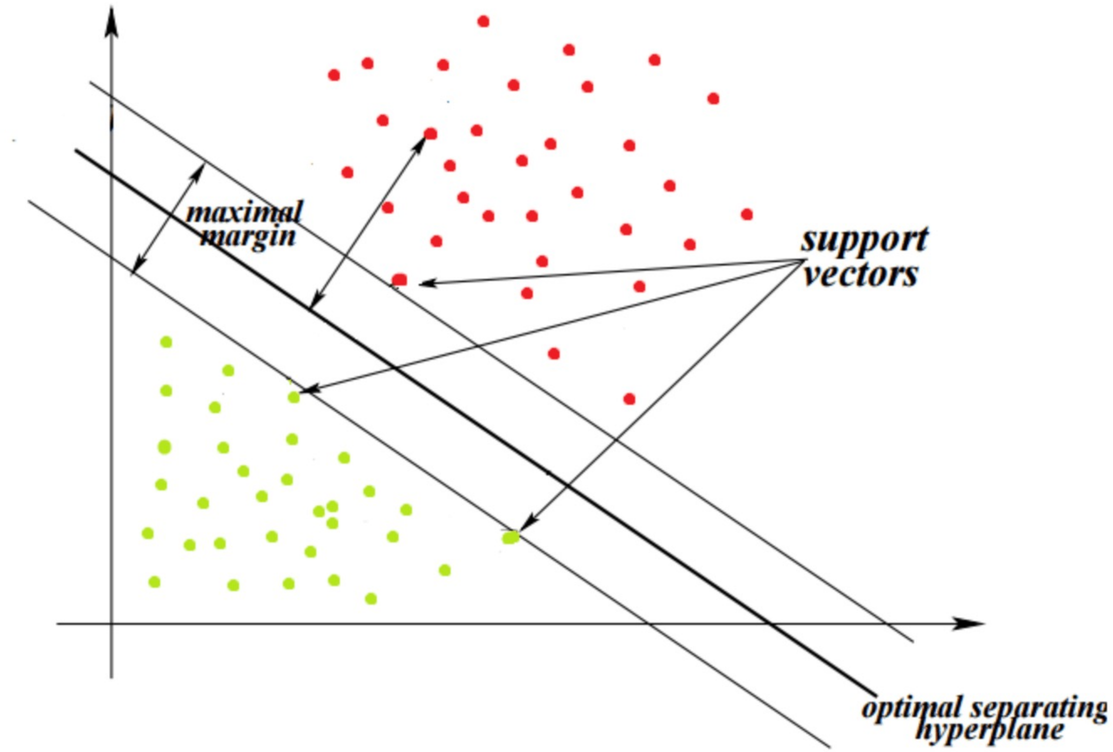
Introduction to Support Vector

A ***Support Vector Machine*** **(SVM)** is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification and regression.

It is one of the most popular models in Machine Learning, and is a must learn for anyone interested in Machine Learning.

SVMs are particularly well suited for classification of complex but small or medium-sized datasets.

Large Margin Classification



The SVM method can be explained with the help of the figure.

The data has two classes as noted by the colors and they are clearly linearly separable.

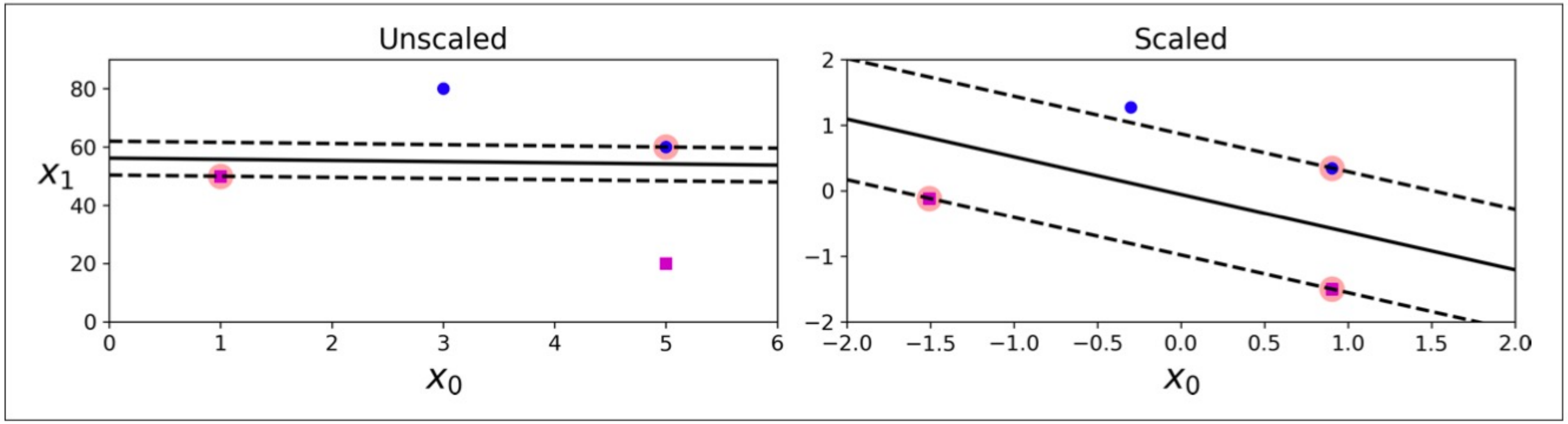
The goal of the SVM Classification challenge here will be to find the widest possible street between the classes.

This is called "**large-margin-classification**". The bold line in the middle is the "**decision boundary**".

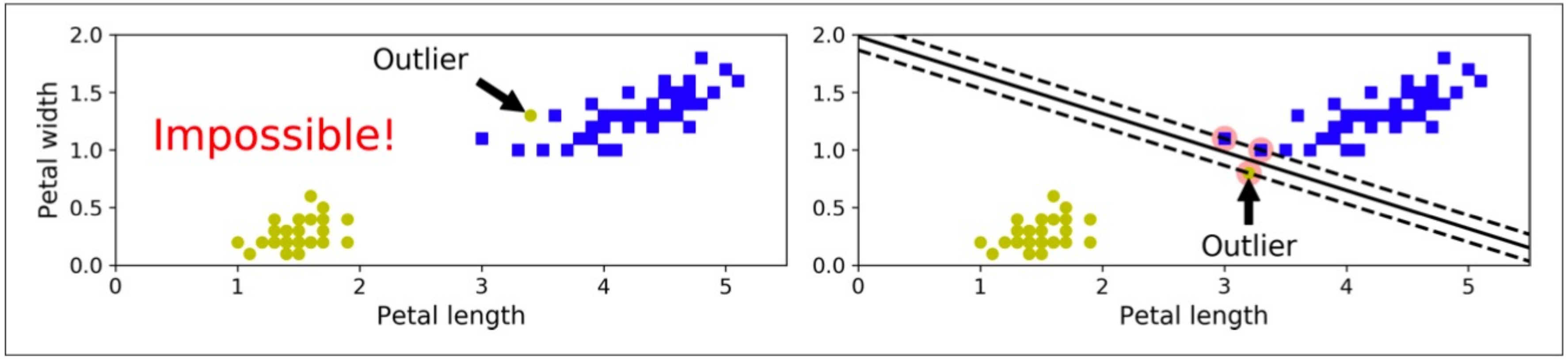
Notice that adding more training instances "off the street" will not affect the decision boundary at all: it is fully determined (or "supported") by the instances located on the edge of the street. These instances are called the **support vectors**.

Sensitivity to Scales

SVMs are sensitive to the feature scales, as you can see in Figure below: on the left plot, the vertical scale is much larger than the horizontal scale, so the widest possible street is close to horizontal.



Hard Margin Classification



If we strictly impose that all instances be off the street and on the correct side, this is called ***hard margin classification***. There are two main issues with hard margin classification.

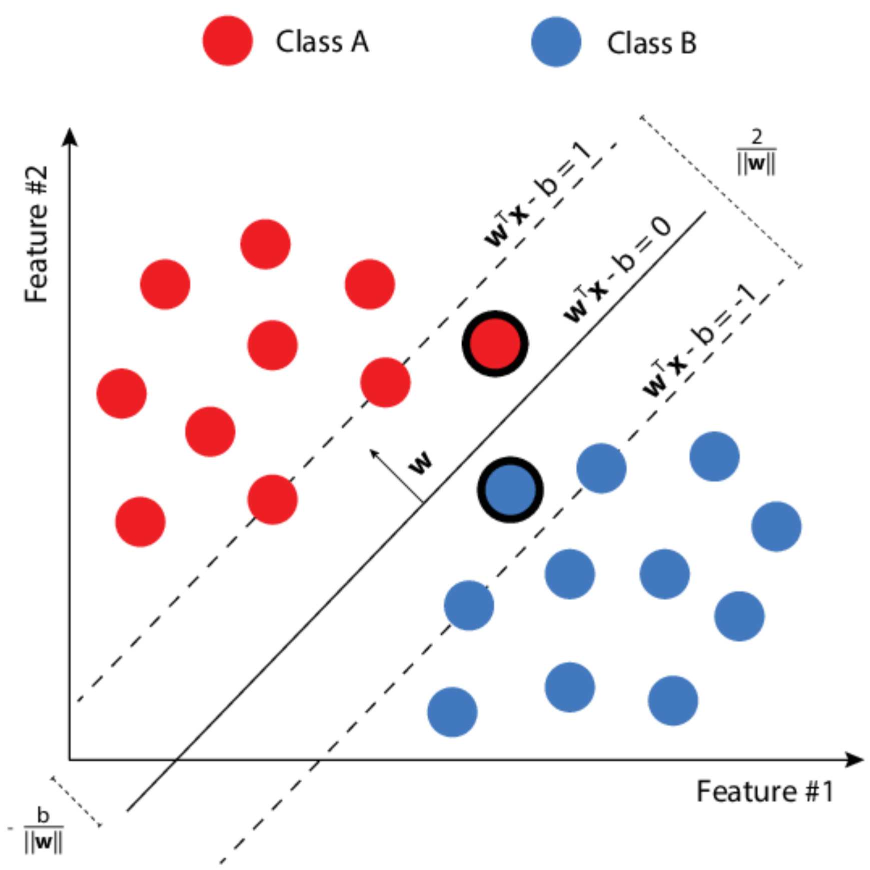
First, it only works if the data is linearly separable, and second it is quite sensitive to outliers.

Figure shows the iris dataset with just one additional outlier: on the left, it is impossible to find a hard margin, and on the right the decision boundary ends up very different from the one we saw in Figure without the outlier, and it will probably not generalize as well.

Soft Margin Classification

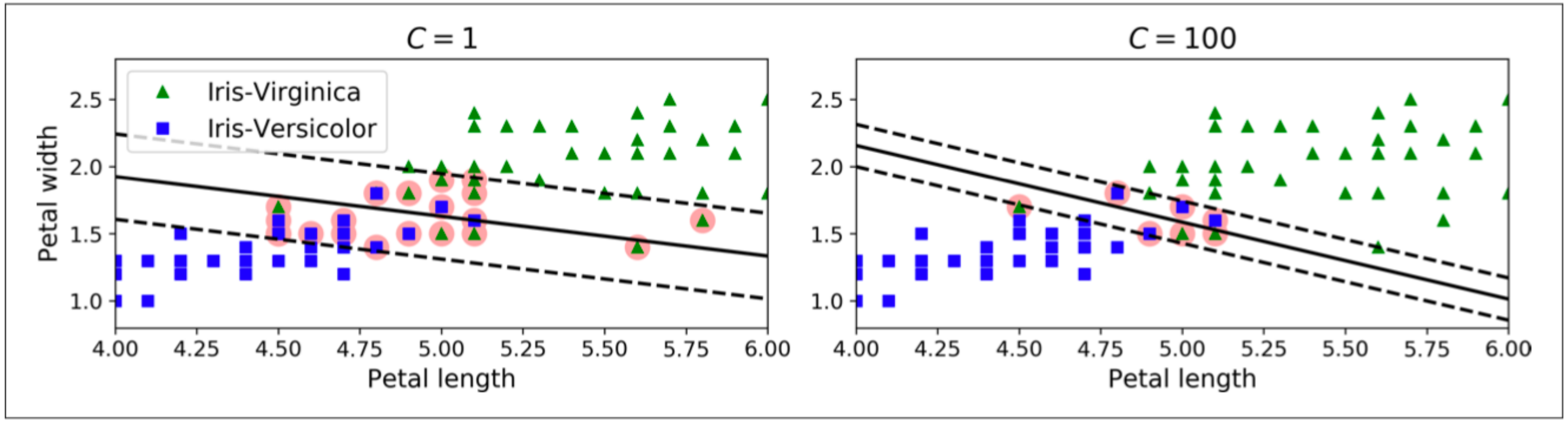
To avoid these issues it is preferable to use a more flexible model. ***The objective is to find a good balance between keeping the street as large as possible and limiting the margin violations*** (i.e., instances that end up in the middle of the street or even on the wrong side).

This is called ***soft margin classification***.



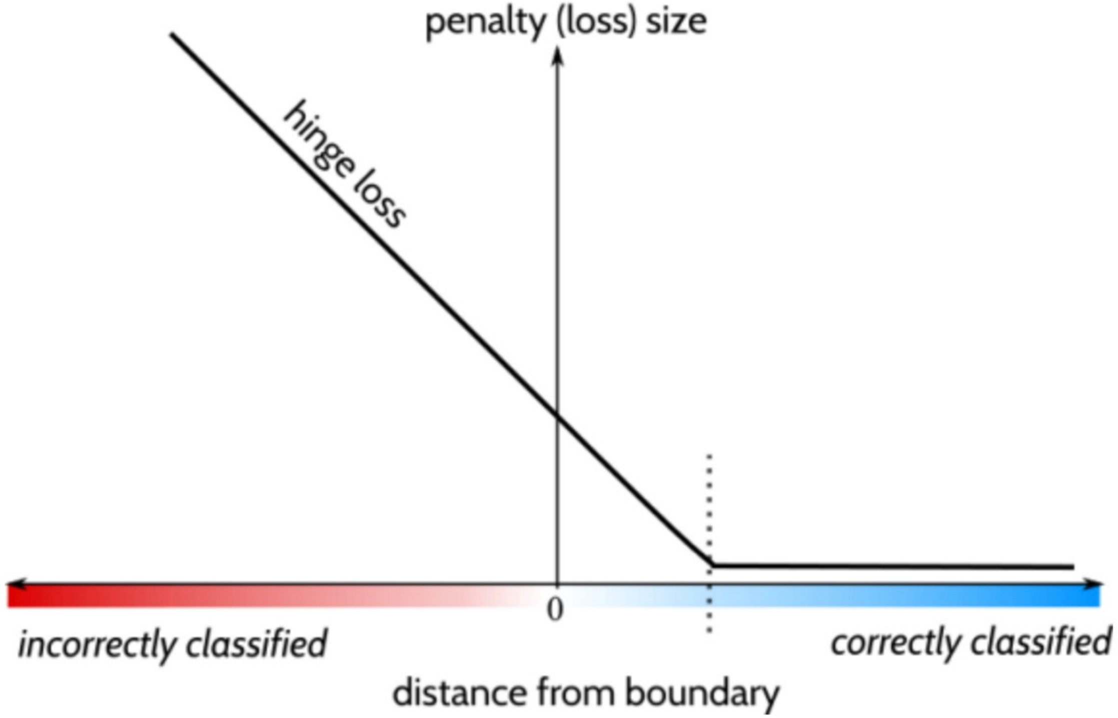
The ‘c’ Hyperparameter

- In Scikit-Learn’s SVM classes, you can control this balance using the ***‘c’ hyperparameter***: a smaller ‘c’ value leads to a wider street but more margin violations.
- The figure shows the decision boundaries and margins of two soft margin SVM classifiers on a nonlinearly separable dataset.
- On the left, using a ***low c value*** the margin is quite large, but many instances end up on the street.
- On the right, using a ***high c value*** the classifier makes fewer margin violations but ends up with a smaller margin.
- However, it seems likely that the first classifier will generalize better: in fact even on this training set it makes fewer prediction errors, since most of the margin violations are actually on the correct side of the decision boundary.



The Hinge Loss

The hinge loss is a loss function used for training Classifiers such as the SVM.



- The x-axis represents the distance from the boundary of any single instance, and the y-axis represents the amount of loss or penalty.
- **The dotted line on the x-axis represents 1. When an instance's distance from the boundary is greater than or at 1, the loss is 0.**
- If the distance from the boundary is 0 (meaning that the instance is literally on the boundary), then the loss size is 1.
- We see that correctly classified points will have a small (or none) loss size, while incorrectly classified instances will have a high loss size.
- A negative distance from the boundary incurs a high hinge loss. This essentially means that we are on the wrong side of the boundary, and that the instance will be classified incorrectly.
- On the other hand, a positive distance from the boundary incurs a low hinge loss, or no hinge loss at all, and the further we are away from the boundary (and on the right side of it), the lower our hinge loss will be.

The Hinge Loss

Let's look at an example numerically:

	actual	predicted	hinge loss
[0]	+1	0.97	0.03
[1]	+1	1.20	0.00
[2]	+1	0.00	1.00
[3]	+1	−0.25	1.25
[4]	−1	−0.88	0.12
[5]	−1	−1.01	0.00
[6]	−1	−0.00	1.00
[7]	−1	0.40	1.40

The Hinge loss separates negative and positive instances as +1 and -1, with -1 being on the left side of the boundary and +1 being on the right.

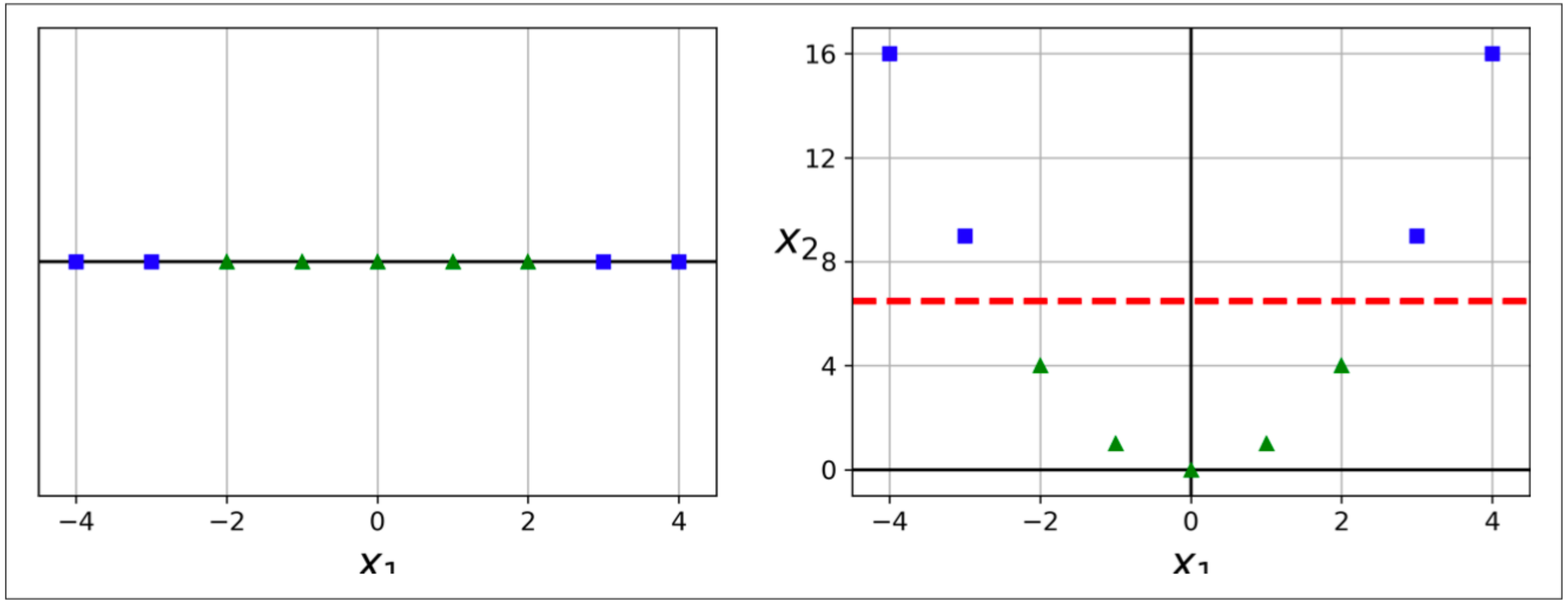
[0]: the actual value of this instance is +1 and the predicted value is 0.97, so the hinge loss is very small as the instance is very far away from the boundary.
[1]: the actual value of this instance is +1 and the predicted value is 1.2, which is greater than 1, thus resulting in no hinge loss
[2]: the actual value of this instance is +1 and the predicted value is 0, which means that the point is on the boundary, thus incurring a cost of 1.
[3]: the actual value of this instance is +1 and the predicted value is -0.25, meaning the point is on the wrong side of the boundary, thus incurring a large hinge loss of 1.25
[4]: the actual value of this instance is -1 and the predicted value is -0.88, which is a correct classification but the point is slightly penalised because it is slightly on the margin
[5]: the actual value of this instance is -1 and the predicted value is -1.01, again perfect classification and the point is not on the margin, resulting in a loss of 0
[6]: the actual value of this instance is -1 and the predicted value is 0, which means that the point is on the boundary, thus incurring a cost of 1.
[7]: the actual value of this instance is -1 and the predicted value is 0.40, meaning the point is on the wrong side of the boundary, thus incurring a large hinge loss of 1.40

Non-Linear SVM

Although linear SVM classifiers are efficient and work surprisingly well in most cases, many datasets are not even close to being linearly separable.

One approach to handling nonlinear datasets is to add more features, such as polynomial features . In some cases this can result in a linearly separable dataset.

Consider the left plot in Figure below. It represents a simple dataset with just one feature $x1$. This dataset is not linearly separable. However, if we add a second feature $\boldsymbol{x2 = (x1)^2}$, the resulting dataset is perfectly linearly separable.



Polynomial Kernel Trick

Adding polynomial features is a solution that can be used to solve classification challenges involving complex data.
However, a low polynomial transformation cannot deal with very complex datasets, and with a high polynomial degree it creates a huge number of features, making the model too slow.

Fortunately, when using SVMs we can apply a mathematical technique called the ***kernel trick***. It makes it possible to get the same result as if you added many polynomial features, even with very high-degree polynomials, without actually having to add them as features. So there is no combinatorial explosion of the number of features since you don't actually add any features.

This trick is implemented by the SVC class in Scikit-Learn.

SVM in Practice

Python Demo

Hope you have liked this Video.
Please help us by providing your Ratings and Comments for this Course!

Thank You!!
Manas Dasgupta

Happy Learning!!

Fully Editable Icon Sets: A

You can Resize without losing quality

You can Change Fill Color & Line Color

FREE

PPT TEMPLATES

www.allppt.com

Fully Editable Icon Sets: B

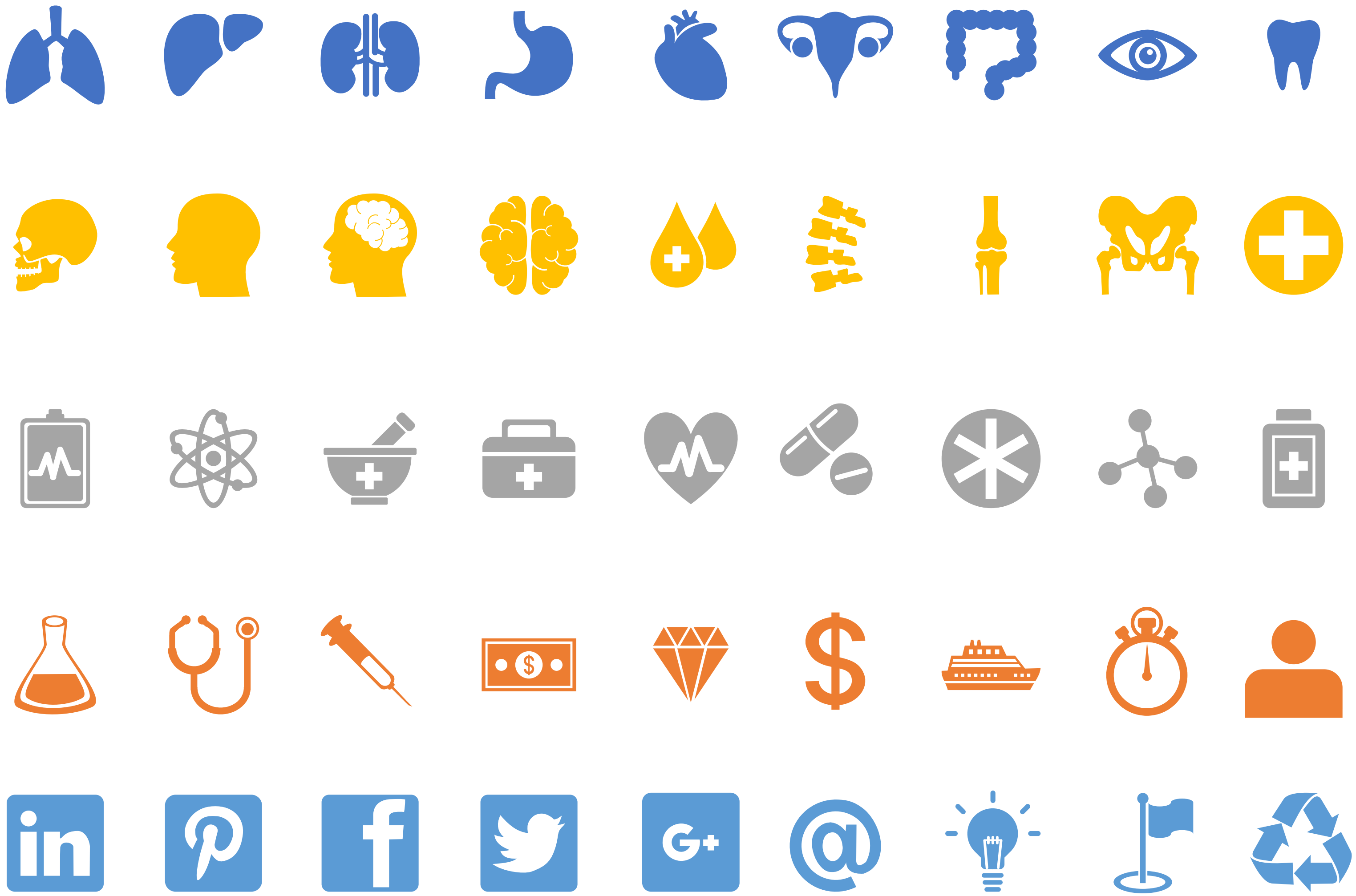
You can Resize without losing quality

You can Change Fill Color & Line Color

FREE

PPT TEMPLATES

www.allppt.com



Fully Editable Icon Sets: C

You can Resize without losing quality

You can Change Fill Color & Line Color

FREE

PPT TEMPLATES

www.allppt.com