When Should I Use SVC?

SVC is a classifier, so in short, you could use it on any classification problem. Other algorithms like K-Neighbors are instantaneous with training, but require you traverse a complicated tree structure for each sample you want to classify. That can become a bottle-neck in realtime applications like self driving cars that need to rapidly be able to tell the difference between a plastic bag and a large rock. One of the advantages of SVC is that once you've done the hard work of finding the hyperplane and its supporting vectors, the real job of classifying your samples is as simple as answering *what side of the line is the point on*? This makes SVC a classifier of choice for problems where *classification speed* is more critical than training speed.

SVC is extremely effective, even in high dimensional spaces. Just as you saw in the billiards explanation earlier, even after the instructor added in the rest of the balls onto the table, the accuracy of the original pool-stick classification was still *pretty good.* With SVC, most of your dataset actually doesn't even matter. The only important samples are those closest to the decision boundary, called the support vectors. Those samples determine the position of the separating hyperplane and the size of its margin. If you have a very large dataset consisting of many samples and want to speed it up simply by throwing away samples, a way to do so without sacrificing your classification accuracy too much would be by using SVC.

There may be cases where number your dataset has more features than the number of samples. Not all machine learning algorithms will be able to work with that, however such datasets aren't an issue for SVC. In fact, at least conceptually, if you use the kernel trick then at some point your data will almost assuredly be at a higher dimensionality than the number of features, depending on which kernel you use. And with the ability to use different kernel functions or even define your own, SVC will prove to be a very versatile classifier for you to have down in your machine learning arsenal.

Lastly, SVC is non-probabilistic. That means the resulting classification is calculated based off of the geometry of your dataset, as opposed to probabilities of occurrences. Once you get to decision trees, you'll see an example of a classifier that works using probabilities and not the geometric nature of your dataset.