The chapter introduces the concept of kernel methods, which provide a powerful way to extend many machine learning algorithms to operate on non-vectorial data objects like strings, trees, graphs, and other structured/unordered data types that are not easily represented as fixed-length feature vectors. The key idea is to define a kernel function $\kappa(x, x') \geq 0$ that measures the similarity or kernel between any two data objects x and x'. With an appropriate kernel function defined, many algorithms can then be "kernelized" to work with general data objects, just by replacing any inner product computations in the algorithm with calls to evaluate the kernel function on pairs of data objects.

Several examples of kernel functions defined for different data types are discussed. The radial basis function (RBF) or Gaussian kernel is a popular choice that measures similarity between vectors x and x' in Euclidean space based on their squared Euclidean distance scaled by a bandwidth parameter. For text document data, the bag-of-words kernel counts the number of shared words or character n-grams between two documents, with optional weighting schemes like TF-IDF. The string kernel more generally measures the number of shared substrings of all lengths between two string sequences. For comparing images, the pyramid match kernel computes a weighted histogram intersection between multi-resolution histograms of image feature sets.

Other kernel functions can be derived from generative probabilistic models. Probability product kernels evaluate the integral of the geometric mean of two distributions fit to the objects being compared. Fisher kernels instead use the derivatives of the data log-likelihood as features, weighted by the inverse Fisher information matrix.

To deploy kernels within generalized linear models like logistic regression for classification or regression, one simple approach is to define a kernelized feature map $\varphi(x) = [\kappa(x, \mu 1), ..., \kappa(x, \mu K)]$ that measures the similarity of x to a set of K centroid objects μk. The linear model can produce non-linear decision boundaries or regression functions in the original data space by replacing the original feature vector x with this kernel feature map $\varphi(x)$. Using all training instances themselves as the centroids μk yields an extremely high N-dimensional feature map, which can be made sparse using L1 regularization (L1VM), automatic relevance determination priors (RVM), or the support vector machine (SVM) framework.

The "kernel trick" provides an elegant way to modify many existing machine learning algorithms to operate on general kernel functions, rather than just vectorial data, simply by replacing any inner product computations $x^T x'$ with kernel evaluations $\kappa(x, x')$. However, this requires the kernel function κ to satisfy Mercer's condition of being a positive semi-definite kernel. Kernelized versions are derived for several fundamental algorithms like nearest neighbor classification, K-medoids clustering, ridge regression, and kernel PCA.

Kernel methods provide an elegant and principled way to extends linear algorithms designed for vector data to operate on much more general non-vectorial data objects, by implicitly mapping data into a high-dimensional feature space defined by the kernel function. This bypasses the need for extensive manual feature engineering on complex data types. However, the price paid is higher computational cost that scales with the number of training examples N, since kernelized algorithms essentially operate in an N-dimensional space. Therefore, it is highly desirable to have sparse kernel machines like L1VM, RVM, or SVM that only use a small subset of training examples as basis functions when making predictions.

Beyond having efficient training and prediction algorithms, the other key factor for good performance with kernel methods is choosing an appropriate kernel function that defines a suitable similarity measure between data objects for the problem at hand. The kernel function encodes assumptions about the patterns and structure in the
data.
Some kernels like the RBF kernel are fairly general and make no assumptions about the data, instead using a simple distance-based similarity measure controlled by a bandwidth parameter. Other kernels incorporate a great deal of prior knowledge - for example the string kernel is specifically designed to identify patterns of shared subsequences between biological sequence data. Fisher kernels tap into generative models of the processes that could have generated the data.
While kernel methods are widely and successfully used across many applications, some limitations are that they cannot naturally handle missing data or learn compositional models. There are also fundamentally difficult open problems in automatically learning optimal kernel functions from data in a computationally tractable way.
But overall, kernel-based learning algorithms are a tremendously useful and influential framework that has extended the applicability of many statistical machine learning techniques to a vast array of data types beyond just vectorial data. By defining a suitable similarity function, kernel methods allow operating on virtually any kind of data object, while preserving the convexity and optimizability of algorithms formulated in terms of inner products and distances. Their compositionality, modularity, and firm theoretical grounding have made kernel methods an indispensable tool for modern machine learning and data mining.