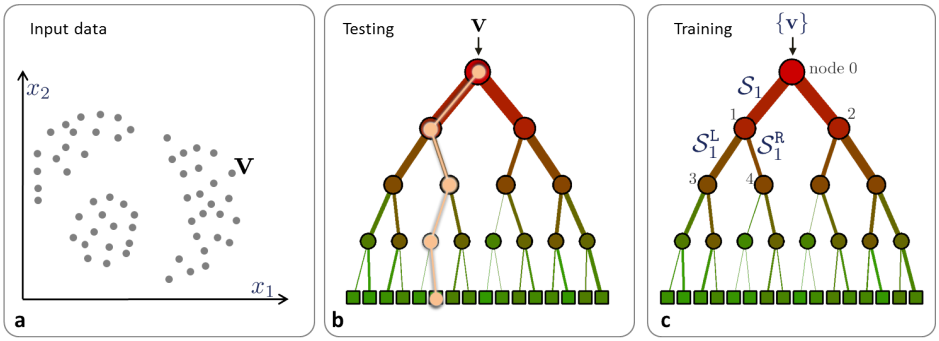

Fig. 2.2: **Basic notation. (a)** Input data is represented as a collection of points in the d-dimensional space defined by their feature responses (2D in this example). **(b)** A decision tree is a hierarchical structure of connected nodes. During testing, a split (internal) node applies a test to the input data $\mathbf{v}$ and sends it to the appropriate child. The process is repeated until a leaf (terminal) node is reached (beige path). **(c)** Training a decision tree involves sending all training data $\{\mathbf{v}\}$ into the tree and optimizing the parameters of the split nodes so as to optimize a chosen energy function. See text for details.

$\boldsymbol{\phi} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, with $d' << d$.

2.1.3 Training and testing decision trees

At a high level, the functioning of decision trees can be separated into an off-line phase (training) and an on-line one (testing).

Tree testing (runtime). Given a previously unseen data point $\mathbf{v}$ a decision tree hierarchically applies a number of predefined tests (see fig. 2.2b). Starting at the root, each split node applies its associated split function to $\mathbf{v}$. Depending on the result of the *binary* test the data is sent to the right or left child.[1] This process is repeated until the data point reaches a leaf node.

[1] In this work we focus only on binary decision trees because they are simpler than n-ary ones. In our experiments we have not found big accuracy differences when using non binary trees.