

- **Denoising:** In this type of task, the machine learning algorithm is given in input a *corrupted example* $\tilde{\mathbf{x}} \in \mathbb{R}^n$ obtained by an unknown corruption process from a *clean example* $\mathbf{x} \in \mathbb{R}^n$. The learner must predict the clean example \mathbf{x} from its corrupted version $\tilde{\mathbf{x}}$, or more generally predict the conditional probability distribution $p(\mathbf{x} \mid \tilde{\mathbf{x}})$.
- **Density estimation or probability mass function estimation:** In the density estimation problem, the machine learning algorithm is asked to learn a function $p_{\text{model}} : \mathbb{R}^n \rightarrow \mathbb{R}$, where $p_{\text{model}}(\mathbf{x})$ can be interpreted as a probability density function (if \mathbf{x} is continuous) or a probability mass function (if \mathbf{x} is discrete) on the space that the examples were drawn from. To do such a task well (we will specify exactly what that means when we discuss performance measures P), the algorithm needs to learn the structure of the data it has seen. It must know where examples cluster tightly and where they are unlikely to occur. Most of the tasks described above require the learning algorithm to at least implicitly capture the structure of the probability distribution. Density estimation allows us to explicitly capture that distribution. In principle, we can then perform computations on that distribution in order to solve the other tasks as well. For example, if we have performed density estimation to obtain a probability distribution $p(\mathbf{x})$, we can use that distribution to solve the missing value imputation task. If a value x_i is missing and all of the other values, denoted \mathbf{x}_{-i} , are given, then we know the distribution over it is given by $p(x_i \mid \mathbf{x}_{-i})$. In practice, density estimation does not always allow us to solve all of these related tasks, because in many cases the required operations on $p(\mathbf{x})$ are computationally intractable.

Of course, many other tasks and types of tasks are possible. The types of tasks we list here are intended only to provide examples of what machine learning can do, not to define a rigid taxonomy of tasks.

5.1.2 The Performance Measure, P

In order to evaluate the abilities of a machine learning algorithm, we must design a quantitative measure of its performance. Usually this performance measure P is specific to the task T being carried out by the system.

For tasks such as classification, classification with missing inputs, and transcription, we often measure the **accuracy** of the model. Accuracy is just the proportion of examples for which the model produces the correct output. We can