Data Input Format

In this document, we explain how the Julia implementation accepts data in input.

1 Preliminaries

First, we need to introduce some concepts for better readability. A *Kripke frame* $F = (\mathcal{W}, \mathcal{R})$ is a composed by a set of *worlds* $\mathcal{W}$ and a binary *accesibility relation* $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$, and a *Kripke model* $K = (F, V)$ is a Kripke frame enriched with a *valuation function* $V : \mathcal{W} \rightarrow 2^{\mathcal{AP}}$, which associates each world with a set of *propositional letters* from a set $\mathcal{AP}$ that are true on it. A *dimensional data set* $\mathcal{I} = \{I_1, \ldots, I_m\}$ is a finite collection of m *instances*, each of which is associated to a set of ℓ Kripke frames $\mathcal{F}_i = \{(F_i^j, R_i^j) \mid 1 \leq i \leq m \text{ and } 1 \leq j \leq \ell\}$ and each world of the j-th frame F^j is characterized by the value of n_i distinct attributes $\mathcal{A}^j = \{A_1^j, \ldots, A_{n_i}^j\}$.

Consider a clinical situation where each patient is described by a set of static attributes (e.g., age, sex), a multivariate time series (e.g., her fever and blood pressure evolution in time) and an image (e.g., a RGB image of her). Table 1 illustrates a typical, abstract view of an unlabeled data set where each instance is described by 3 (Kripke) frames: (a) a static frame, (b) a temporal frame, and (c) a spatial frame. Each frame is described by some attributes: (a) $\mathcal{A}^1 = \{A_1^1 = age, A_2^1 = sex\}$, (b) $\mathcal{A}^2 = \{A_1^2 = fever, A_2^2 = blood\_pressure\}$, and (c) $\mathcal{A}^3 = \{A_1^3 = red, A_2^3 = green, A_3^3 = blue\}$. Such instance has 3 Kripke frames $\mathcal{F} = \{(F^j, R^j) \mid 1 \leq j \leq 3\}$. For the sake of brevity, we omit the details on how such frames are defined.

In general, a data set is a collection of m instances as the one described earlier, where each instance can be labeled with some value (e.g., with classes for a classification task), also called supervision. The goal here is to explain the general file system structure of Figure 1. We assume that the reader is familiar with the CSV file format; specifically, the first row within a CSV-like file contains the names of the columns, and the remaining rows are the data contained in the file. At the root there is the name of the data set; in our example, for simplicity, the data set is called "Dataset", meaning that there is the directory `~/Dataset/` in the file system of the machine. Such directory must contain:

- `~/Dataset/Metadata.txt`, specifying the metadata file relevant to the entire dataset,
- `~/Dataset/Example` i`/`, a folder for the i-th example/instance, for $1 \leq i \leq m$,
- `~/Dataset/Labels.csv`, optional labels file if the data set is labeled.

The `~/Dataset/Metadata.txt` file is a properties file, where each row/line within the file stores a single property. The format for specifying properties is `key=value`, where `key` is the name of the property and `value` is the value for such property. Such properties can be viewed as a dictionary mapping `keys` to `values`, and this motivates the syntax for the properties. The required properties that `~/Dataset/Metadata.txt` must include are:

- `name=`$\langle name \rangle$, specifying the (string) name of the data set; for example, `name="Dataset"`,
- `supervised=`$\langle true \mid false \rangle$, specifying if the examples/instances are labeled; for example, `supervised=true` if there are labels,
- `num_frames=`$\langle \ell \rangle$, specifying the number ℓ of frames for each instance; for example, `num_frames=3` specifies that there are 3 frames, and
- `frame`$j = \langle 0 \mid 1 \mid 2 \mid \ldots \rangle$, specifying the dimension of the j-th frame, for $1 \leq j \leq \ell$; for example, `frame`$1 = 0$ specifies a static frame having 0 dimensions, `frame`$2 = 1$ specifies a temporal frame having 1 dimension, `frame`$3 = 2$ specifies a spatial frame having 2 dimensions, `frame`$4 = 3$ specifies a spatio-temporal frame having 3 dimensions, and so on.

If the data set is supervised (or labeled), then `~/Dataset/Labels.csv` must be defined as specified in Table 2. The labels can be either numeric or categorical. Categorical labels must be specified using a C-like string definition (i.e., using double quotes). It is important to note that there must be m

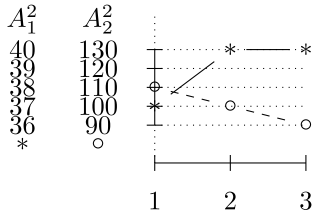
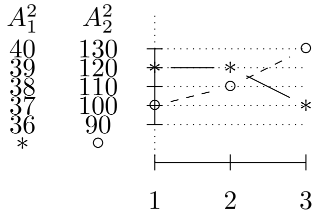
	Static Information	Temporal Information	Spatial Information
Patient 1	A_1^1 $\boxed{24}$ A_2^1 $\boxed{F}$		A_1^3 $\begin{array}{\|c\|c\|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline\end{array}$ A_2^3 $\begin{array}{\|c\|c\|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array}$ A_3^3 $\begin{array}{\|c\|c\|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array}$
Patient 2	A_1^1 $\boxed{27}$ A_2^1 $\boxed{M}$		A_1^3 $\begin{array}{\|c\|c\|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array}$ A_2^3 $\begin{array}{\|c\|c\|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline 0 & 1 \\ \hline\end{array}$ A_3^3 $\begin{array}{\|c\|c\|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array}$

Table 1: Abstract example of an unlabeled dataset with 2 instances.

of such labels for consistency with the number of instances in the data set. Finally, each `~/Dataset/Example` i`/` folder, for $1 \leq i \leq m$, contains the frames' information for the particular instance. Each such folder, must contain:

- `~/Dataset/Example` i`/Metadata.txt`, specifying the metadata for the particular example/instance,

- `~/Dataset/Example` i`/Frame` j`.csv`, a CSV file with the j-th frame information.

As before, `~/Dataset/Example` i`/Metadata.txt` contains the properties for the particular example:

- `dim`j`=`$\langle dimension \rangle$, specifying the dimension of the j-th frame, for $1 \leq j \leq \ell$; for example `dim`j`=()` specifies a static frame, `dim`$j = (n)$ specifies a temporal frame where each attribute is a time series having n point

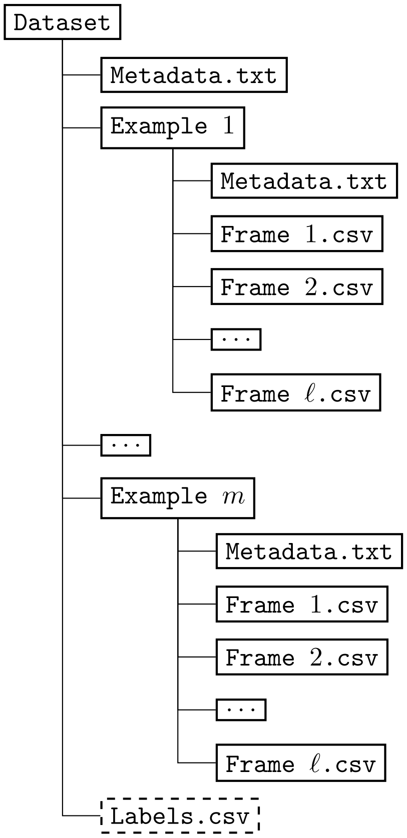


Figure 1: General file system structure of a data set.

Example	**Label**
1	0
2	1
m	0

Table 2: An example of ~/Dataset/Labels.csv data format.

observations, and dim$j = (h, w)$ specifies a spatial frame where each attribute is a $h \times w$ image.

2 Example

Now, we are interested in translating the abstract view of our example in Table 1 to the concrete data format presented in the previous paragraph. The `~/Dataset/Metadata.txt` has the following properties:

- `name="Dataset"`,
- `supervised=false`,
- `num_frames=3`,
- `frame1` $= 0$,
- `frame2` $= 1$, and
- `frame3` $= 2$.

From this, it follows that there is no `~/Dataset/Labels.csv` file. Having 2 instances, there are the following files:

- `~/Dataset/Example 1/Metadata.txt`,
- `~/Dataset/Example 1/Frame 1.csv`,
- `~/Dataset/Example 1/Frame 2.csv`,
- `~/Dataset/Example 1/Frame 3.csv`,
- `~/Dataset/Example 2/Metadata.txt`,
- `~/Dataset/Example 2/Frame 1.csv`,
- `~/Dataset/Example 2/Frame 2.csv`, and
- `~/Dataset/Example 2/Frame 3.csv`.

The properties in `~/Dataset/Example 1/Metadata.txt` are:

- `dim1` $= ()$, for the static frame,
- `dim2` $= (3)$, for the temporal frame having 3 time-points, and
- `dim3` $= (2, 2)$, for the spatial frame having 2×2 red, green, and blue images.

Age	**Sex**
24	*F*

Table 3: The `~/Dataset/Example 1/Frame 1.csv` data format for the first frame from Table 1.

Fever	**BloodPressure**
37	110
40	100
40	90

Table 4: The `~/Dataset/Example 1/Frame 2.csv` data format for the second frame from Table 1.

Red	**Green**	**Blue**
0	1	0
1	0	1
0	1	1
1	0	0

Table 5: The `~/Dataset/Example 1/Frame 3.csv` data format for the third frame from Table 1.

On the other hand, the properties in `~/Dataset/Example 2/Metadata.txt` are:

- `dim1` $= ()$, for the static frame,
- `dim2` $= (3)$, for the temporal frame having 3 time-points, and
- `dim3` $= (3, 2)$, for the spatial frame having 3×2 red, green, and blue images.

The `~/Dataset/Example 1/Frame 1.csv` is described in Table 3, which is what is to be expected. The `~/Dataset/Example 1/Frame 2.csv` is described in Table 4, that represents the multivariate time series (i.e., $|\mathcal{A}^2| > 1$) into its linearized left-to-right version with $n = 6$. Finally, in Table 5 the `~/Dataset/Example 1/Frame 3.csv` is described, that represents the linearized left-to-right and top-to-bottom version with $(h, w) = (2, 2)$, meaning that there are $2{\times}2$ rows for each attribute. Similarly, the `~/Dataset/Example 2/Frame 1.csv`, `~/Dataset/Example 2/Frame 2.csv`, and `~/Dataset/Example 2/Frame 3.csv` files are described in Tables 6, 7 and 8, respectively.

Age	**Sex**
27	M

Table 6: The `~/Dataset/Example 2/Frame 1.csv` data format for the first frame from Table 1.

Fever	**BloodPressure**
39	100
39	110
37	130

Table 7: The `~/Dataset/Example 2/Frame 2.csv` data format for the second frame from Table 1.

Red	**Green**	**Blue**
1	0	1
0	1	0
1	0	0
0	1	1
1	0	1
0	1	0

Table 8: The `~/Dataset/Example 2/Frame 3.csv` data format for the third frame from Table 1.

We include a ZIP file with the example to guide the user in generating such data input format.