MANUAL:

RAPIDMINER: ADVANCED CHARTS

rapid-i
REPORT THE FUTURE

RapidMiner 5.2: Advanced Charts

Manual

Rapid-I
www.rapid-i.com

Contents

1 Introduction

As of release 5.2, RapidMiner features a new visualization module, *RapidMiner's Advanced Charts*. This module has been developed as an alternative to the well known *Plot View* from previous releases and is planned to replace the old view completely in future releases.

The new module allows you to create, combine and overlay a variety of charts. Data can be grouped and aggregated directly during the creation of the chart, such that many complex data transformations with RapidMiner operators can be omitted.

As opposed to the Plot View, with the Advanced Charts the chart type is not configured statically at the beginning, but can be changed on the fly at any time of the chart creation process. All this can be done comfortably from within a flexible, easy to use graphical user interface, which is completely integrated into RapidMiner. It is described in chapter 2: *Graphical User Interface*. In chapter 3: *Creating Line and Shape Charts* the reader will learn how to create charts composed from lines and shapes, such as scatter plots or time series. The following chapter 4: *Creating Bar Charts* describes the creation of bar charts and introduces grouping and aggregation techniques. Chapter 5: *Formatting the Chart* shows how a chart can be formatted and customized in terms of colors, labels, titles etc.

After reading these chapters, the reader is capable of creating custom charts and using the most important features of RapidMiner's Advanced Charts. In chapter 6: *Advanced Plotting* more advanced topics are approached. Section 6.1 explains how ranges and data filters are defined. The creation of a multi-series chart which is composed of several, overlaying charts is treated in section 6.2. Section 6.3 details the grouping and aggregation mechanism, and section 6.4 demonstrates how to add error bars and other indicators to the chart. Cumulative charts and charts of the moving average are covered in section 6.5.

A final wrap-up of almost all previously learned techniques applied to a single chart is given in the last chapter 7: *Wrap-up: Creating a Lift Chart*.

The chapters are best read in order, since they often refer to techniques explained earlier and in that case don't explain the necessary steps in detail. However, the impatient reader may skip chapter 3 or 4 after reading chapter 2 and proceed directly to chapter 5 (if he is prepared to not understand some details). Before approaching chapter 6 the reader should be familiar to all the basic techniques, whereas the section in that chapter can be read in any order. The wrap-up chapter 7 assumes that the reader has worked through all other chapters.

2 Graphical User Interface

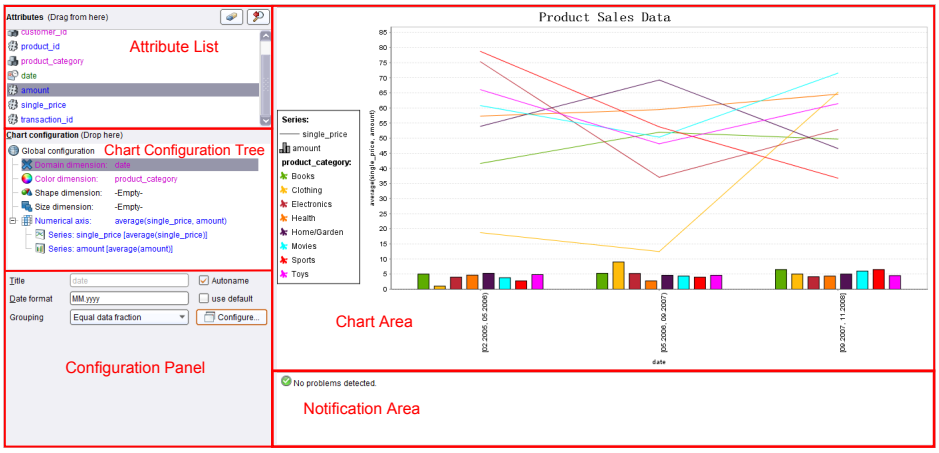

Figure 2.1: Graphical User Interface Overview

In this chapter we will have a look at the Graphical User Interface of the *Advanced Charts* view. It is divided into five major parts (see fig. 2.1) that help you to configure your chart the way you want it to be. All panels on the left side are used to configure the chart, whereas the *Notification Area* on the lower right side is used to display notifications regarding your chart configuration, e. g. errors and warnings, and the *Chart Area* is used to display the configured chart. Since the two panels on the right side are used to display charts and notifiactions only and thus are self-explanatory we only will have a closer look on the leftmost panels.

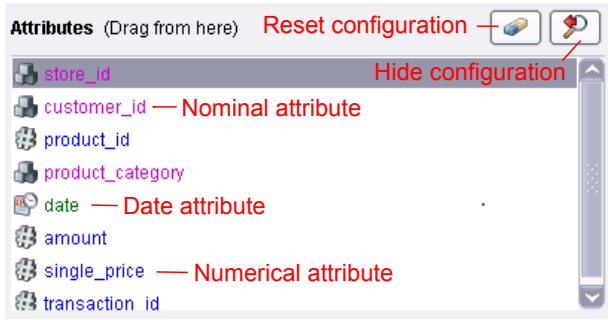

Figure 2.2: The Attribute List.

2.1 Attribute List

The first panel in the upper left corner is the *Attribute List* (see fig. 2.2). In this list all available attributes that can be used to configure the chart are shown. The value type of the attributes can easily be distinguished by color and icon. Nominal attributes are represented by the color purple and the icon . Numerical attributes are shown in a blue color and with the icon , whereas date attributes have the icon and a green color. Furthermore directly above the *Attribute List* there are two additional buttons. The button on the right hides all chart configuration panels and the *Notification Area* thus showing only the *Chart Area* afterwards. The button on the left can be used to reset your so far configured chart and start over again.

2.2 Chart Configuration Tree

To configure a chart the *Chart Configuration Tree* (see fig. 2.3) is the most important panel, where the whole configuration for your chart's data can be seen at a glance. The root node is the *Global configuration* node. Directly below the root node there are four different dimension configurations, particularly the *Domain dimension*, the *Color dimension*, the *Shape dimension* and the *Size dimension*. These dimension configurations can be used to group your data and to automaticly provide different colors, shapes or sizes for your series. A special role has the *Domain dimension*. It's attribute determines the values that will be shown on the x axis. Furthermore it provides the domain grouping for all grouped series.

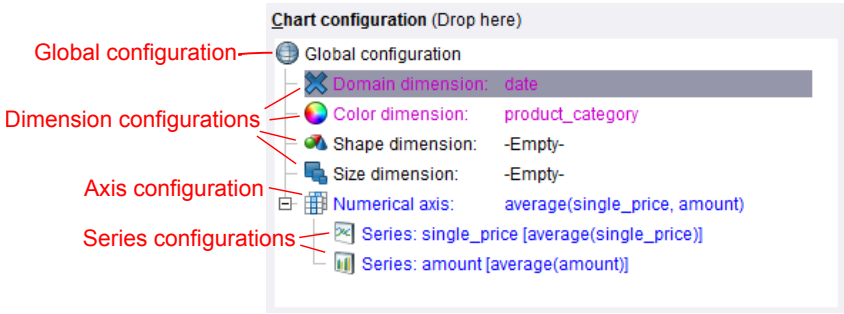

Figure 2.3: The Chart Configuration Tree.

On the same tree level as the dimension nodes there are one or more axis configurations. An axis configuration represents the configuration for an axis in the chart and consists of an abritrary number of series configurations. A series configuration represents a series that will be placed on its parent axis. The visualization type of a series can be distinguished by its configuration node's icon. Bar charts are shown with  , area charts with  and line and shape charts with  . Furthermore note that all dimension, axis, and series configurations are colored in their particular value type's color. This means for example that dimensions with a nominal attribute or a categorical grouping are colored purple, whereas dimensions with a numerical attribute or numerical grouping are shown in a blue color. The same logic applies to axis and series configurations. But there is an exception: In case of errors or warnings the configuration tree nodes are colored red for an error or yellow for a warning thus helping you to quickly find configuration problems.

2.2.1 Interacting with the Configuration Tree

There are several possibilities to interact with the *Chart Configuration Tree*. Most of the time you will start to configure a chart by dragging attributes from the *Attribute List* onto the Tree. When hovering over a drop target tooltips are shown that give you a clue what will happen when you drop an attribute or why a dropping is not possible (see fig. 2.4). Another possibility to interact with the tree is to right-click on tree elements thus invoking a context sensitive popup menu. This way it is for example possible to remove attributes from dimensions or delete entire axes (see fig. 2.5). Moreover it is possible to adjust the current configuration by left-clicking on the tree nodes and configuring them afterwards

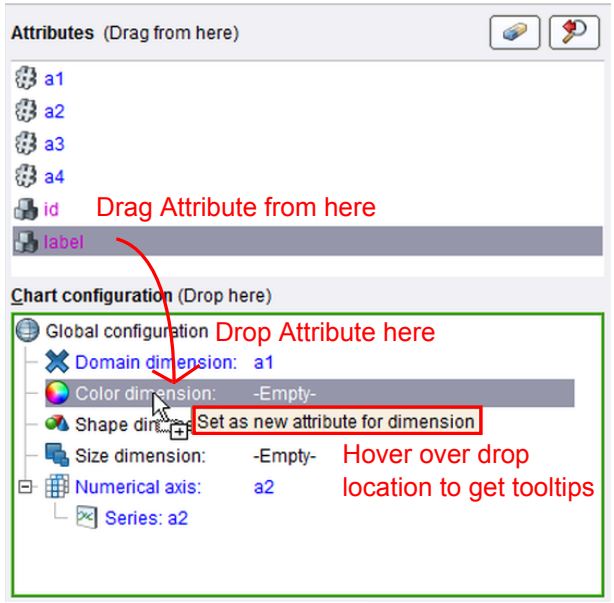

Figure 2.4: Dragging attributes from *Attribute List* on *Chart Configuration Tree*.

in the configuration panel, which is described in the next section.

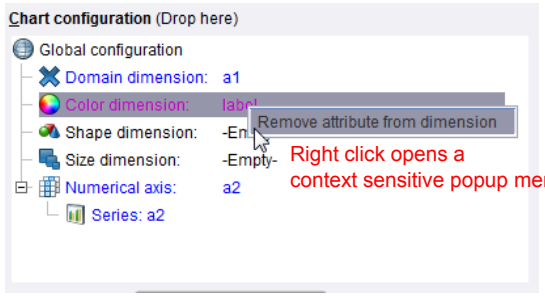

Figure 2.5: Right click on tree nodes opens popup menu.

2.3 Configuration Panel

There are four different types of configuration panels – for every kind of *Chart Configuration Tree* node a different one. The panels are selection dependent and can be viewed by selecting nodes in the *Configuration Tree* (see fig.2.6). The

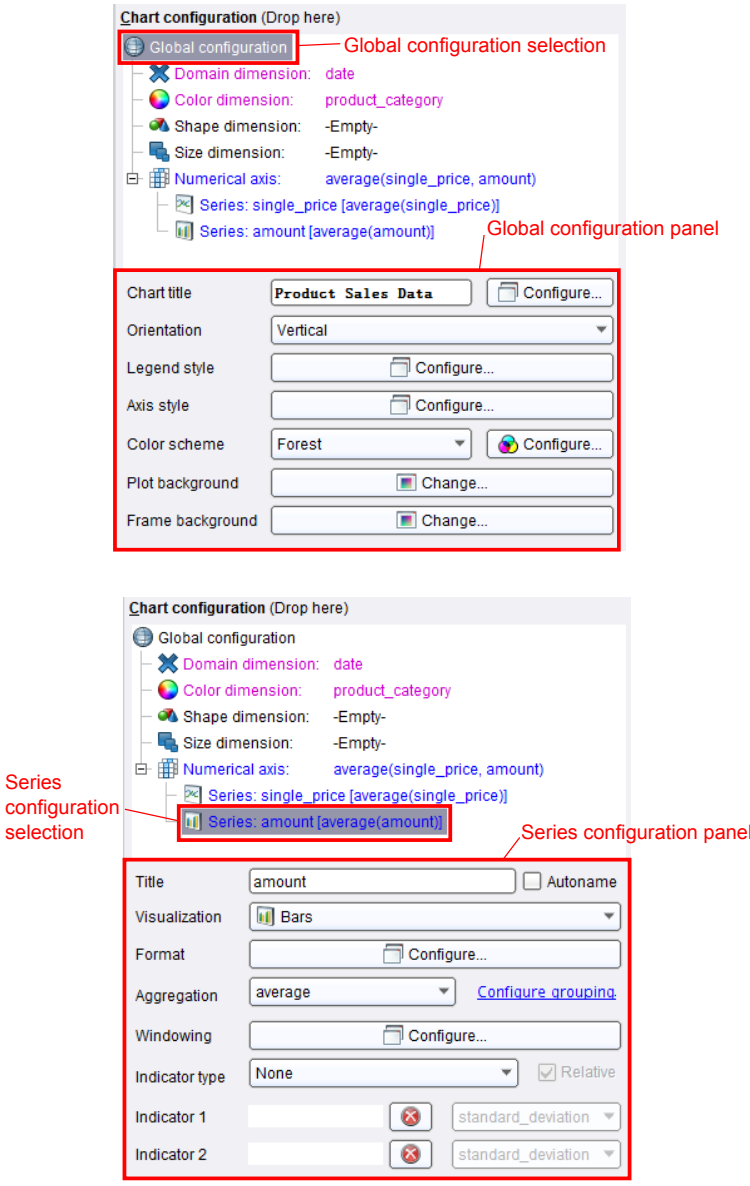

Figure 2.6: Selection dependent configuration panels.

Global configuration panel allows you to specify global settings like used color scheme, the legend style, etc. See 5.3 for a closer view on this topic. By selecting the Domain, Color, Shape or Size Dimension Configurations it is possible to configure settings like label, filter ranges (see 6.1), grouping (see 6.3), etc. for each dimension. The Axis Configuration panel allows you to adjust settings like view range on axis, logarithmic scaling and label, whereas the Plot Configuration panel contains the biggest number of adjustable options like label, series type, format (see 5.2), aggregation, indicator type (see 6.4) and windowing (see 6.5).

3 Creating Line and Shape Charts

3.1 A Simple Scatter Chart

First of all we will start with creating a simple scatter chart with the *Iris* dataset, which can be found in the samples repository of RapidMiner. After opening the *Advanced Charts* view we will see an empty chart configuration which has an error sign shown beside the *Domain dimension* and a warning signs shown beside the "Empty axis". A look at the *Notification Area* gives us a clue what has to be done first to configure our first chart (see fig. 3.1).

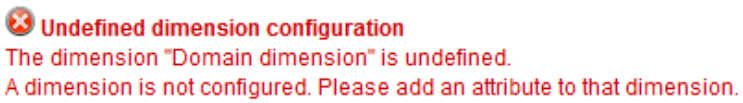

Figure 3.1: Empty domain dimension error.

The error states that the Domain dimension has no attribute assigned but requieres an attribute. The reason for this is that the Domain dimension's attribute defines the values that will be shown on the x axis, thus without an attribute on this dimension plotting a chart is not possible. Hence our first action will be to drag the attribute $a1$ from the *Attribute List* onto the domain dimension (see fig. 3.2). Afterwards the error sign besides the Domain dimension will be gone but our chart is still empty and yet another error is shown in the *Notification Area*. This time it states that there is no valid axis defined and without a valid axis it is, again, not possible to draw a chart. As you can see in the *Chart Configuration Tree* there is an axis with a warning icon that reminds us that this axis is not being used because it is empty. To resolve the error we have to fill the empty axis. This can be done by dragging the attribute $a2$ onto the axis (see

fig. 3.3). After doing this all errors will be gone and our first scatter chart will be drawn. Congratulations!

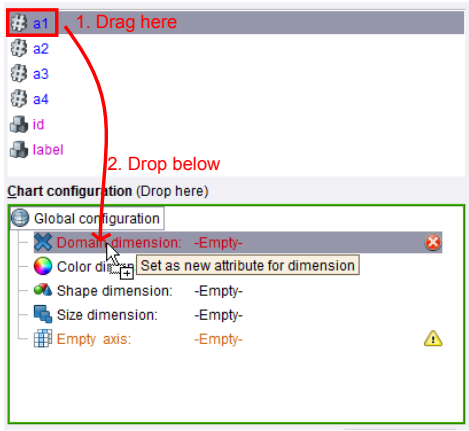

Figure 3.2: Drop Attribute $a1$ on Domain dimension.

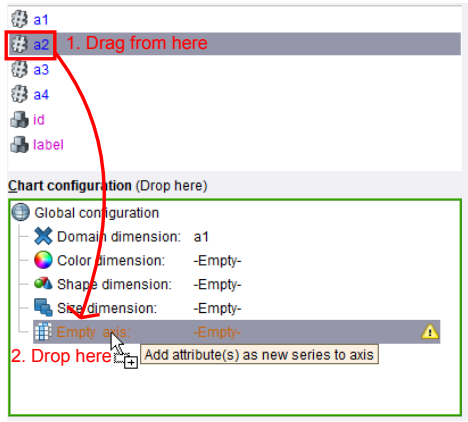

Figure 3.3: Drop Attribute $a2$ on empty axis.

3.1.1 Zooming

Now that we have configured our first scatter chart we want to have a closer look on the four upper points that have $a2 \geq 3.95$ and $5.0 \leq a1 \leq 6.0$. To achieve this you can press the left mouse button on the upper left corner of your desired area and then move the mouse to the lower right corner where you can release the

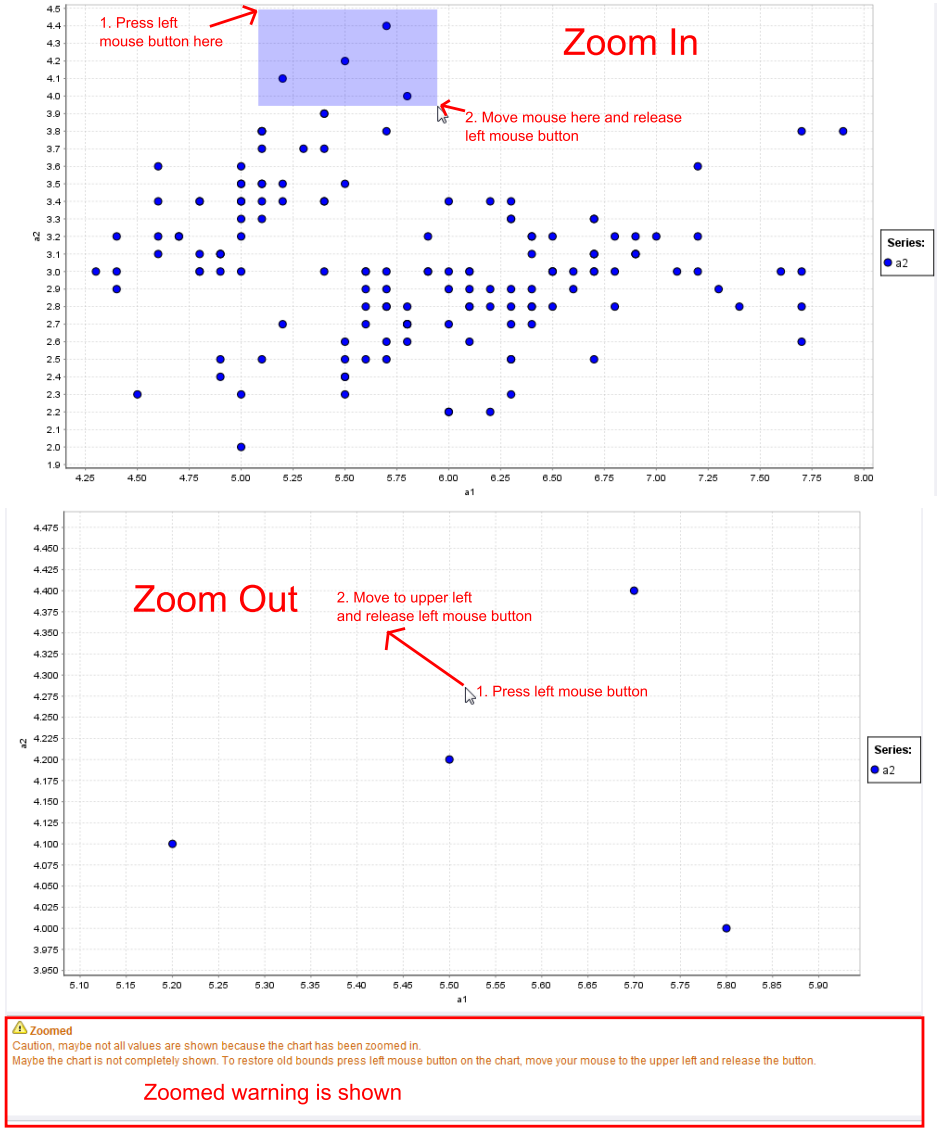

Figure 3.4: Zoom in and out on a configured chart. Furthermore if zoomed in a warning is shown.

button to zoom in. While selecting the zoomed area will be highlighted by a blue painted overlay as seen in Figure 3.4. After zooming in a warning will be shown in the *Notification Area* that remindes you that the chart has been zoomed in. Furthermore you will see only the four points on that you have zoomed in. Note that there have been no changes to the underlying data, only the chart display has changed. To zoom out you can press the left mouse button anywhere on the *Chart Area*, then move the mouse to the upper left and release the left mouse button afterwards. Thereby the old view will be restored and the whole data will be shown again.

3.1.2 An Advanced Scatter Chart

For now our scatter chart shows the attribute $a1$ on the x axis and the attribute $a2$ on the range axis. To enhance our scatter plot with even more information we can drag the Attribute $a3$ on the *Color dimension*. Now each data point is colored with the value of $a3$ - points with a low value of $a3$ are colored blue whereas points with a high value are colored red.

3.1.3 Exercises

Exercise 3.1. *Add the attribute* $a4$ *onto the* Size dimension. *What did you expect? What can you actually see?*

Exercise 3.2. *Furthermore drag the label onto the* Shape dimension. *Are you suprised by the result?*

3.2 A Time Series Chart

As we have seen it is pretty easy to create scatter charts. But how about time series charts? They can be configured with only a few more clicks. This time we will use a dataset provided by a *Generate Sales Data* operator (see fig. 3.5). We drop the *date* attribute on the *Domain dimension* and the *amount* attribute on the empty axis thus receiving a simple scatter chart. To transform this scatter chart into a time series chart we have to adjust two settings. First of all we will set the Item shape to *None*. This will bring up an warning that a series isn't

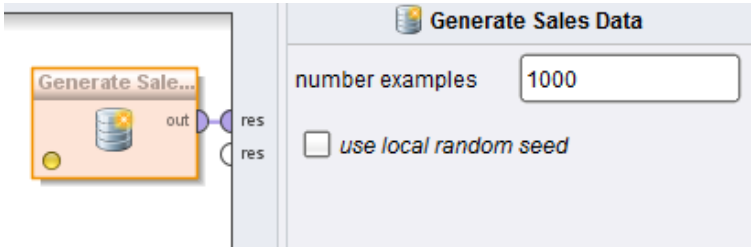

Figure 3.5: Example process to generate time series data.

visible because both Item shape and Line style are set to *None*. Again there is an easy solition: We obviously have to enable lines which can be achieved by setting the Line style of the series format to *Solid* (see fig. 3.6). That's it. Pretty easy, isn't it?

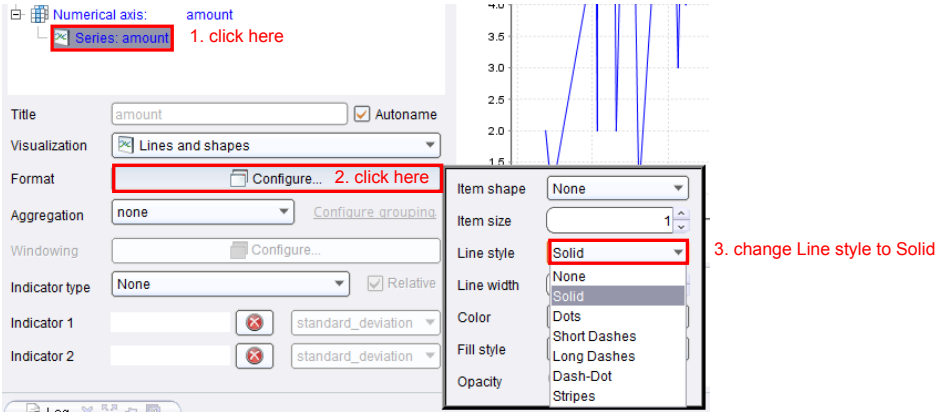

Figure 3.6: Set line style to solid to get a time series chart.

4 Creating Bar Charts

4.1 A Simple Histogram

Now that we have seen how to create scatter charts, our next target will be to create nicely aggregated *bar charts*. We will explain bar charts by the example of a histogram of our beloved Iris data set.

First of all, open the Iris dataset from the samples repository, switch to the *Advanced Charts* view and click the *Reset* button to clear any left-overs from previous charts. Next, drag the attribute *a1* from the *Attribute List* onto the *Domain Dimension* in the *Configuration Tree*. That will display *a1* on the domain axis (x-axis) in the chart when it is drawn. However, as before, the chart will remain empty until we drag an attribute on a range axis: do so now by dragging *a2* on the range axis.

This will result in a scatter chart of *a2* versus *a1* with blue round dots. To change the chart to display a bar chart, we need to change the *Visualization* in the newly created *series configuration*. Click on the sub-item of the numerical axis in the configuration tree labelled *a2*. In the configuration panel below change the *Visualization* to *Bars* (see fig. 4.1). If you expected to see a bar chart right now, your enthusiasm will be diminished by an error message with the title *Duplicate Value* (see section 4.2). Read the message carefully, since most of the times the messages and warnings in the notification area provide you with valuable information about the error and often propose a resolution.

In this case, the error message tells us, that the series representation that we have configured does not support duplicate values. Thinking about it this makes perfectly sense: in the Iris dataset, for each value of attribute *a1* (which we placed on the domain dimension) there is more than one value for *a2* (which we

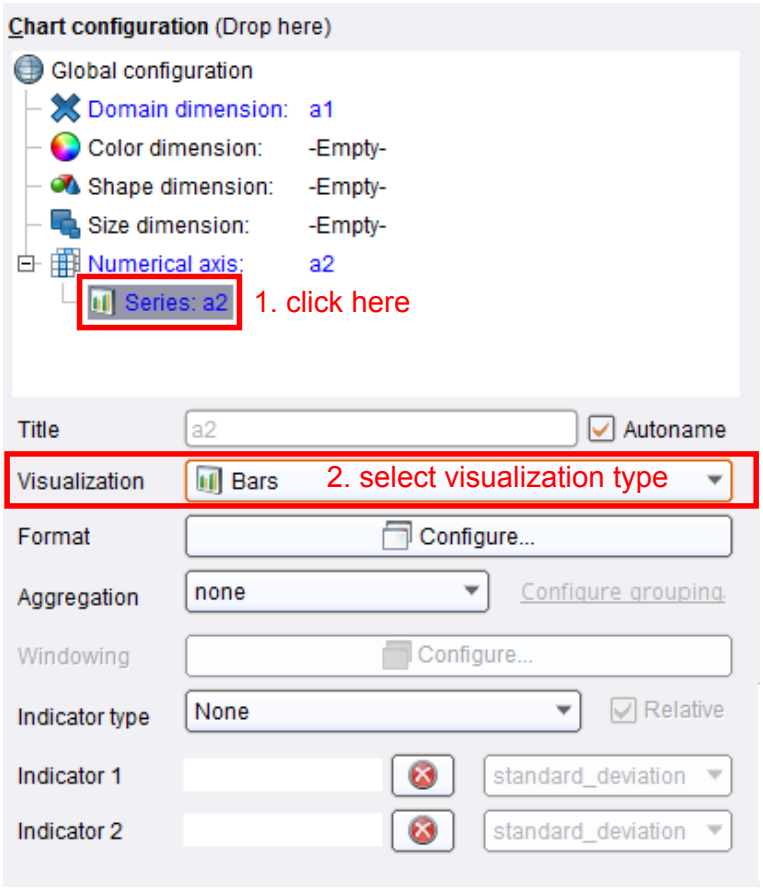

Figure 4.1: Configuring the visualization type for the series.

placed on the range axis). Thus we would have to draw several bars for each x value, which is not possible in a sensible way.

What we have to do is to *group* all examples with the same x value, i.e. in this case examples with identical values for *a1*. Once grouped, we can calculate an aggregate function like average, minimum or maximum on the values of one group. Since we are going to create a histogram, we have to *count* the examples combined in each single group. But first, let's configure the grouping of the domain dimension. It is located in the configuration panel of the Domain dimension: click it in the configuration tree.

What's that? It seems that there already is an *equidistant fixed bin count* group-

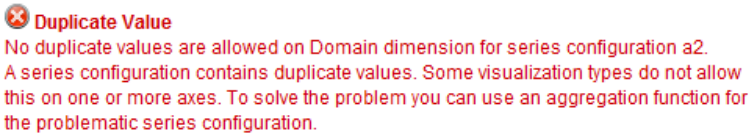

Figure 4.2: Error message stating that the series contains duplicate values.

ing defined: this is the default for a new chart configuration. In our case we change the default *Equidistant fixed bin count* grouping to *Distinct values*, which will create a group for each distinct value on the domain axis (see fig. 4.3). It is configurable via the configure button next to the drop down list, but for now we are fine with the defaults.

Even though this is the correct grouping for our task, the message about the duplicate values persists: to get rid of it, an *aggregation function* for the series must be created: go back to the series configuration (you remember that this is the sub-item of the numerical value axis labelled *a2*). Here, the cause of the error about the duplicate values is hidden: the *aggregation function* is still set to *None*, and thus the grouping of the domain dimension is not used at all. For our histogram, change it to *count* (see fig. 4.4), and your first histogram made with RapidMiner's *Advanced Charts* appears on screen. It should be similar to figure 4.5.

4.1.1 Understanding the Chart

Let's analyze this chart; what is shown now? Not surprisingly, on the domain axis the values of attribute *a1* are shown. Since attribute *a1* is a numerical attribute, the domain axis is by default also numerical, i.e. it shows a continuous range of numerical values. This is indicated also by the blue text for *a1* in the attribute list and for the *Domain dimension* in the configuration tree.

Because we chose *distinct values* as grouping type for the domain dimension and specified an aggregation function for the series *a2*, examples with identical values for *a1*, the attribute on the domain axis, are grouped together and counted (since we specified the *count* aggregation method). For each group one bar is shown, and its height indicates the result of the aggregation function, and its horizontal center specifies the domain value of its group.

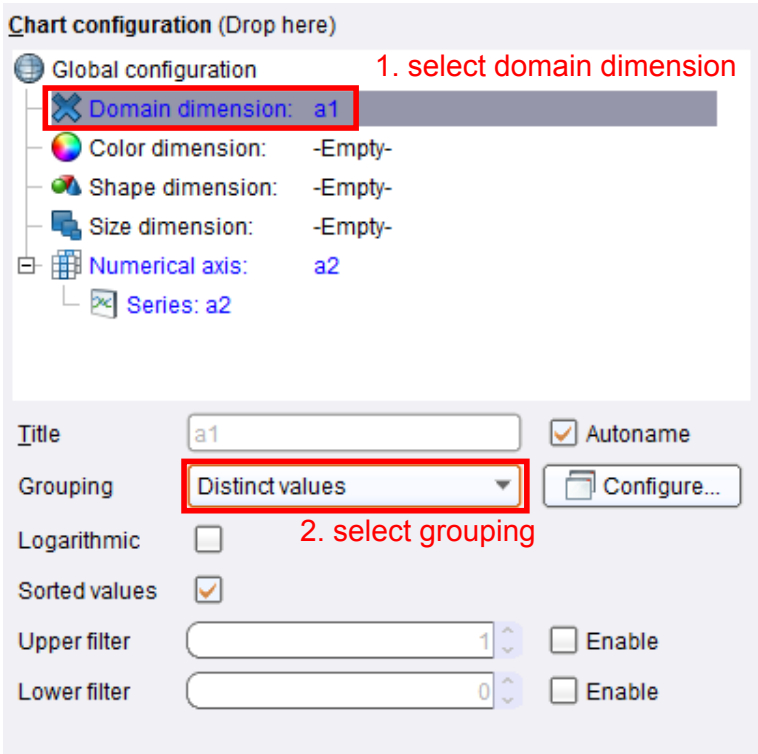

Figure 4.3: Selecting the grouping.

Next to each axis the label is shown, which states what the axis showing. On the right there is the legend located, with an icon showing the pictogram of a bar chart with the same color as in the chart, and the series' label next to it.

4.1.2 Fine Tuning 1: Categorical Grouping

After understanding the chart some possibilities to improve its readability and expressiveness might come into your mind. First of all, it is quite hard to recognize the exact domain value of each bar, since your eyes have to approximate the center of each bar and then guess the value of this point on the axis. To come around this, the grouping can be changed to be *categorical*: that way, the domain axis is converted from a numerical axis into a categorical axis, and to each bar or each group a label will be assigned. Go to the domain configuration and hit the *Configure* button next to the drop down list for the grouping type (see fig. 4.6).

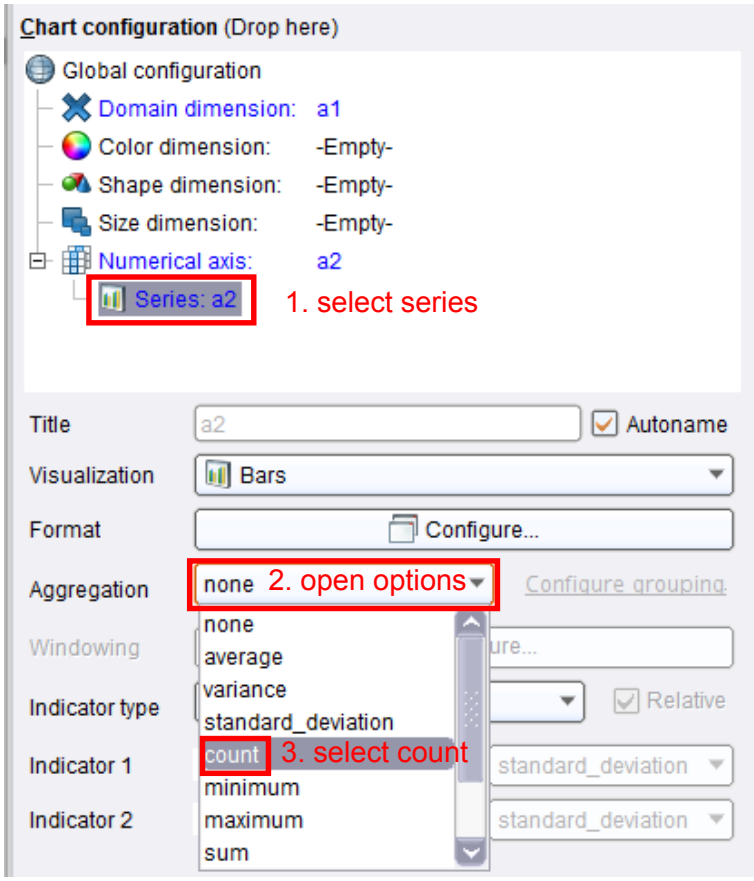

Figure 4.4: Selecting a series aggregation function.

A popup panel will show up with a checkbox to make the grouping categorical. Since the distinct value grouping has no further options the panel is empty otherwise. Enable and disable the checkbox a couple of times and observe the chart and also the configuration tree: in the chart you will see that with the categorical grouping each bar is labelled with the according value. You will also see that the two empty groups on the far right vanish and the adjacent bars are direct neighbors in the categorical chart. Whereas categorical dimensions haave labels that represent ranges in that dimension, their position on the domain axis does not correspond to this range. Rather than that, they are equidistantly placed on the axis. You recognize a categorical domain axis at first glance by the rotated labels.

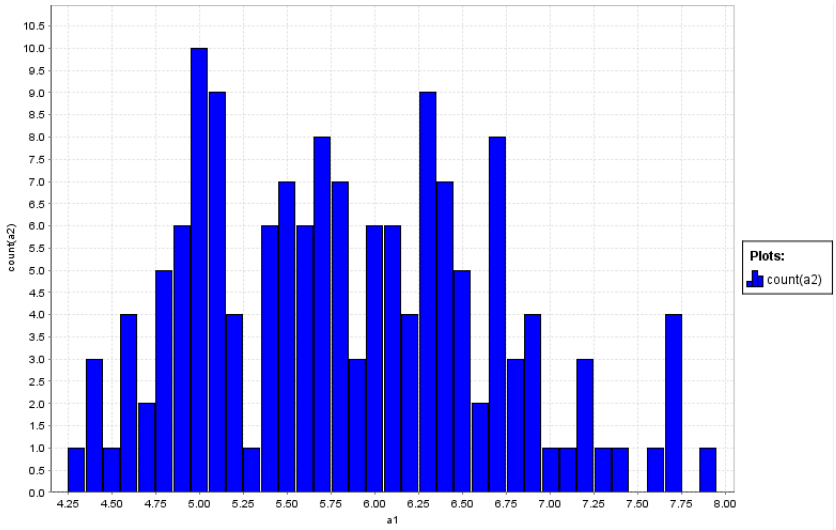

Figure 4.5: The final histogram with distinct value grouping.

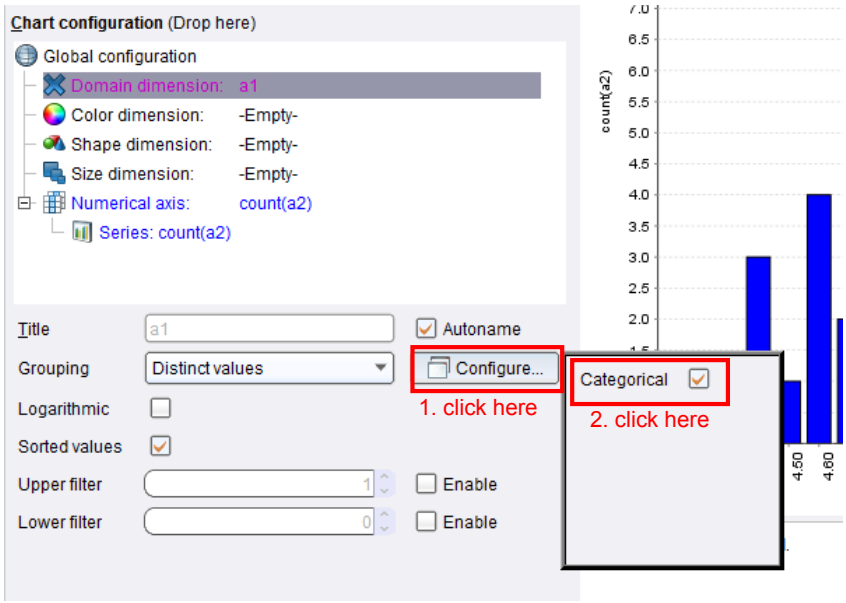

Figure 4.6: Set grouping to categorical.

But not only the chart view changes, but also the configuration tree: when you enable the categorical grouping, the color of the domain dimension switches to purple to indicate that it is now categorical.

4.1.3 Fine Tuning 2: Configuring Different Groupings

Currently, our series contains one group for each distinct value of the attribute on the domain axis. For nominal attributes (indicated by the color purple and a special icon in the attribute list) this is the only choice you have. For numerical attributes however you have two other choices: the *equidistant fixed bin count* grouping and the *equal data fraction* grouping. The equidistant fixed bin count grouping divides the data range on the domain axis into n intervals of equal size (hence the name), whereas the equal data fraction grouping tries to find interval bounds such that each of the n intervals contains the same amount of data points (i.e. examples). Note that depending on the distribution of the data this is not always possible.

In addition to setting the *bin count*, the equidistant fixed bin count grouping allows you to specify manual bounds for creating the grouping: if you disable the *auto range* option and specify manual bounds, the grouping creates as many groups as specified by *bin count* within the bounds. Data points located outside the range are collected in so-called *overflow bins* which are displayed left and right of the normal bins. *Note that the overflow bins are only displayed when the grouping is set to* categorical.

Try out all three grouping types and play around with their options.

4.1.4 Summary: Four Steps to a Simple Bar Chart

These are the steps to create a simple bar chart:

1. Drag the attribute on which you want to create the grouping onto the domain dimension.
2. Configure the grouping.
3. Drag the attribute to be aggregated onto a range axis.

4. Open the series configuration for the attribute and select an aggregation function.

4.1.5 Exercises

Exercise 4.1. *Load the Iris dataset and create a bar chart which displays the number of examples of each* label *value. Looking at the* Meta Data View *(leftmost radio button on the very top of the results view), are you surprised by the result?*

Exercise 4.2. *Change the chart from the previous exercise to display the average of* a1 *for each label.*

4.2 A Histogram for each Label in one Chart

In the previous section you have learned how to create a simple bar chart, e.g. a histogram over one attribute. In this section you will learn how to split up the histogram into three histograms with different colors, where each color represents one label value, i.e. *Iris-setosa*, *Iris-versicolor* and *Iris-virginica*.

We could simply create such a chart with a small change to the previous configuration, but for the sake of learning we will start with a fresh configuration: please click the *Reset* button to clear the previous configuration.

Now let's start constructing a new histogram: as before, drag atttribute *a1* from the attribute list to the Domain dimension. Configure the grouping to an *equidistant fixed bin count* grouping with 10 bins (remember that it will not be used until you select an aggregation function in a series configuration).

Then drag *a2* to the Range axis. A scatter chart with blue points will appear, since this is the default configuration (again: the grouping is not yet used, since there is no aggregation function selected yet). For the moment we will leave it like this, and instead of directly selecting an aggregation function, we drag the *label* attribute onto the Color dimension. Now, instead of a single-colored chart, the dots are colored according to the label values.

The legend on the right provides the mapping from colors to values, and the color of the *a2* series in the legend switched to a neutral grey, indicating that the

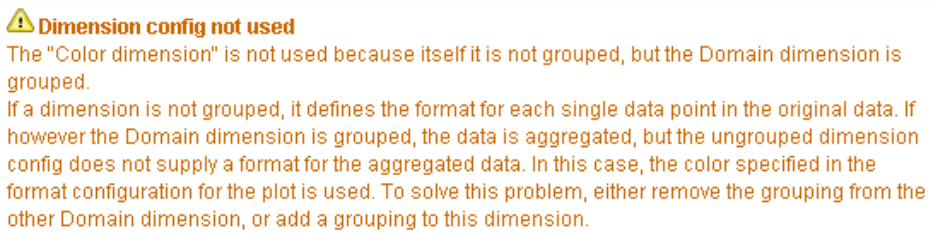

Figure 4.7: Warning about a dimension not being used.

legend entry only refers to the shape, not to the color which is now specified by the Color dimension.

On the other hand, the icons of the color dimension are unshaped splashes, which indicates that the label values only affect the color, but not the shape.

However, our actual goal is to create a colored three-in-one histogram for all label values. It seems that we halfway reached this goal, at least the colors are already there. Remembering the previous section the natural workflow will be to switch the *Series type* of the *a2* series configuration to *Bars* and then select the aggregation function *count*.

The result is not quite satisfying: the error message about the duplicate values vanishes after selecting the aggregation function, but also the colors are gone! Instead a rather long warning is shown in the notification area (see fig. 4.7).

To understand this message, we would have to dive down into the mechanics of the grouping and aggregation framework, but we will postpone that excursion till section 6.3. For now only the solution is provided, which is to go to the configuration panel of the Color dimension and switch the grouping from *None* to *Distinct Values* (see fig. 4.8).

Now, the series contains three histograms, one for each label value. The result should be similar to figure 4.9. Awesome, isn't it?

4.2.1 Stacking the Bars

In the just created chart, the bars of the different colors are displayed next to each other. It is easily possible to stack them: just go to the series configuration

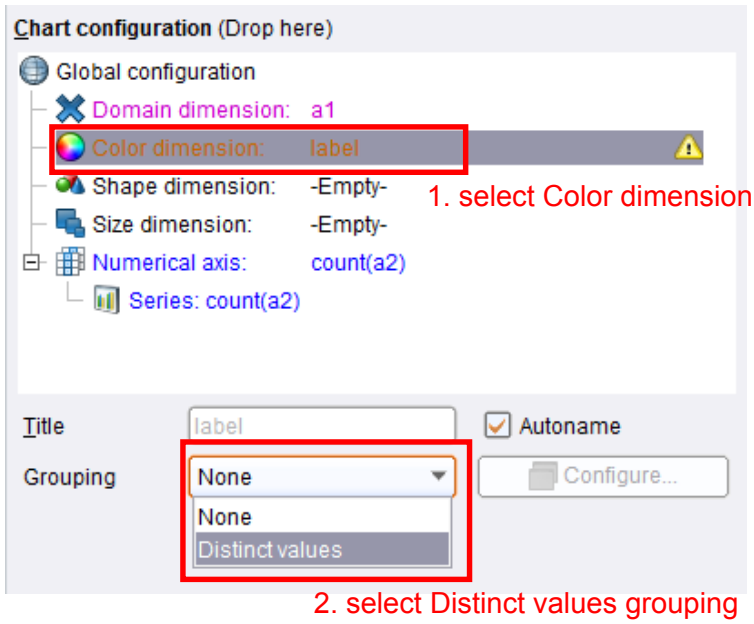

Figure 4.8: Configuring a grouping for the color dimension.

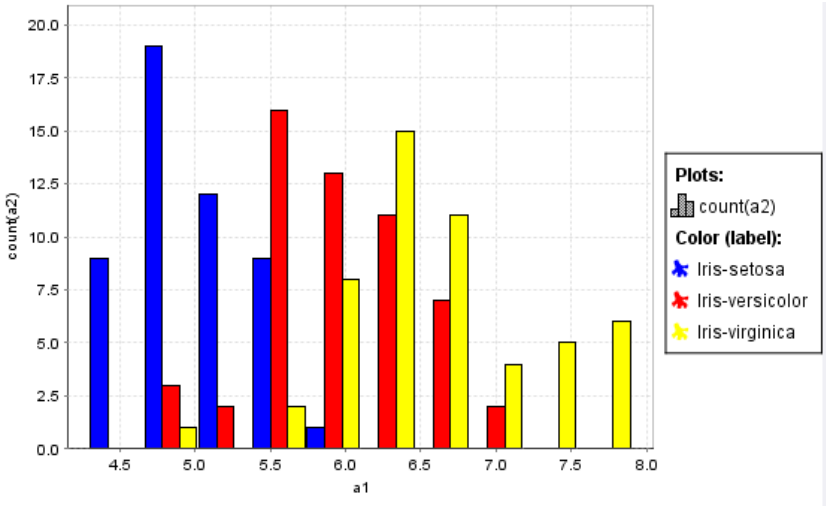

Figure 4.9: A bar chart with colored histograms for each value of the label.

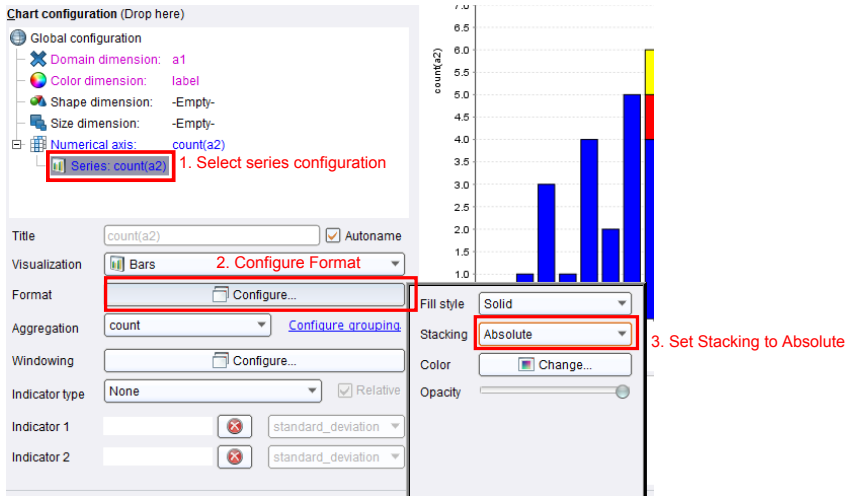

Figure 4.10: Configure stacked bars.

of the aggregated attribute (i.e. *count(a2)*), click on *Configure Format* and select the desired stacking (see fig. 4.10).

The default setting is to use no stacking: as you have seen, all the bars are positioned next to each other. If you select an *Absolute* stacking, the bars are placed on top of each other: now the height of the bars is identical to the height of an uncolored bar chart, and the ratios of the colors indicate the amount of examples of the specified color (see fig. 4.11).

Relative stacking is similar, but here the height of all bars is normalized to 1.

Note: you can only stack bars of different colors which have been configured the way we did in this chapter. There are other ways to create more than one histogram in a chart (you will learn about these ways later in this document), but most of them are not stackable. More precisely, you can only stack bars which are created within a single series configuration.

4.2.2 Summary: Seven Steps to an Advanced Bar Chart

These are the steps to create a simple bar chart:

1. Drag the attribute on which you want to create the grouping on the domain

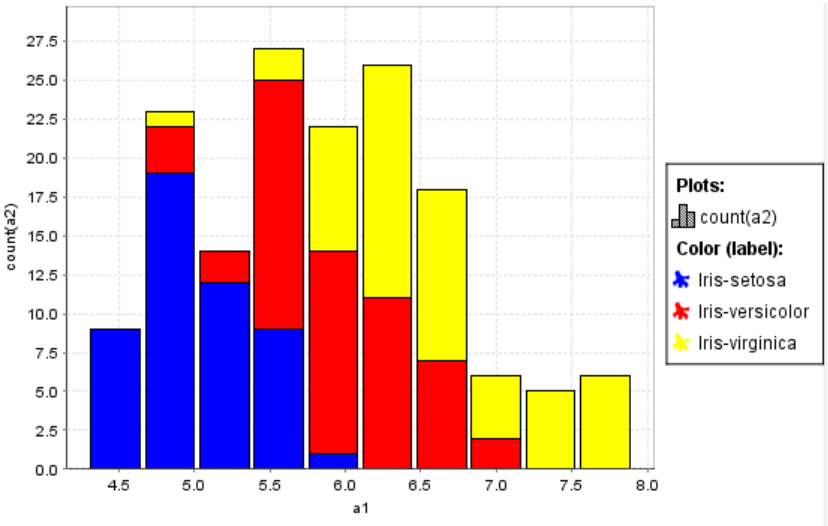

Figure 4.11: A bar chart with colored histograms for each value of the label.

dimension.

2. Configure the grouping.
3. Drag the attribute to be aggregated on a range axis.
4. Open the series configuration for the attribute and select an aggregation function.
5. Drag the attribute whose values provide the colors on the color dimension.
6. Configure a grouping on the color dimension.
7. In the series configuration of the aggregated attribute, configure the stacking (if desired).

4.2.3 Common Pitfalls

When creating bar charts, remember the following things to stay out of trouble:

- As you have seen in the description above, it is important that all dimensions with an attribute on them have a grouping if a series configuration uses an aggregation function. The reason for this will be explained in sec-

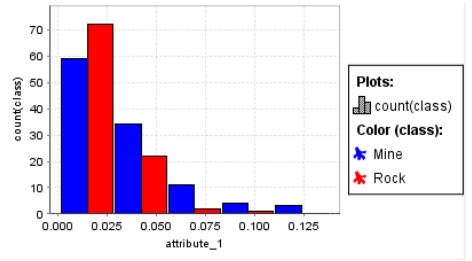

Figure 4.12: Solution for Exercise 1.

tion 6.3.

- Only bars created within a single series configuration can be stacked.

4.2.4 Exercises

Exercise 4.3. *Open the* Sonar *dataset from the Samples repository and plot the distribution (i.e. a histogram) of the label (attribute* class*) versus* attribute_1*.*

The solution should be similar to figure 4.12.

Exercise 4.4. *Configure the previously created chart to display the exact group boundaries on the domain axis.*

The solution should be similar to figure 4.13.

Hint: *Remember categorical groupings.*

Exercise 4.5. *Instead of a histogram, now show the average of* attribute_2 *in each interval on the domain axis.*

The solution should be similar to figure 4.14.

Hint: *you will need to change the aggregation function.*

4.3 Further Reading

Chapter 5.3 describes how to create a bar chart with horizontal bars. For a more detailed discription of the grouping mechanism read chapter 6.3. If you need

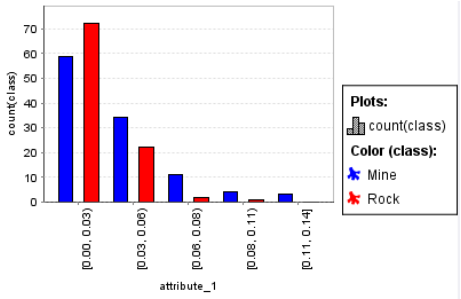

Figure 4.13: Solution for Exercise 2.

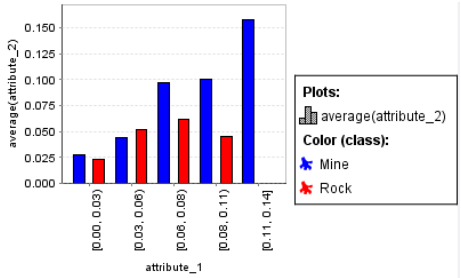

Figure 4.14: Solution for Exercise 3.

cumulative bar charts look at chapter 6.5.

5 Formatting the Chart

RapidMiner's Advanced Charts are fully customizable. In this chapter we will learn how to configure the format of both the chart itself, e.g. chart and axis titles, used fonts, background colors etc., and the series in the chart, including different line styles, shapes, colors etc.

5.1 Preparing a Sample Chart

Obviously to format a chart, we need a chart – let's create one! Load the Iris dataset, and create a simple scatter chart by dragging attribute *a1* onto the Domain dimension and *a2* onto the range axis. You should end up with a scatter chart.

5.2 Formatting Series

5.2.1 Formatting a Series for a Scatter Chart

The scatter chart consists of blue, fully opaque circles of a certain size, which are not connected by any lines. You can change all these features in the format configuration popup of the series configuration. Navigate to the series configuration panel by clicking the series configuration *a2* in the configuration tree. Then click the *Configure* button for *Format* (see fig. 5.1).

Let's handle the options from top to bottom. The very first entry is the *Item shape*. From the combobox you can select the shape to your liking. Just give it a try. If you switch the shape to *None* the chart will be empty and an according

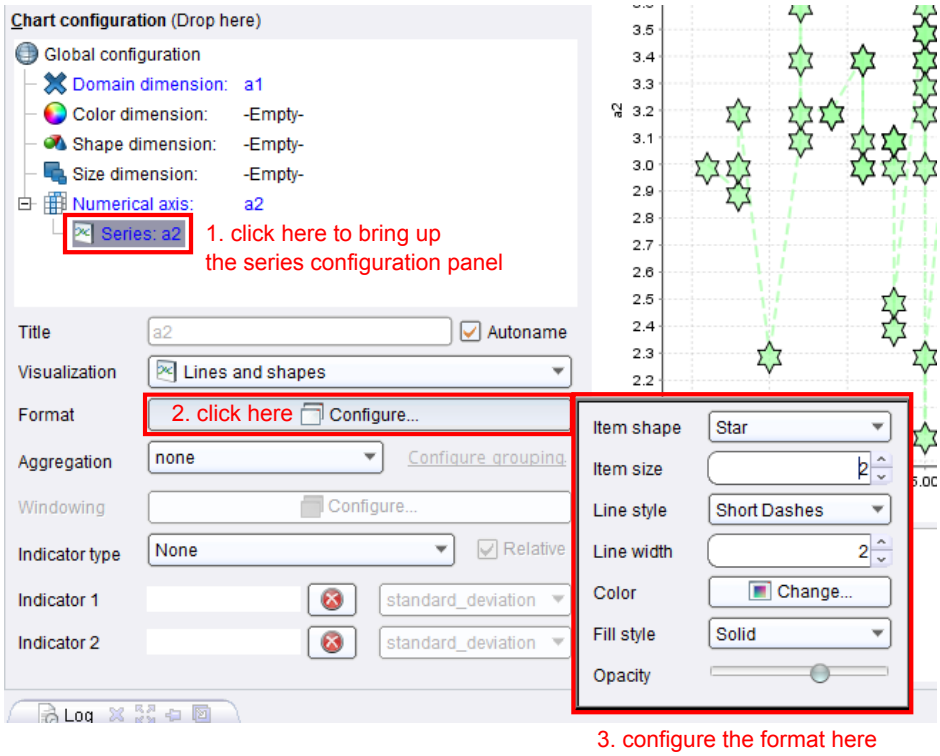

Figure 5.1: Formatting a series for a scatter chart.

warning will be shown. You will observe that the change of the item shape is immediately resembled in the legend. This is true for all settings available in the format configuration panel.

The second entry *Item size* should be quite self-explanatory: here you can change the display size of the items in the chart.

The next entry *Line style* allows you to add lines to the chart. If you change it from the default setting *None* to something else, you will see that the points are connected with lines of the specified style. The width of the lines can be adjusted with the following entry *Line width*. If you now set the item shape to *None* you end up with a chart containing only lines, but to continue with this manual, please configure an item shape different from *None*.

To change the color of the series use the *Change* button next to the label *Color*.

To get to grips with the next entry *Fill style* you best select a large *Item size* (sth. like 5) and set the *Line style* back to *None*. Now you can change the *Fill style* of the points with the according combobox.

Now please reset the *Fill style* to *Solid* and try out the *Opacity* slider: it affects the opacity or transparency setting of the points. If you move the slider to about the half of range, you will notice that some points appear darker than others: that it because actually there are multiple points directly above one another, and their opacity values add up to create a darker color.

Ordering of Data Points

By default, the points are drawn left to right, i.e. they are sorted by their domain values (in this case, by *a1*). You can also draw the points in the order of appearance in the dataset. To change the setting go to the configuration of the domain dimension and deactivate the option *Sorted values* (see fig. 5.2).

Now the points are drawn in the same order as the examples appear in the dataset. To observe the impact of this setting you have to set a line style other than *None* on your series.

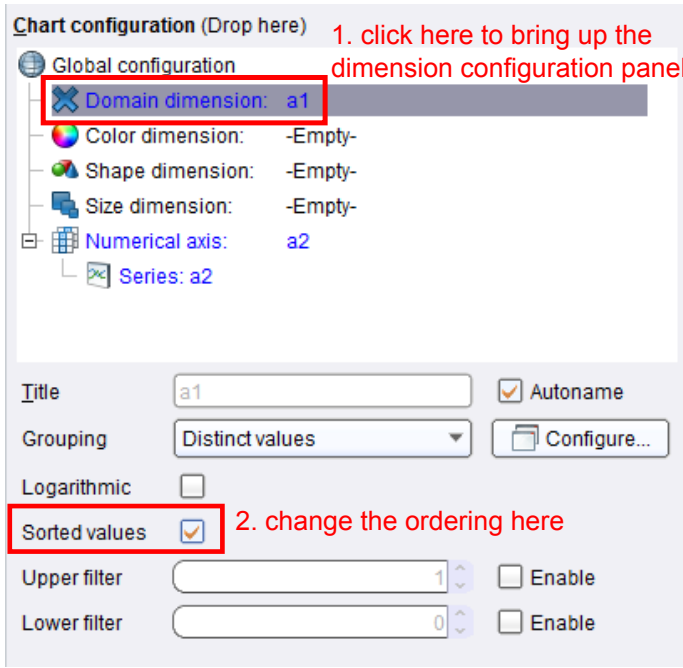

Figure 5.2: Configuring domain value sorting.

5.2.2 Formatting Series for Area and Bar Charts

To see the format options of area and bar charts, please change the *Series type* to *Bar* (or *Area* if you like, the options are identical). Additionally you have to select an aggregation function further down to avoid the error about duplicate values (see chapter 4 for further information).

If you now reopen the format configuration of the series, you will see that only fill style, series color and opacity are left, and a new setting *Stacking* has appeared. The stacking is explained in detail in chapter 4. The other three options fullfill the same purpose as described in the section above. Setting a line style or an item shape obviously makes no sense for area and bar charts, so these options are not available.

5.2.3 Common Pitfalls

Not all of the format settings are always used. The color is ignored if you drop an attribute on the Color dimension in the configuration tree, the size is ignored if you drop an attribute on the Size dimension, and the shape is ignored if you drag an attribute on the Shape dimension.

5.3 Formatting the Chart Area

In the previous sections we concentrated on changing the appearance of the *series representations*, that is the items drawn in the chart. Now we change the focus to the chart area itself: after reading this chapter you will be able to change the background colors of the chart area, the axis fonts and colors, configure the legend style and define a chart title.

First of all, create a simple chart, you could e.g. load the Iris dataset from the Samples repository and drag *a1* onto the Domain dimension, *a2* on the range axis and the *label* on the Color dimension. Now we have a good point to start.

All format options for the chart can be configured in the *Global configuration* panel. Open it by clicking *Global configuration* in the configuration tree (see 5.3).

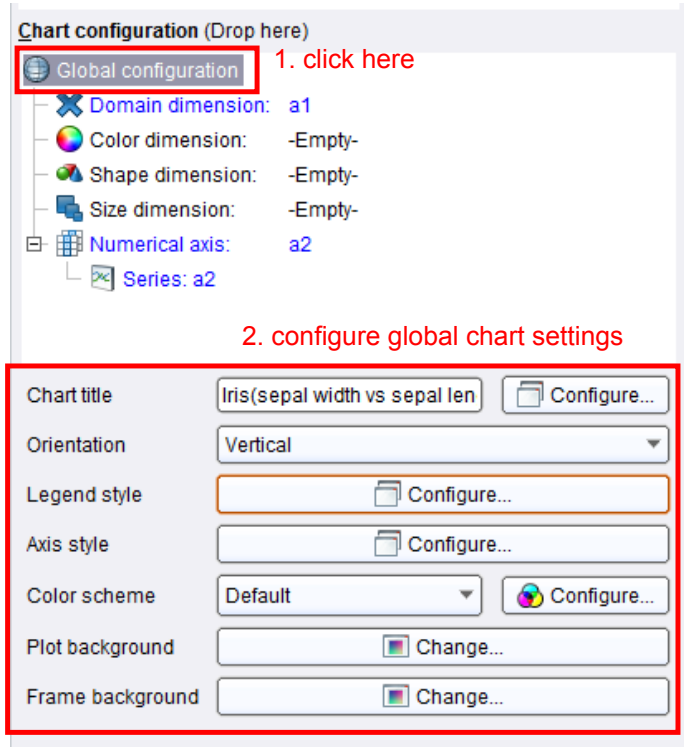

Figure 5.3: The global configuration panel.

The panel contains several options, which we will discuss top to bottom. In the input field next to *Chart title* you can specify a heading for the chart. It will be updated as soon as you start typing. With the *Configure* button next to it you can change its color and font.

By changing the *Orientation* of the chart it is basically mirrored at its diagonal, i.e. the domain axis is drawn vertically on the left, the range axis horizontally at the bottom. With this option you can draw bar charts with horizontal bars.

The next entry is the *Legend style* which is treated in detail in the next section.

In the *Axis style* popup the font, color and width of the axes is configurable.

The *Color scheme* is used to determine the colors used when an attribute is placed on the Color dimension, and also determines the automatic colors for additional series (see chapter 6.2). You can select a predefined scheme from the drop down list, or define custom schemes via the *Configure* button. Note that it is not yet possible to permanently save custom schemes, i.e. they are valid only for the current chart.

The last two options *Plot background* and *Frame background* allow you to specify background colors for the background of plot area and the chart frame.

5.3.1 Configuring the Legend

To open the legend configuration panel open the global configuration panel (see section above) and click the *Configure* button next to the label *Legend style* (see fig. 5.4). Here you are offered options to configure the legend *Position* (where you can also completely hide the legend), *Font* style and *Font color*, legend *Background color* and the presence and *color* of the legend *Frame*.

The option *Show dimension type* affects only charts where an attribute is placed on one of the dimensions, e.g. the color dimension. Thus, if you want to try this option, create a chart as described in the beginning of the previous section.

Now your legend contains an entry like *Color (label)* and below it three entries for the three different label values, each preceded by a splash of the corresponding color (see fig. 5.5). Since you have these splashes, the word *Color* in the heading might seem redundant to some users. By deactivating the option *Show dimension*

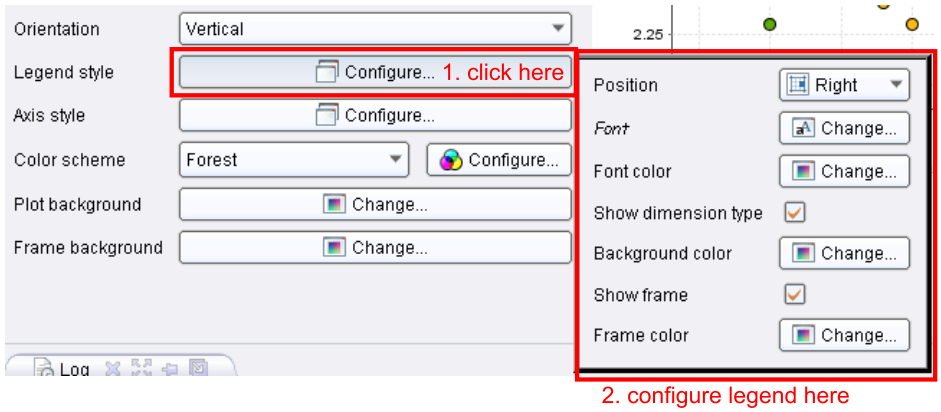

Figure 5.4: The legend configuration panel.

Figure 5.5: A formatted legend (not pretty, but omnipotent).

type you can hide the dimension name from the legend, and only the attribute on the dimension remains, in this case *label*.

5.3.2 Setting Axis, Dimension and Series Labels

The labels for the axes and the dimensions can be configured directly in the corresponding axis and dimension configuration panels. For example, to change the word *label* in the legend entry for the Color dimension in the sample legend in figure 5.5 open the configuration panel for the Color dimension by clicking the entry in the configuration tree, deactivate the option *Autoname* and enter the title of your choice into the textbox.

The same way you can change the label of the domain axis.

To change the label of the Range axis, bring up the range axis configuration panel by clicking the *Numerical axis* entry in the configuration tree and execute the same steps as for the dimensions. If you re-enable *Autoname*, the label of the range axis is composed from the labels of the series on this axis.

The name of series in the legend can be changed directly in the series configuration panel: just click the series entry *a2* in the series configuration tree to find the same options as described above for axes and dimensions. If *Autoname* is enabled, the name is constructed from the attribute name and the aggregation function (if present).

5.4 Exercises

Exercise 5.1. *Starting from the chart described at the beginning of section 5.3 (Iris;* a1 *on Domain dimension,* a2 *as series,* label *on Color dimension), create a chart similar to the one shown in figure 5.6.*

Hints:

- *Start with the axis and dimension labels.*
- *Then configure the background and axis colors and the axis font, the chart title etc.*
- *To change the color of the data points you have to change the color scheme.*
- *Finally configure the legend style.*

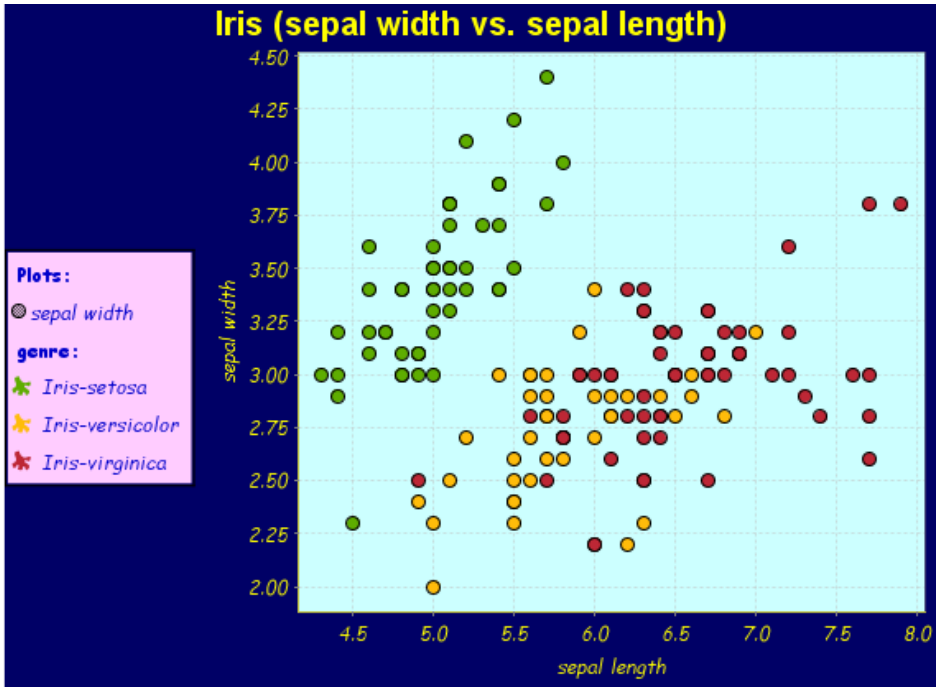

Figure 5.6: Fully customized chart.

6 Advanced Plotting

This chapter describes advanced plotting techniques. Before approaching this chapter you should have read and understood all previous chapters.

Section 6.1 explains how axis ranges and dimension ranges work. Then, in section 6.2 you will learn how you can create a single chart which contains several series. The next section 6.3 explains the grouping and aggregation mechanism in detail. Drawing error indicators and graphically comparing two series is explained in section 6.4, and the advanced topic *windowing* is handled in the last section 6.5. That technique allows you to create cumulated charts as necessary e. g. for lift charts.

6.1 Defining Axis and Dimension Ranges

Often it is desirable to show only parts of the data or of a series, or to enforce certain ranges on the axes to guarantee comparability between different charts. How to do this with RapidMiner's Advanced Charts is described in this chapter. Please note that there is a substantial difference in setting ranges on the range axis and on the domain dimension (or other dimensions). These differences are handled in the following chapters.

6.1.1 The Defaults

The domain dimension (x-axis) by default is scaled such that all of the data is visible, i.e. no filtering or range cuttings are performed. This is also true for other dimensions like the Color dimension.

The default values of the range axis (the y-axis) depends on the *Visualization* type of the current series. For series with a visualization of type *Lines and shapes* the range axis is scaled similar to the domain axis, i.e. such that all data is visible. In contrast, if the axis contains a *Bar* or *Area* chart, it is ensured that zero is contained in the visible range. The other end of the range is set such that, again, all data is visible.

Exercises

- Create a sample chart from the Iris dataset in the Samples repository. Create series with different visualization types and observe how the default ranges on the axes change:
  - Put different attributes on the range axis and the Domain dimension.
  - Create charts with different series visualization (scatter, bars and areas).

    Hint: If you experience problems, please work again through sections 3.1 and 4.1.

6.1.2 Preparing a Sample Chart

To follow the instructions given in the following chapters, please create a simple scatter chart from the Iris dataset: put attribute *a1* on the Domain dimension, attribute *a2* on the range axis and the *label* on the Color dimension.

6.1.3 Zooming

The simplest method to define the visible ranges of the chart is the zooming tool which has been described before in section 3.1.1: just move the mouse to the upper left part of the area you want to focus on, press the left mouse button, drag the mouse to the lower right corner of the desired area and release the button. The view will change to the just-selected area. To zoom out and return to the complete view drag the mouse from the lower right to the upper left while pressing the left mouse button.

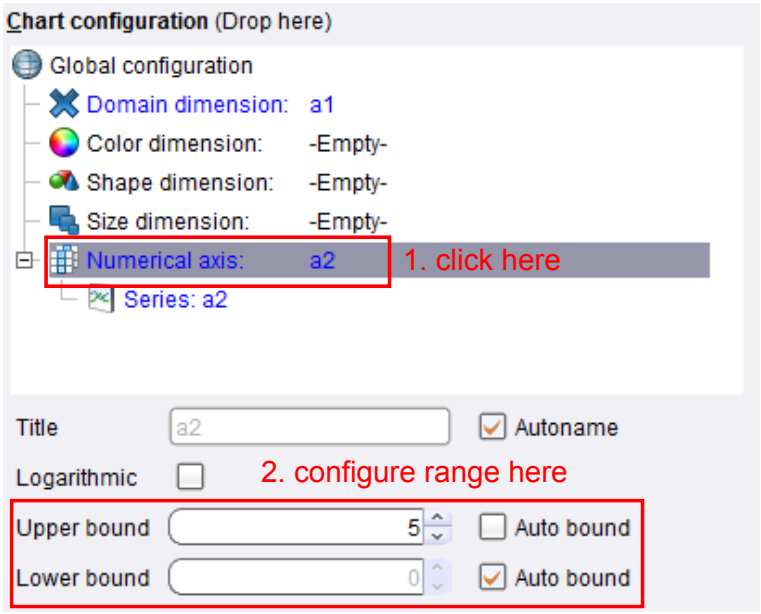

Figure 6.1: Configuring the range of the range axis.

Zooming has no impact on the underlying data, it is just a change of the visualization.

6.1.4 Visible Range of the Range Axis

The range axis is the vertical axis (unless you switched your chart to be horizontally oriented in the global configuration panel). You can define its range in the axis configuration panel which is accessible by clicking on the axis in the configuration tree (see fig. 6.1).

Here you can manually specify values for the bounds of the range axis, individually for the upper and lower bound: just disable *Auto bound* for the bound you want to define and enter a value in the spinner.

If *Auto bound* is activated for a bound, it is determined automatically in accordance to section 6.1.1.

If the axis contains nominal values, the filter values refer to the categories or nominal values in ascending alphabetical order. Counting starts at 0. That

means that if you have the categories A, B and C and set the range from 0.5 to 2, only the categories B and C will be visible.

It is important to understand that the bounds you set don't have an impact on the underlying data or the groupings and aggregations – the range is only applied visually, i.e. it acts similar to normal zooming as described above.

Common Pitfalls

When setting bounds manually, three errors can occur. Firstly, you might enter an upper bound which is lower than the lower bound you entered. This is easy to fix by entering sensible values for upper and lower bound.

Secondly, the range you entered might contain no data. Attribute *a2* of Iris for example ranges from 2.0 to 4.4. If you scale the axis to the range from 0 to 1 your input will result in a warning in the notification area stating that the axis contains no data, including the name of the data.

Lastly, if you enter only a value for the upper bound and leave the lower bound to be automatically calculated, and the entered value is less than the minimum value in the data, the plotter can't calculate the range properly and shows an error. In this case increase the value you entered. An analogous error may occur if you only enter a value for the lower bound.

Exercises

Exercise 6.1. *Play around with the range settings of the range axis.*

Exercise 6.2. *Try to provoke the errors described above, understand the error messages and find solutions.*

6.1.5 Filtering on Dimensions

Also in the dimension configuration panels for domain, color etc. you can define ranges in a similar way, with two differences: first, only dimensions with an attribute of type *numerical* or *date* can be filtered. Second, the data is actually *filtered* according to the user defined range. That means that e.g. an equal data fraction grouping does not consider data points outside of the range.

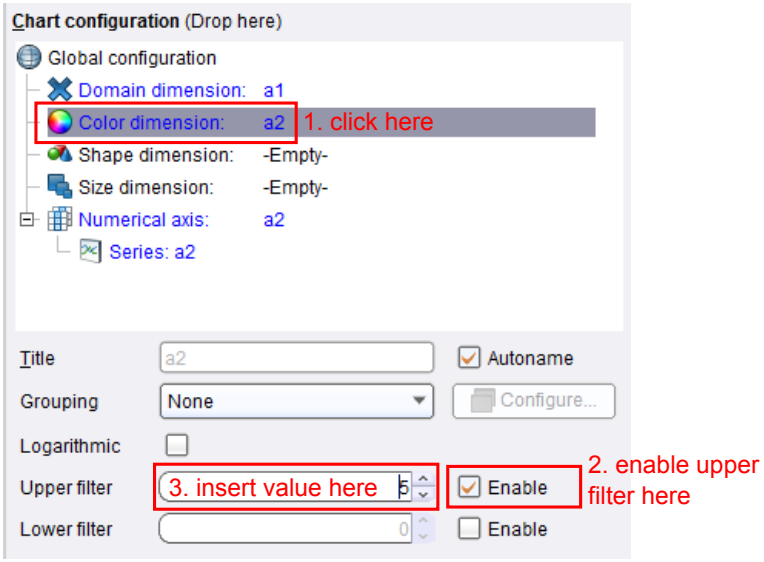

Figure 6.2: Configuring the upper filter value of the Color dimension.

Since the data points are actually removed from the dataset (only for plotting and only as long as you define a filter), filters on different dimensions are combined with a logical *AND*.

Let's demonstrate this with an example. Reset the the chart configuration with the *Reset* button, then add attribute *a1* of the Iris dataset on the Domain dimension and *a2* on the range axis. Then again use attribute *a1* and also drop it on the Color dimension and think for a moment to understand what you are seeing: the data points are colored according to their *a1* value which is also placed on the domain dimension, and thus you see a nice and continuous gradient with ascending x-values.

By looking at the domain dimension or into the legend, we see that *a1* ranges from 4.3 to 7.9. What would happen if you set the *upper* bound of the color dimension to 5? Figure 6.2) shows the result: not only the color dimension has its upper limit at 5 (see legend), but also the domain dimension (see x axis), since it uses the same attribute.

Now let's set the *lower* bound of the dimension config to a value larger than 5, let's say 6. What do you expect? Try it out: as you expected, the chart is empty, and two warnings appear stating that the Domain dimension and the

Color dimension contain no data. Why is this? The Color dimension restricts the examples to the condition $a1 \leq 5$, whereas the Domain dimension requires $a1 \geq 6$. Obviously, no example can exist which fulfills both conditions, and thus the chart is empty.

6.1.6 Common Pitfalls

Remember that ranges on dimensions are actually filters (read section 6.1.5), whereas ranges on the range axis only affect the visualization.

6.1.7 Exercises

Exercise 6.3. *Reset the chart, then drag* $a1$ *on the domain dimension,* $a2$ *on the range axis and* $a3$ *on the color dimension.*

Exercise 6.4. *Set the upper bound of the color dimension to* 7*, then reduce it in steps of* 0.1 *(just use the arrow down button of the spinner). Observe the chart closely and try to figure out what's happening.*

6.2 Several Series in one Chart

Up to now, we only created charts with exactly one series – sometimes it was grouped by color, but anyway we never had more than one series configuration in the configuration tree. But with RapidMiner's Advanced Charts you can do way more: you can add additional series, either on the same range axis (y-axis), or even create one or more new range axes and add series to them. Let us explain this with an example. Assume you created a histogram. Now, to visually check the validity of the histogram, or just to add more information to the chart, you want to overlay it with a scatter chart of the actual, unaggregated data. But beware, this guide will lure you into some traps, because learning by error will make sure you never forget (but be reassured that we will always get you out of any trap).

To start, create a simple histogram on the Iris dataset, as described in section 4.1: drop attribute *a1* on the Domain dimension, attribute *a2* on the range axis, set the aggregation to *count* and switch the series type to *Bars*. Then click the

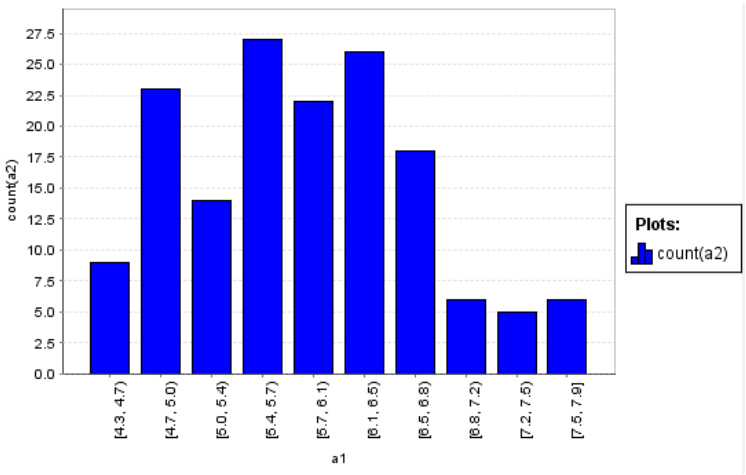

Figure 6.3: A categorical histogram on Iris.

Configure gropuing link to open the domain configuration panel and set the *bin count* of the grouping to 10 for a higher resolution. Since we want to see the exact bin boundaries, switch the grouping to categorical. The chart should now be identical to figure 6.3.

6.2.1 Adding a Second Series

Our goal is to create a chart like figure 6.8, so the next logical step is to drag attribute *a2* again from the attribute list onto the range axis as outlined in figure 6.4. Be sure to drop directly onto the *axis configuration* tree node, not onto the already present *series configuration*. That will append a new series configuration to the range axis.

In the configuration tree a second series configuration also labelled as *count(a2)* appears (see figure 6.5), and the chart shows red dots on top of the bars. Click the new, i.e. the *second* series configuration and examine its configuration. You will note that the aggregation function is copied from the already present series on the axis. The color of the series is determined accordingly to the color scheme, and the series type defaults to *Lines and shapes*. The latter is what we need for our scatter chart, but it should show the unaggregated values: set the aggregation function to *None*.

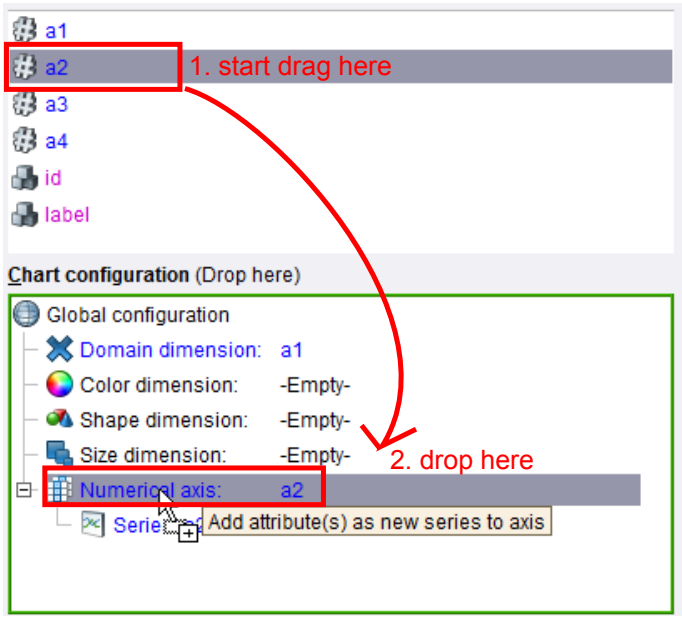

Figure 6.4: Drop a second attribute on the range axis.

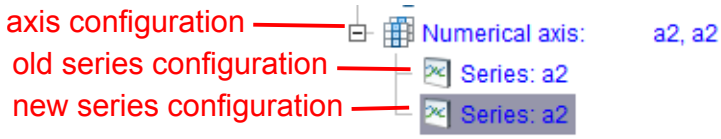

Figure 6.5: The range axis with to series configurations.

But what's that? You just hit the first trap! An error appears in the notification area stating that you mixed *categorical* grouped values and ungrouped values on the same axis, and the configuration tree shows error icons next to the range axis and the domain dimension. What does that mean? Remember that we told you to set the grouping on the domain dimension to categorical? Well, that's the reason for this mess. The categorical grouping causes the domain axis to become categorical (with one category for each group), whereas the unaggregated scatter chart you just tried to create would need a numerical domain axis, since without being grouped and aggregated it has no knowledge about any categories. This conflict causes the error message. The solution in this case it to deactivate the *categorical* option of the domain dimension grouping: click the domain dimension in the tree and click the *Configure* button next to the grouping list to show the option, then deactivate it.

6.2.2 Drawing Order of the Series

Now the chart shows up, and the legend indicates that you added a second series. But you can only catch tiny glimpses of it shining through the gaps between the bars. Obviously the bar chart is drawn over the scatter chart. Have a look at the configuration tree: the histogram configuration, labelled *count(a2)*, is located above the configuration of the scatter chart with the label *a2*. In RapidMiner's Advanced Charts the series are drawn bottom to top, that means that plots that appear above another one in the configuration tree are also drawn on top of the other one in the chart.

You can easily change the order of the series by dragging the scatter series configuration above the histogram configuration. So to bring the scatter chart above the histogram drag it to the top as depicted in figure 6.6.

This looks quite good, the scatter chart is now drawn above the histogram.

6.2.3 Adding a Second Range Axis

Still the chart looks a bit odd: while the bars reach into the sky, the scatter chart si a bit lost at the bottom of the plot area. That is because the values of the bar chart are way higher than the values of the scatter chart. The solution is to draw the scatter chart on another range axis. Just drag the scatter series

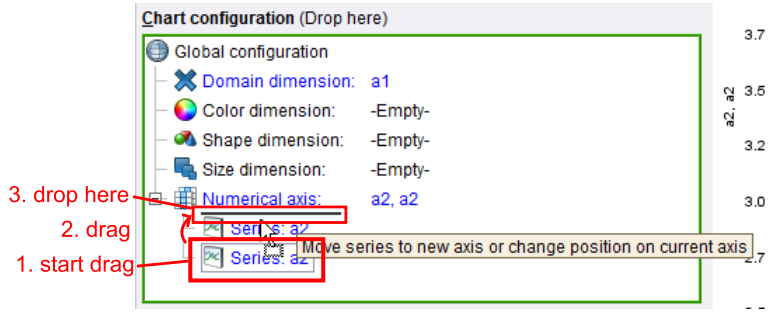

Figure 6.6: Changing the drawing order of the series.

configuration with the label *a2* to the bottom of the configuration tree. A popup menu appears, which gives you the choices to change the order of the series on the same axis, or to create a second axis. Since the latter is what you want, you choose *Create new axis*, and voilà, a new axis is born.

The axis does not only appear in the configuration tree, but also on the right side of the chart, and the scatter chart now fills the complete plot area. But again, it is drawn below the histogram. Looking at the configuration tree the reason should be obvious: the axis for the scatter chart, and thus the scatter chart itself, is located below the axis of the histogram. You can easily fix this by dragging the complete axis directly above the histogram axis. (Actually this was another small trap, you could have simply moved the histogram series configuration to the new axis in the first step and the order would have been correct.)

6.2.4 Adding Colors

We almost accomplished our goal of recreating the chart in figure 6.8. The only thing missing are colors. Before reading on, do you remember how to add colors to a scatter chart?

Exactly, we need to drop an attribute on the Color dimension. To replicate our targeted chart, the chosen attribute has to be the *label*. Drag it now onto the Color dimension. This produces a chart similar to figure 6.7. Even though this chart makes perfectly sense (see chapter 4.2) it is not quite what we want. What has happened?

If you drop an attribute on the Color dimension while at least one aggregated

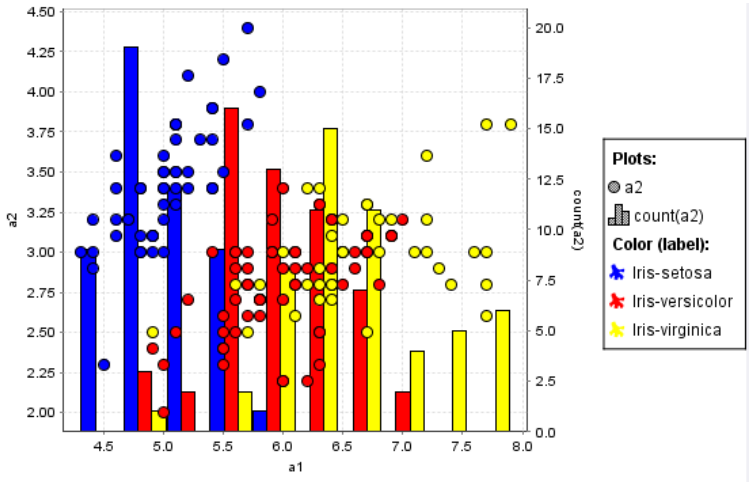

Figure 6.7: A grouped histogram on Iris, overlayed with a colored scatter chart.

series is present in the chart, a grouping is automatically configured for the Color dimension, and thus the bars are split up into different colors for each label value as described in chapter 4.2.

We, however, only want the scatter chart to be colored according to the color dimension. Thus we have to reset the grouping of the color dimension to *None*. If you do so now, your chart is almost identical to our targeted chart. For final polishing you could reduce the opacity of the bar chart in the *Format* configuration of the bar chart to make its appearance less intrusive. Since aggregated series don't use the ungrouped color dimension, you could also change the color of the bars here.

6.2.5 Removing Series and Axes

To remove a series or a complete range axis with all contained series, right click it in the configuration tree and select *Remove axis* or *Remove series*. You could also left click and hit the *Delete* key on the keyboard.

6.2.6 Three Steps to a Multi-Series Chart

1. Drag an attribute onto the range axis and configure the series.

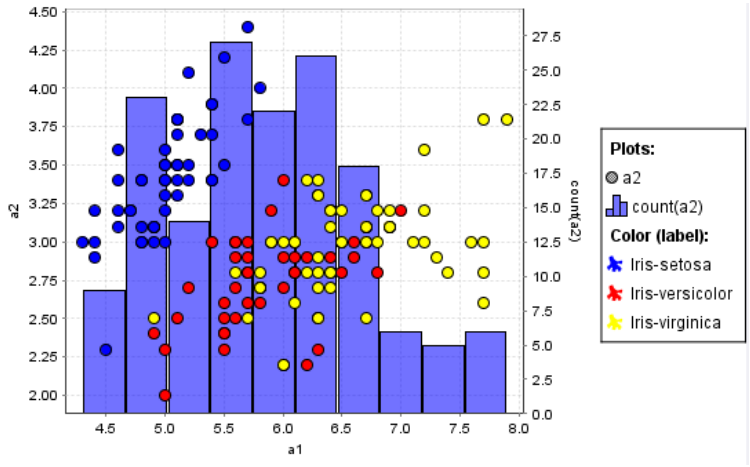

Figure 6.8: A histogram on Iris, overlayed with a colored scatter chart.

2. Drag one or more other attribute on this axis or create new axes and configure them.

3. Configure the drawing order of the series by dragging them around in the configuration tree.

6.2.7 Common Pitfalls and Useful Hints

You might find these hints and warnings useful when creating a mult-series chart:

- Each range axis can handle at most one nominal or categorical, unaggregated series. Thus it is not possible to draw e.g. a nominal label and another nominal attribute on the same axis.

- If you create a categorical grouping on the domain dimension, you can't create ungrouped series in the chart. Either configure an aggregation function for the series, or switch the grouping on the domain axis to not categorical.

- If your series covers only a small part of the chart, you probably put in on a range axis together with another series with a very different range. You should consider to create a separate range axis for the problematic series.

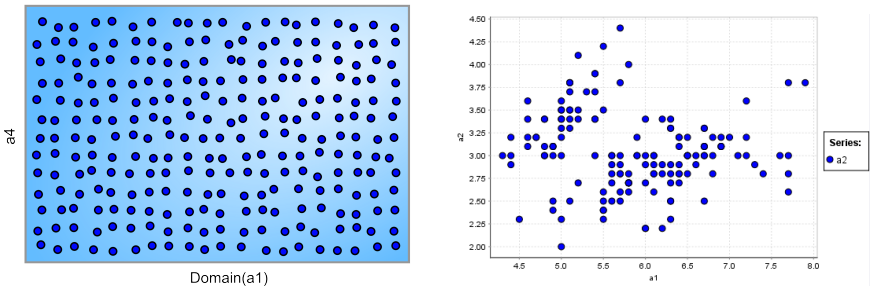

Figure 6.9: The dataset.

6.3 Advanced Grouping and Aggregation

6.3.1 Understanding Groupings and Aggregations

In section 4.2 we learned how to create a grouping on the Domain dimension and plot aggregated values. Additionally we grouped a series via the Color dimension. We did not yet dive into the depths of the grouping framework, nor did we use other dimensions than Domain and Color.

Let's examine the grouping framework on the example of the Domain and Color dimension, i. e. we will analyze what is going on with the data when we create a histogram similar to the one from section 4.2.

The left side of figure 6.9 shows an abstraction of the Iris dataset where *a4* is drawn against *a1*. On the right side of the figure, the same dataset is plotted, but in this case *a2* on the range axis versus *a1* on the domain dimension.

As soon as we drag attribute *a4* onto the Color dimension, the data points are drawn in the color according to *a4*, that means that the Color dimension assigns a color to each single data point according to its *a4*-value (see fig. 6.10). On the left side of the figure a nice gradient appears, since we put a continuous, i. e. numerical attribute on the Color dimension. On the right the points are also colored, but seemingly unordered, because the chart on the right is not sorted by *a4*.

Please note again that in the abstraction on the left *a4* is drawn versus *a1*, whereas the plot on the right shows *a2* versus *a1*.

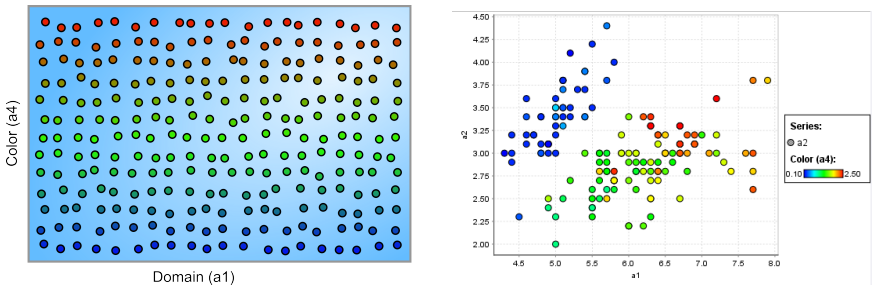

Figure 6.10: The dataset including the color dimension.

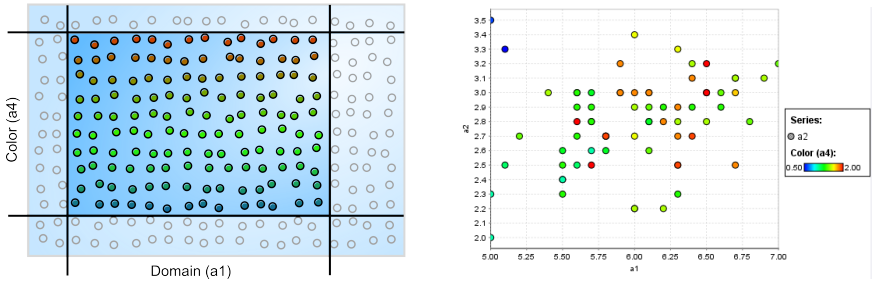

Figure 6.11: The dataset with applied ranges.

In the next step we define filters on both the Domain and the Color dimension. Thereby we remove some data from the plot, as depicted in figure 6.11. The filter on the domain translates directly into the actual plot, since the domain is visualized as the x-axis. The filter on the Color dimension results in the removal of some data points, namely those points whose value for *a4* is smaller than the lower filter value or greater than the upper filter value on the Color dimension.

The gradient on the left side only reaches from the lower filter bound to the upper filter bound. That shows that it is scaled accordingly to the filter values, i. e. such that the "maximal color" is reached at the upper filter bound and the "minimum color" at the lower filter bound.

Now that we understood multi-dimensional filtering, we create a grouping on the Domain dimension. To make use of the default equidistant fixed bin count grouping, all we have to do is to select an aggregation function in the series configuration. To create a histogram, the chosen function must be *count*.

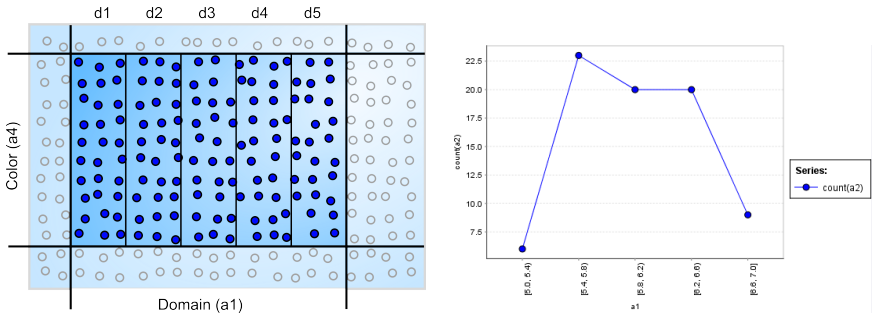

Figure 6.12: The dataset with applied ranges and domain grouping.

This results in 5 groups on the domain dimension as shown in figure 6.12. The aggregation function is calculated separately for each of the bins from all data points in that bin, and one point in the plot on the right represents one aggregated value (for a nicer visualization a solid line style has been configured in the series configuration). Up to now this is straight forward, but as a side effect all colors vanished from the plot and the points are shown in standard blue. Additionally the warning about the Color dimension not being used already known from figure 4.7 in section 4.2 appears in the notification area.

To understand the message you have to recall that an ungrouped Color dimension assigns a color to each single data point. But if you look again at figure 6.12 it becomes clear that each point on the right represents multiple data points of the data set on the left. Thus the Color dimension cannot provide a single color for the aggregated point. To get colors back, a grouping must be defined also on the Color dimension.

In this example we create an equidistant fixed bin count grouping with three bins on the Color dimension. As soon as the grouping is configured, the colors come back. Figure 6.13 shows that the grouping on the Color dimension divides each group of the Domain axis further down into three other groups. Now the Color dimension does not assign colors to single data points, but to each group it defines. Thus you don't see a continuous gradient anymore, but discrete colors for each group on the Color dimension.

In the chart on the right for each group on the Domain dimension one point for each contained Color group is created.

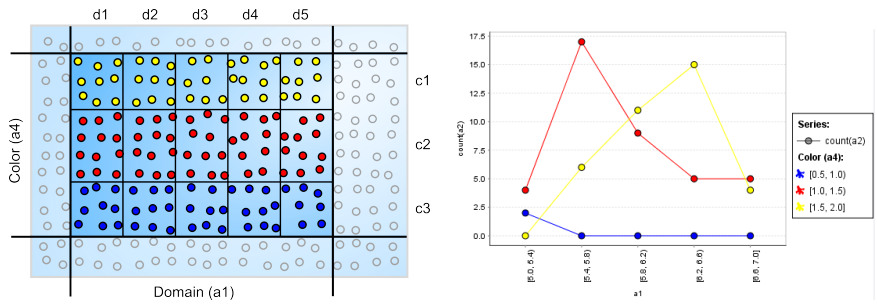

Figure 6.13: The dataset with applied ranges, domain and color grouping.

6.3.2 Configuring More Dimensions

In addition to the Color and Domain dimension RapidMiner's Advanced Charts support a Size and a Shape dimension. Both work the same way as the Color dimension in that they change the format of the data points. You can use them in a grouped or ungrouped fashion, but as with the Color dimension they must be grouped to be applied to aggregated series.

The Shape dimension has the constraint that it only supports nominal/categorical values, because it is impossible to represent continuous values via shapes. Nevertheless you can drag a numerical or date attribute onto the Shape dimension, if you configure a categorical grouping for it.

6.4 Drawing Error Indicators and Comparing Series

In this section you will learn how to create series with error bars, and how to graphically compare different series. For sample charts you may have a look at the final results of this chapter in figures 6.15, 6.16 and 6.18.

6.4.1 Error Bars and Error Bands

Error bars and error bands (or more generally indicator bars and bands) can be used to show additional information about a data point. A common use case is drawing the standard deviation for averaged values, and that is exactly what we are going to do in this section.

We start with preparing a scatter chart on Iris which contains the *label* on the Domain dimension and *a3* on the range axis. Set the aggregation function of the series *a3* to *average*, and for a nicer viualization set the line style to *Solid*, and set the lower bound of the range axis to 0 in the range axis configuration panel.

The final plot shows the average values of *a3* for each value of the label. To add indicator bars for the standard deviation, please switch back to the series configuration panel and set the *Indicator type* to *Bars*. The notification area now shows an error which states that no indicator attribute is defined. That basically means that the plotter complains that it does not know from which to generate the indicator bars. Since we want to draw the standard deviation of *a3*, we have to drag that attribute into the attribute field for *Indicator 1* (see fig. 6.14). The drop down list next to the attribute field allows you to select an aggregation function for the indicator attribute, but in our case the default *standard deviation* is fine. We leave *Indicator 2* empty for now. We also leave the option *Relative* enabled.

The result is a chart similar to figure 6.15. The points in the chart indicate the average of *a3* for the *label* value on the domain axis, and the error bars show the standard deviation of each point. It is drawn *relative* to the main value of the series, i. e. to the average. To draw the error bars the standard deviation is added to the average, and at the result value the upper error bar is located. For the lower error bar the standard deviation is subtracted from the average. Thus, we have *symmetric, relative* error bars.

Let us now investigate asymmetric error bars and error bands. These are not supported for categorical plots, so we have to place a numerical attribute on the Domain dimension. Drag attribute *a1* onto the Domain dimension, configure an equidistand fixed bin count grouping with 10 bins and make sure that it is *not* categorical.

Now the chart shows the average values of *a3* in each group of *a1* together with the standard deviation. Go back to the series configuration panel and change the *Indicator type* to *Band*. Now, instead of error bars, the standard deviation is painted as an area with the width of the standard deviation (see fig. 6.16).

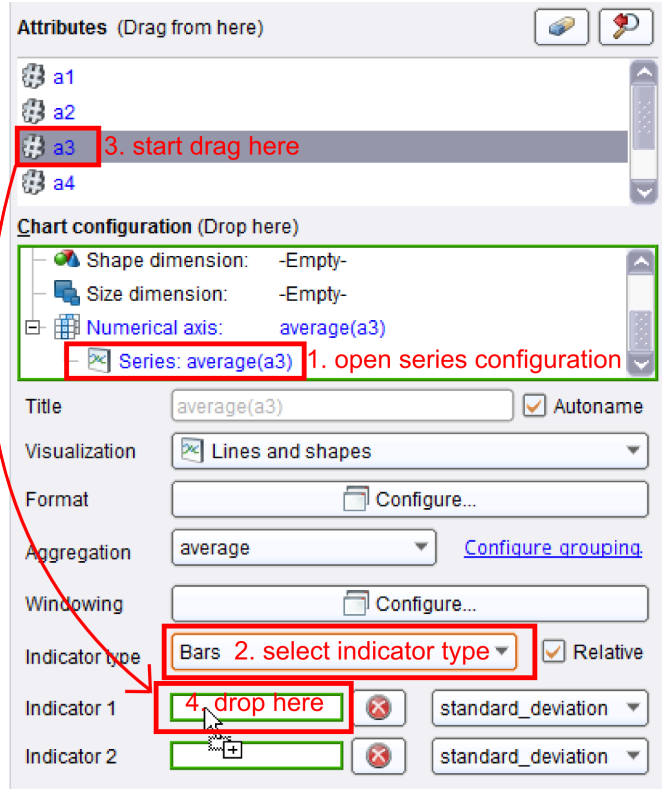

Figure 6.14: Configuring an error indicator.

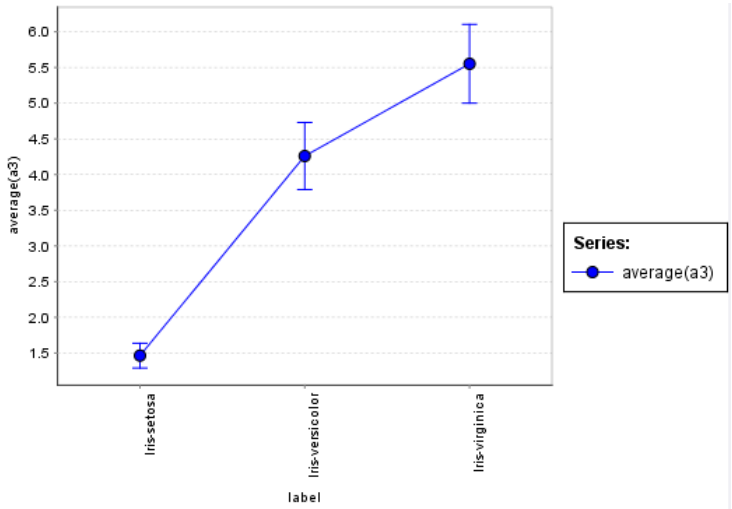

Figure 6.15: A chart with error bars.

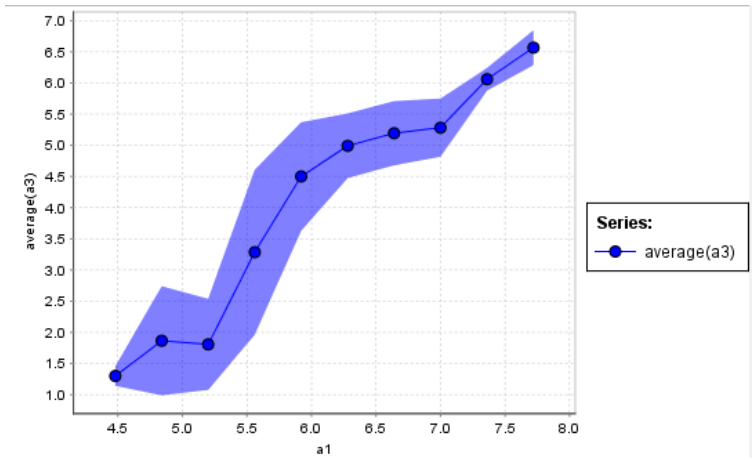

Figure 6.16: A chart with and error band.

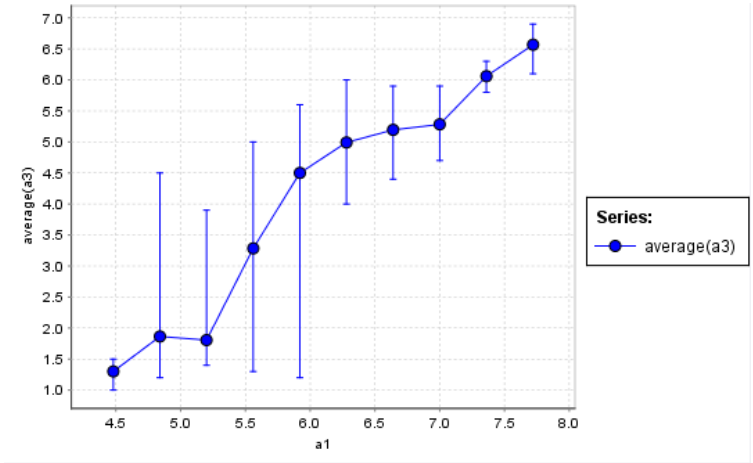

Figure 6.17: A chart with asymmetric, absolute indicator bars showing average, minimum and maximum value of each group.

Asymmetric Indicators and Non-Relative Values

Let us now draw indicator bars which does not visualize the standard deviation, but the minimum and maximum values in each group. So first of all set the Indicator type back to *Bars*. Then change the aggregation function of *Indicator 1* to maximum. Looking at the plot we spot two problems: since there is no *Indicator 2* attribute set, we still have symmetric error bars, and because the option *Relative* is still enabled, the maximum values are added/subtracted from the average value.

To solve the latter, simply disable the *Relative* option. Now the plot correctly shows the upper end of the error indicator. For the lower end just drop *a3* onto the *Indicator 2* attribute field and set its aggregation function to *minimum*. You now should end up with a chart similar to figure 6.17 which shows the average values as main series, and the minimum and maximum values as error bars.

6.4.2 Visual Comparison of Series

Comparison or *difference* charts are created analogously to to charts with indicators. Just drag the comparison attribute into the attribute field of *Indicator 1*, select its aggregation function and switch the *Indicator type* to *Difference*. *Indi-*

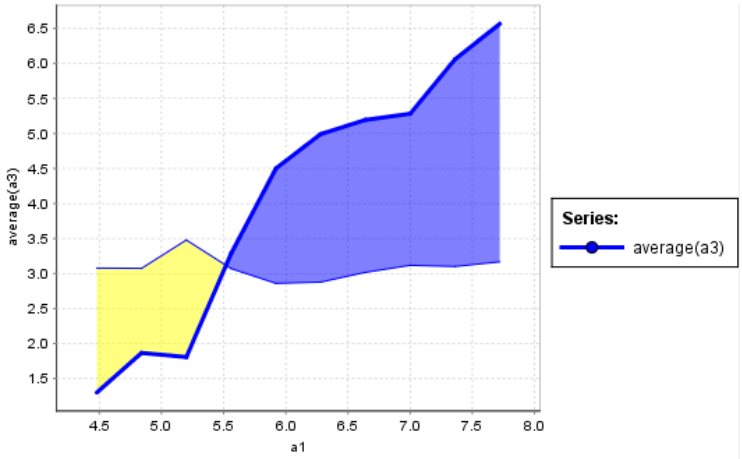

Figure 6.18: Difference plot between the averages of *a3* and *a2*.

cator 2 must be empty for difference charts.

In figure 6.18 the main attribute of the series is *a3*, *Indicator 1* is *a2* and both aggregation functions are set to *average*. Since the values of *average(a2)* are absolute values, the option *Relative* is disabled.

The chart shows one line for each of the two attributes, and the area between them is filled according to the sign of the expression *average(a3) - average(a2)*. If it is positive, the area is filled with the series color configured in the series format. If it's negative, the color is inverted.

To emphasize the main series, you can increase the line width in the series format.

Note: a series showing a difference cannot draw item shapes.

6.4.3 Six Steps for Drawing Indicators

This is the recipe to create a chart with indicators.

1. Create a series.
2. If necessary, configure an aggregation.
3. Set the desired indicator type.

4. For symmetric indicators, drag an attribute to *Indicator 1*. For asymmetric indicators drag the upper value onto *Indicator 1*, the lower value onto *Indicator 2*.

5. If the main series is aggregated, select an aggregation function for the indicators.

6. Enable or disable the *Relative* option, according to the plotted series.

6.4.4 Common Pitfalls and Useful Hints

When configuring indicators for a chart, a couple of things need to be considered and some errors must be avoided:

- The only supported indicator type for categorical plots (i. e. a nominal attribute or a categorical grouping on the Domain dimension) are *Bars*. The bars must be symmetric (i. e. *Indicator 2* is empty) and only support *Relative* values.

- Double check that you configured the correct attributes and aggregation functions for the main series and the indicators. Usually, for indicators of type *Bars* or *Band* the attribute is the same, but the aggregation functions differ. For a *Difference* chart usually the indicator attribute differs from the main series attribute, but the aggregation function is the same.

- If configured the correct attributes and aggregation functions, but the plot looks somehow wrong, check the *Relative* option.

- If you are using absolute values and exchange the main attribute, the indicators might be completely above or below the main series. In that case you also have to replace the indicator attributes.

6.5 Windowing

Windowing is an advanced grouping technique in RapidMiner's Advanced Charts, which allows you to use data points from one or more neighboring groups in addition to the points from the group itself to calculate the aggregation function. This allows you to draw cumulative histograms or calculate moving averages.

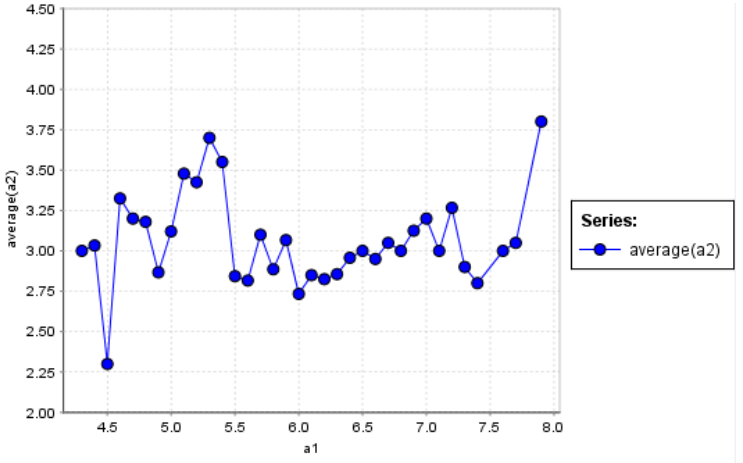

Figure 6.19: The average of *a2* versus *a1* without windowing.

6.5.1 Plotting the Moving Average

In this section we will illustrate the windowing mechanism on the example of plotting the moving average of a series. Here you will get a rough feeling for windowing. In the next section we will explain in detail what is happening with the data.

To begin, please load the Iris dataset or reset the chart configuration if it is already loaded. Than draw the average of *a2* versus *a1* with a *distinct values* grouping on the Domain dimension. Configure the line style to *Solid*. Then set the range of the range axis from 2 to 4.5. This will allow us later to concentrate on changes introduced by the windowing without getting distracted by changes of the automatic range. The result should be similar to figure 6.19.

The graph is quite jagged and volatile, so let's try to smooth it. Open the series configuration *average(a2)* and click the *Configuring* button next to *Windowing*. The windowing configuration popup shows up as shown in figure 6.20.

The popup show three options: *Grab left*, *Grab right* and *Allow incomplete groups*. If you enter a positive value n for *Grab left*, the average is not only calculated from the data points in the current group, but also from the points in the n groups left of the current group. Obviously, the left-most groups cannot grab groups from the left and are thus *incomplete*. By default, incomplete groups are ignored. To

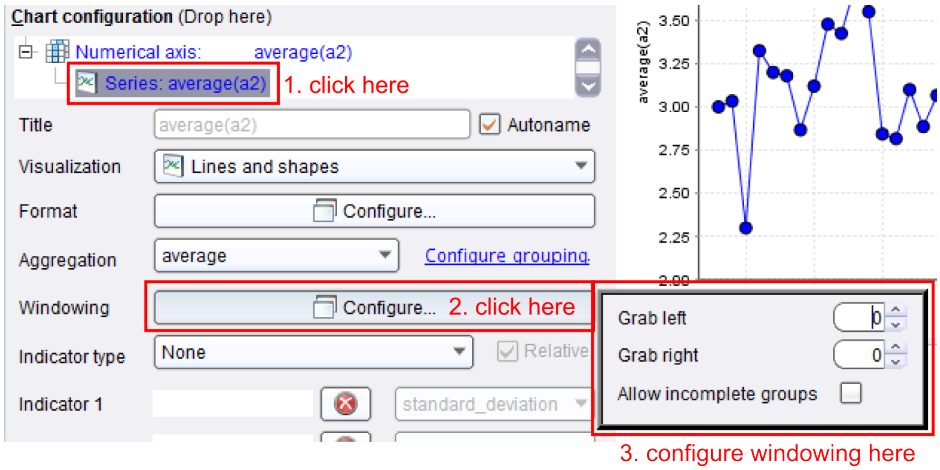

Figure 6.20: The windowing configuration popup.

show them, the option option *Allow incomplete groups* must be enabled.

Grab right works analogously to *Grab left*, but it grabs data points from the right. It is important to understand that the aggregation function is calculated from the *data points* of the current group itself and the grabbed groups, and it is *not* the average of the aggregated values.

After this paragraph of theroy, try out the windowing on the sample plot. Define a value of 1 for both *Grab left* and *Grab right* and observe the chart. Now the graph is much more smooth, since the averages are calculated on more data points. You may also have noticed that the left-most and the right-most data points disappeared. That is because they can't grab a further group from the left respectively from the right, and are thus incomplete. Show them again by enabling *Allow incomplete groups*. The final result is similar to figure 6.21.

You may want to play around with the settings to get a feeling for the windowing.

6.5.2 Cumulative Histograms

In the last section we used windowing for smoothing the average function. Another application for windowing is creating cumulative histograms, or empirical distribution functions.

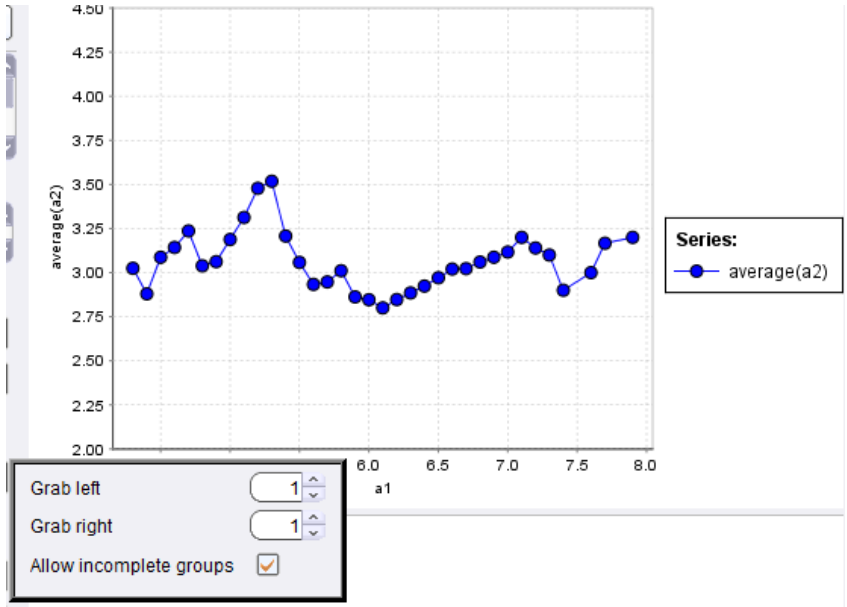

Figure 6.21: The windowing-smoothed average of *a2* versus *a1*.

Start by loading the Iris dataset or resetting the current chart configuration. Then create a bar chart of *a2* on the range axis versus *a1* on the Domain dimension with an equidistant fixed bin count grouping with 20 bins and the *count* aggregation on the series. The resulting chart in figure 6.22 is a histogram and similar to the empirical density distribution.

To make the histogram cumululative, open the windowing configuration for the series and enter the special value -1 for *Grab left*. -1 instructs the windowing engine to use all data points from the current group itself and *all* groups left of it to calculate the aggregation function. For the *count* aggregation that means that it counts all data points left of the current group, and thus the final group has the value 150, which is the total amount of data points in the Iris dataset. This cumulative histogram is similar to the empirical distribution function. It is shown in figure 6.23.

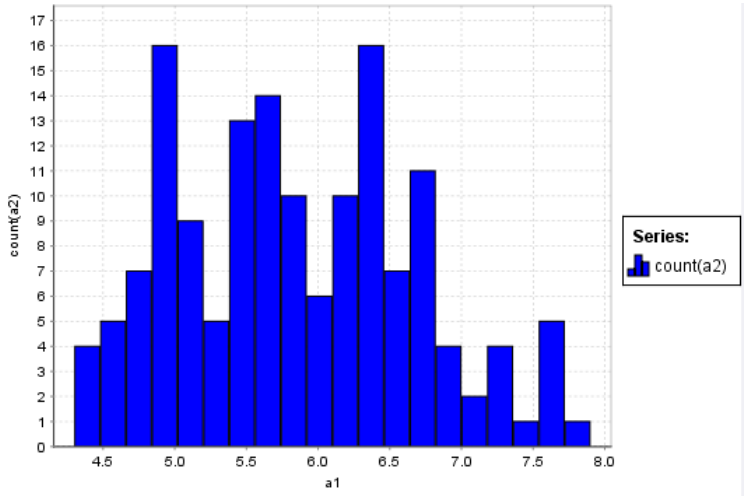

Figure 6.22: A standard histogram.

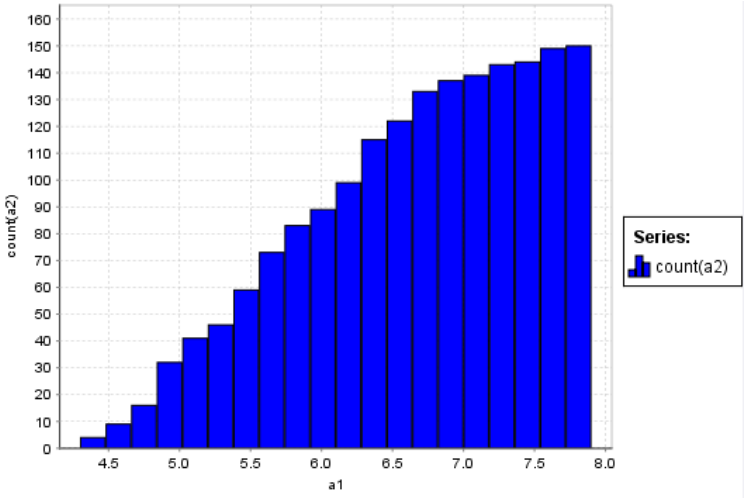

Figure 6.23: A cumulative histogram created with the windowing technique.

7 Wrap-up: Creating a Lift Chart

In this section we use almost everything we learned about RapidMiner's Advanced Charts to create a lift chart. The necessary steps are only drafted and the reader is recommended to execute all necessary steps, if necessary by re-reading the according sections.

A lift chart visualizes the classification behaviour of a model. The prediction confidence is usually placed on the Domain dimension and grouped such that each group contains 10 % of the data, where the groups are sorted by confidence. The lift chart is composed of two series. The first one is a histogram counting how many positive and negative examples are detected in each group. Usually, the groups on the left are almost pure and contain only positive examples, and when you move to the right the intervals contain more and more negative examples.

If you apply a threshold at a confidence value between two groups and classify all examples left of that threshold as positive, you can estimate how pure the dataset will be by comparing the area of the "positive" bars with the area of the "negative" bars. The absolute amount of positive respectively negative examples at a given threshold value is calculated by summing up the heights of all positive or negative bars left of the value.

That results in the second series of the lift chart, a cumulative graph visualizing the absolute values of positive or negative values. For an example of a lift chart have a look at figure 7.4.

What can you learn from such a chart? Assume that in the aforementioned chart we choose a threshold value between the third and the fourth group at 0.06. The bars in the groups left of the threshold are all red, that means that all values predicted as positive are truly positive. The thick red line shows us that we would grab approximately 3 000 (value at threshold) out of a total of

5 000 positive examples (value at the far right), and no negative examples (blue line). If we wanted to detect more positive examples, we would have to move the threshold one group to the right to 0.2. Now from the red line we learn that we would detect about 3 700 positive examples, but also 300 negative examples (blue line).

Thus the lift chart helps the data miner to estimate the trade-off between thresholds for very pure models and thresholds which detect more positive examples, but produce less pure datasets.

In the following sections we will guide you through the creation process of such a chart.

7.1 Preparing the Data

Since the lift chart is based on data classified by a model, we need to create a process which creates and applies a model. The process in figure 7.1 will do the job. It consists of two *Generate Data* operators, a *Naive Bayes* and an *Apply Model* operator. The *Geneate Data* operators are configured identically: for both the *target function polynomial classification* is selected and *number examples* is set to 10 000. The other paramters are left at their default values.

On the output of the first *Generate Data* operator a *Naive Bayes* model is learned, which is then applied to the output of the second *Generate Data* operator. The labelled data is passed from *Apply Model* to the result output.

Run the process and switch to the *Advanced Charts* view of the resulting example set. Besides other attributes you will see the attributes *label*, *confidence(positive)*, *confidence(negative)* and *prediction(label)*. Those are the only ones needed for creating a lift chart.

7.2 Creating the Histogram

We will start with the histogram which counts the number of positive and negative examples in each group. Drag attribute *confidence(negative)* onto the Domain dimension and configure an *Equal data fraction grouping*. Configure 10 bins such

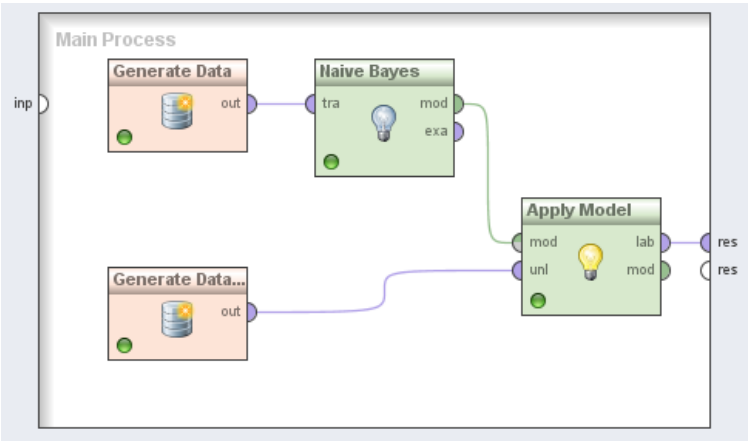

Figure 7.1: The process which creates classified data for creating the lift chart.

that each bin contains 10 % of the data.

Then create a series for the histogram. Since we will just count the examples, the actual attribute does not matter, but we will go with *label*. Set the visualization type to *Bars* and set the aggregation function to *count*. You should now see a rather boring histogram where each bar ends at 1 000, since the equal data fraction grouping places 1 000 examples, i. e. 10 % of the data, in each group. But in the introduction of this section we said that we need a histogram counting the number of positive and negative examples separately, so we have to split up the histogram by the label. Surely you remember from section 4.2 that you can do that by placing the *label* on the Color dimension and specifying a *distinct values* grouping. Do so now and watch the resulting chart (see fig. 7.2).

It shows the amount of positive and negative examples per confidence interval. For low confidences of the negative class (left side of the histogram) the groups contain far more positive than negative examples. The higher the confidence for the negative class, the more negative examples are contained in the groups and the ratio of positive examples decreases. Because of the nature of the equal data fraction grouping, the blue bars and the red bars sum up to the same value, 1 000, in each group.

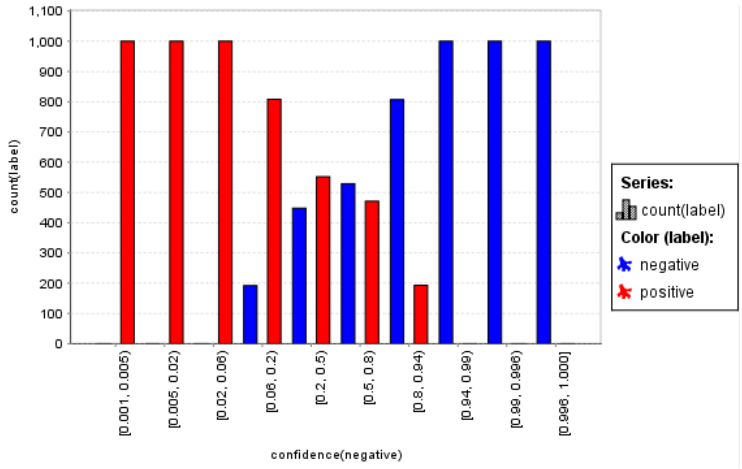

Figure 7.2: The histogram of the lift chart.

7.3 Adding the Cumulative Example Counts

For the cumulative example counts we create a second range axis with a series for the *label* attribute (again the actual attribute doesn't really matter, since we will use the *count* aggregation). Set the aggregation function to *count*, and configure the format such that you only see lines, but no shapes. Up to now the plotted values of this series are equal to the values from the histogram (if you are wondering why they actually seem to be higher than the histogram, have a look at the range axes). Now configure the windowing to sum up the values from left to right. Your plot should then look like the one in figure 7.3).

7.4 Polishing

The chart could need some visual polishing. Adjust line width, opacity and drawing order of the series to format the chart similar to the one in figure 7.4. Also make sure that both plots visually start at the domain axis (remember ranges), and adjust the axis and legend labels.

Congratulations! You are now capable of mastering RapidMiner's Advanced Charts. The Rapid-I team wishes you a lot of fun while visualizing your data.

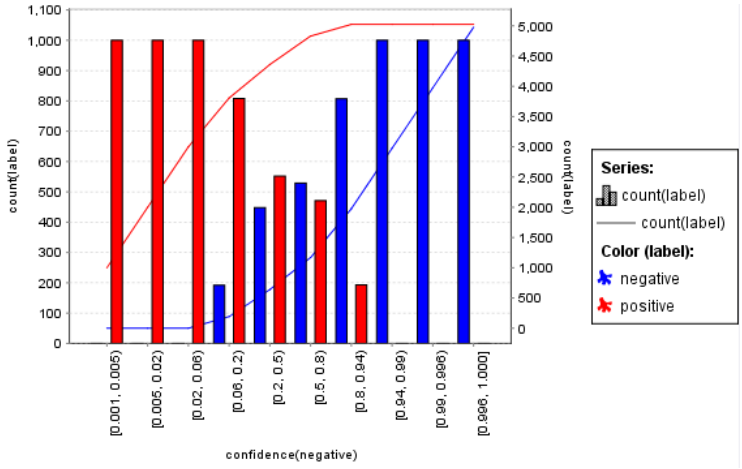

Figure 7.3: The complete, but unpolished lift chart.

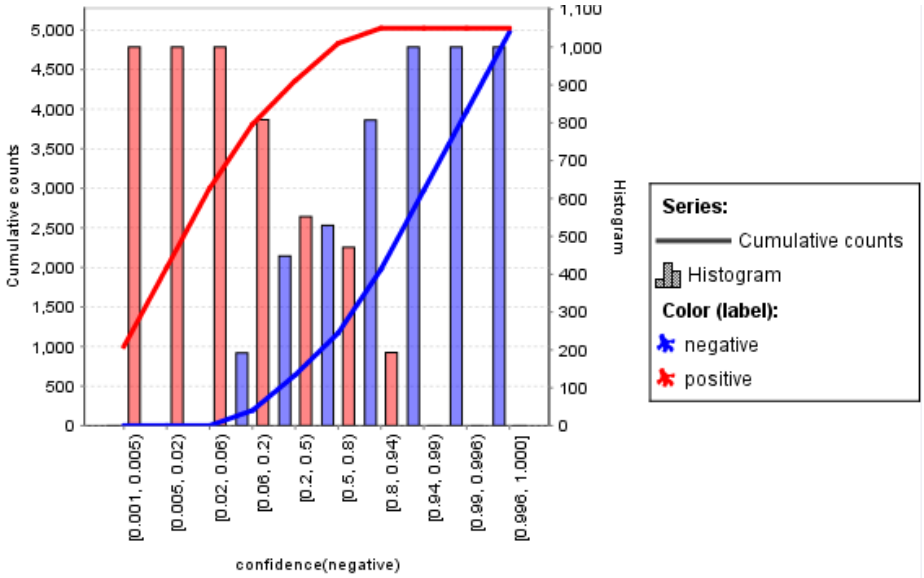

Figure 7.4: The final lift chart.

Rapid-I GmbH
Stockumer Str. 475
D-44227 Dortmund
Tel.: +49 (0) 231 425 786 90
E-Mail: contact@rapid-i.com
www.rapid-i.com