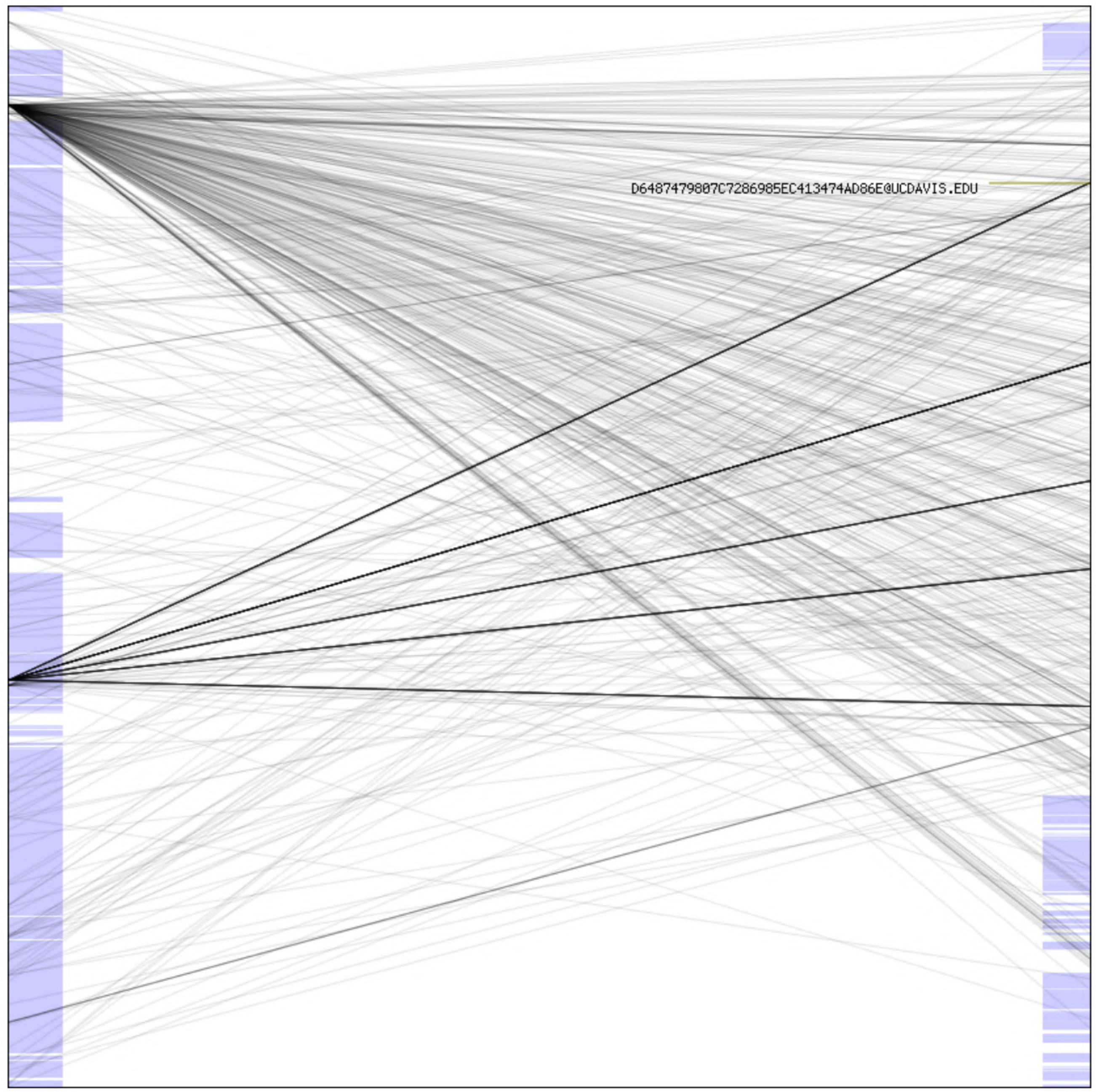
Interactive Machine Learning

Motivation

- Wait, why make machine learning interactive?
- Major purpose of ML: **automation**.
- What is the role of the human in ML?

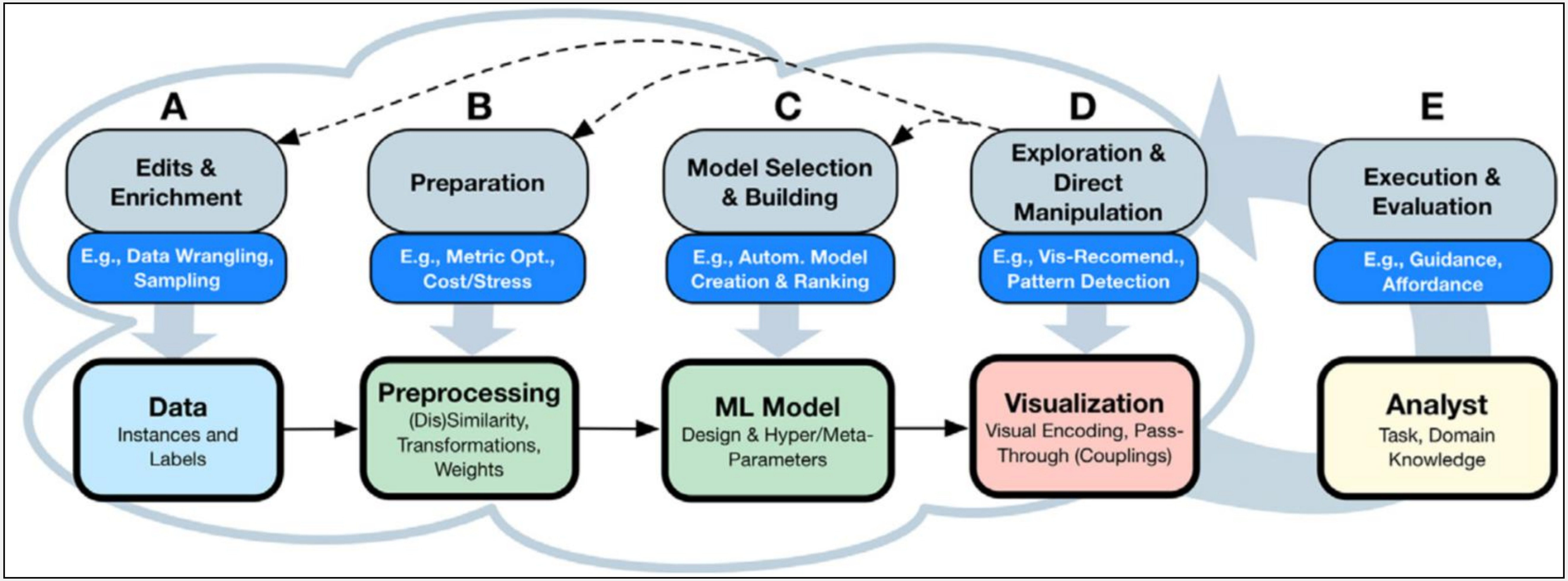
Example: Visually Detecting Spam Emails



[Muelder & Ma 2007]

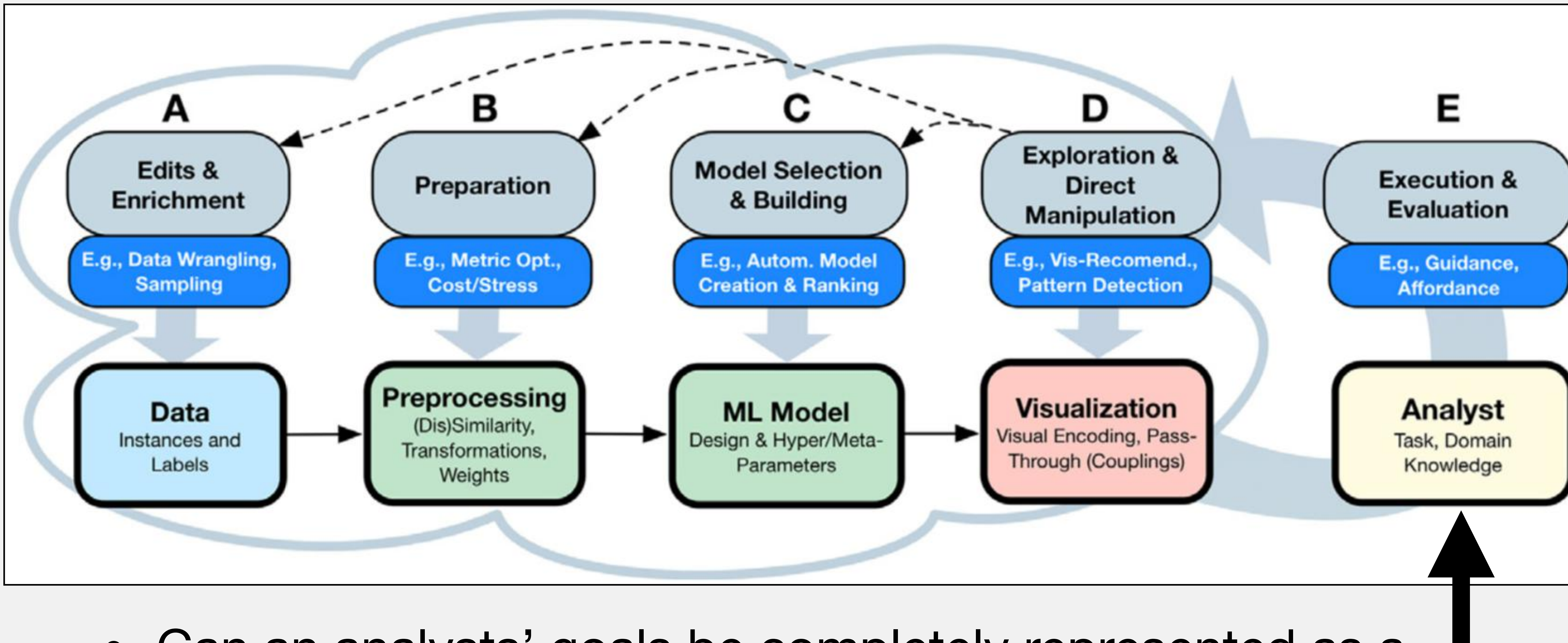
No Unjustified Visualization

- Human involvement needs to be warranted.
- So: in what ways should a human interact with machine learning?



[Sacha et al. 2017]

Goals of Analyst



- Can an analysts' goals be completely represented as a machine learning problem?
  - Data : Model output : Objective function

Example: Spam Filtering

- Data
  - Raw text of body, title, sender ; labels (spam / no spam)
- Model output: is this email spam?
- One possible model: <u>Naive Bayes</u>

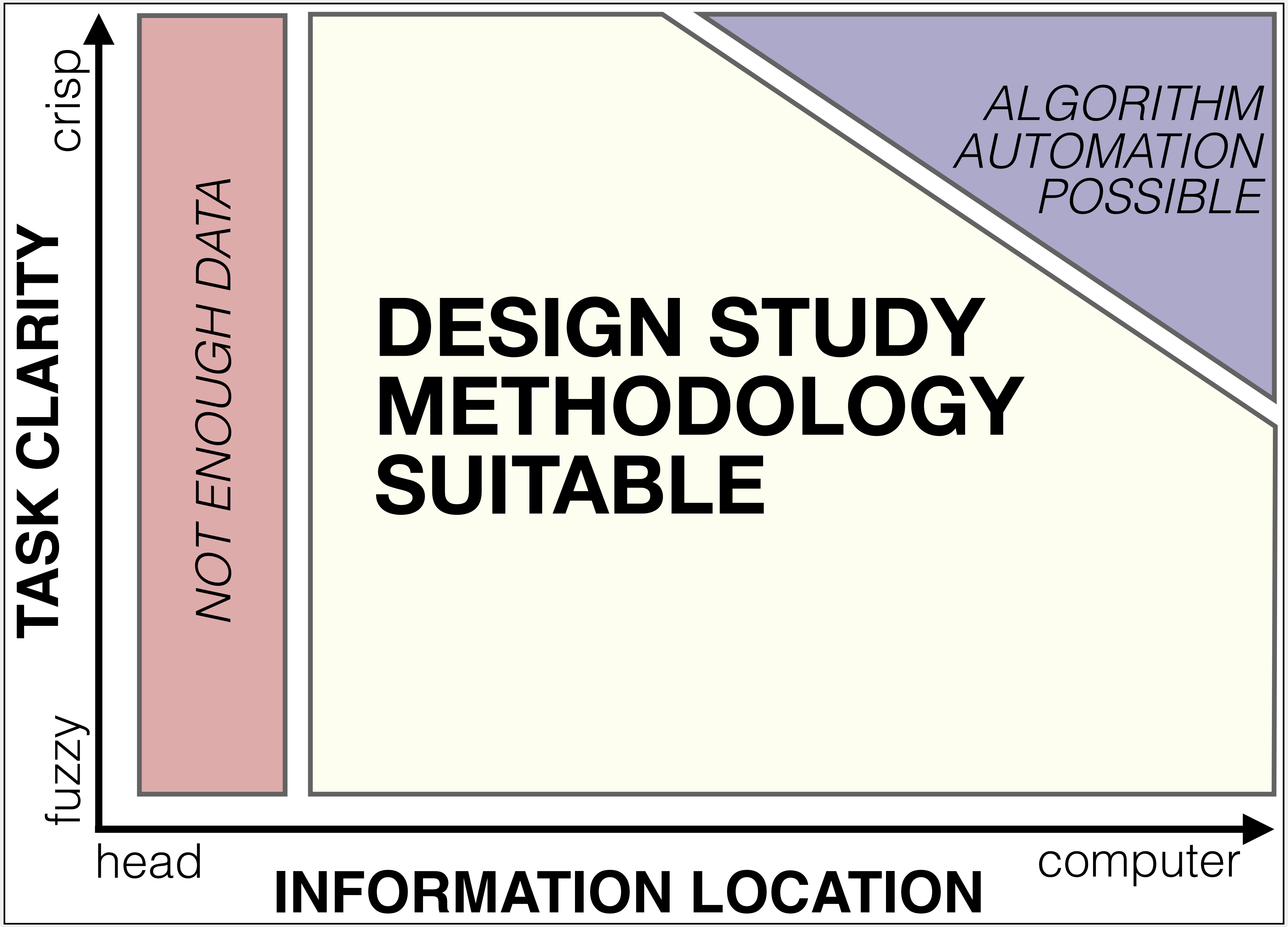
$$p(s \mid \mathbf{e}) \propto p(s) \prod_i p(e_i \mid s) \qquad \big(p(e_1, e_2, \ldots, e_n \mid s) = p(e_1 \mid s) \cdot p(e_2 \mid s) \cdot \ldots \cdot p(e_n \mid s)\big)$$

- Maximum Likelihood Estimation: **counts**

$$\hat{p}(s) = \frac{n_s}{n} \qquad \hat{p}(t \mid s) = \frac{n_{st}}{\sum_{t' \in V} n_{st'}}$$

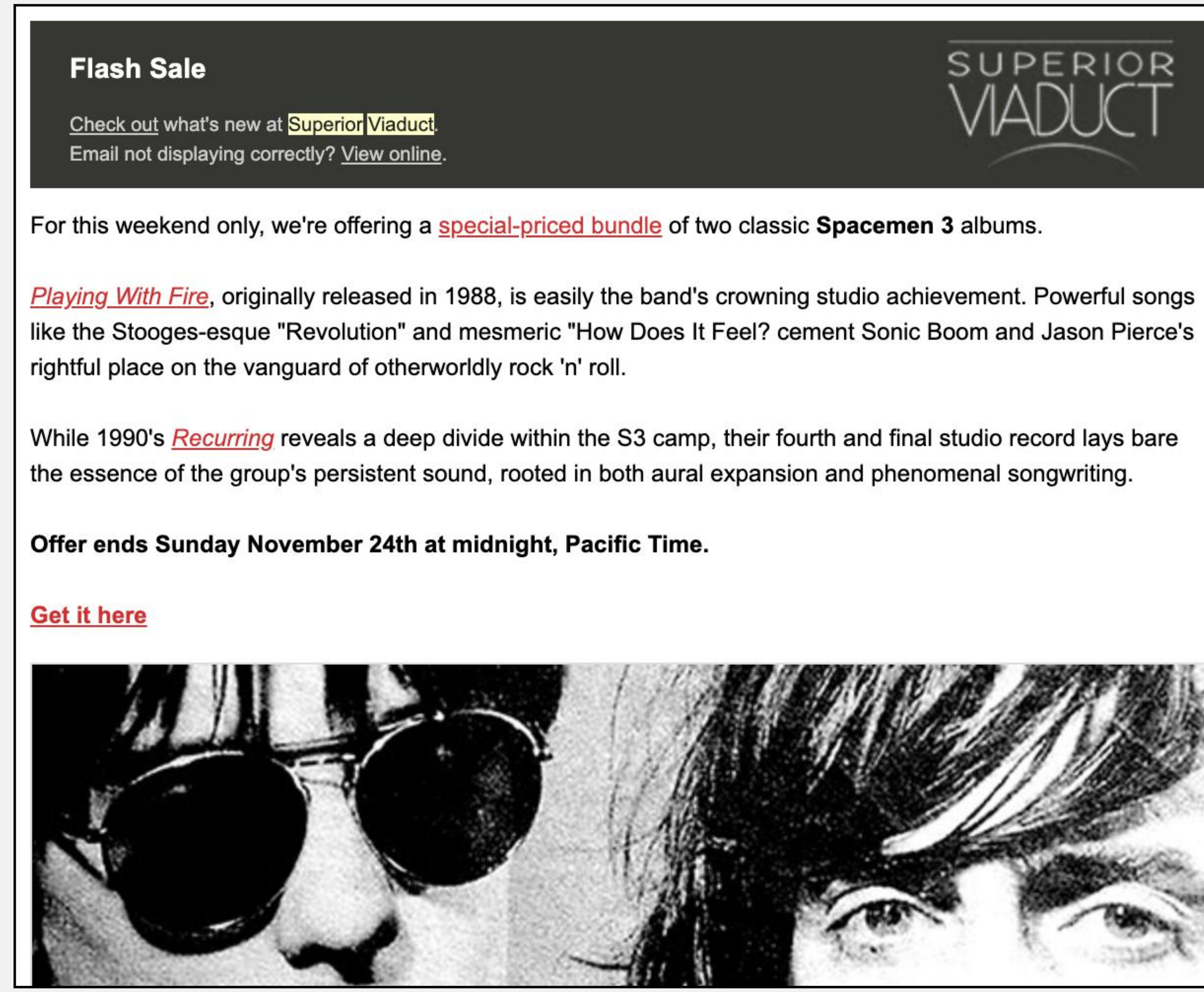
Test time: iterate over terms, sum up (log) probabilities. *Done and done.*

Not all goals are so clear...



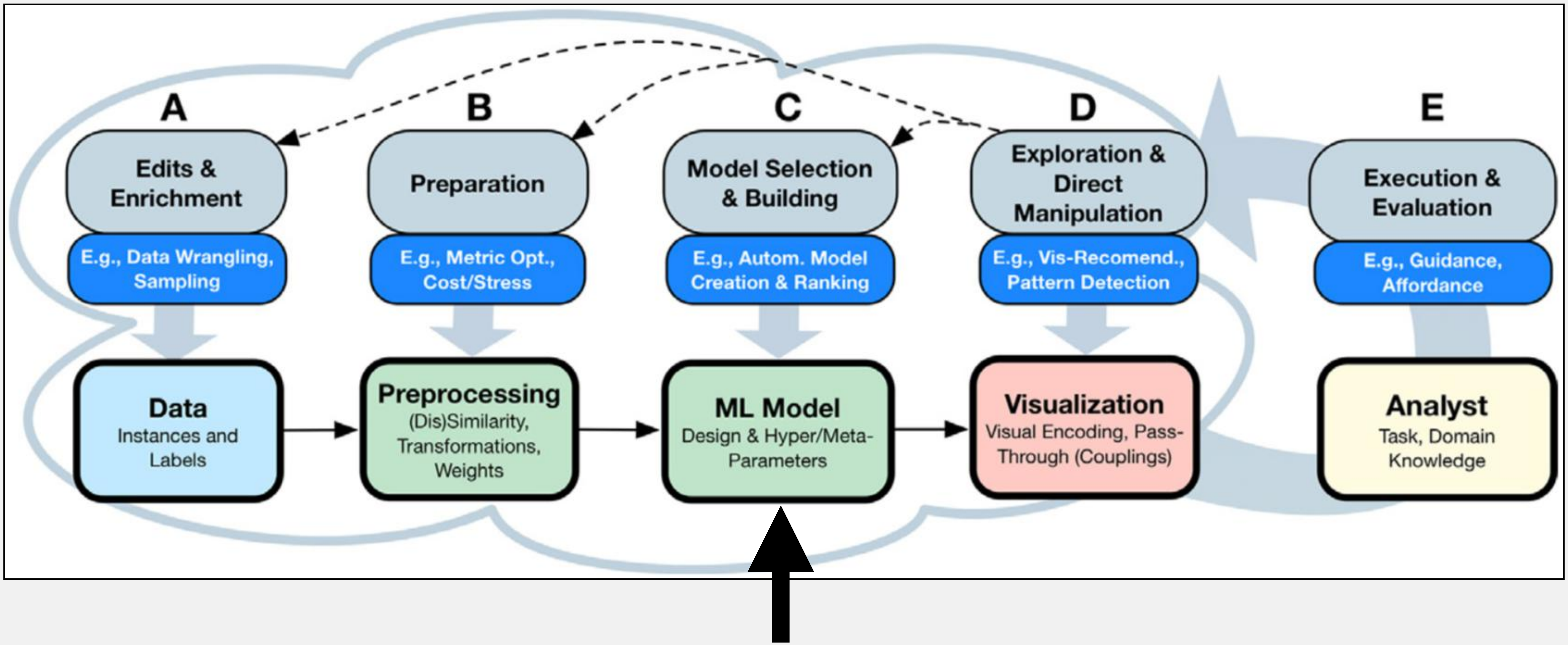
[Sedlmair et al. 2012]

Is all Spam Created Equal?



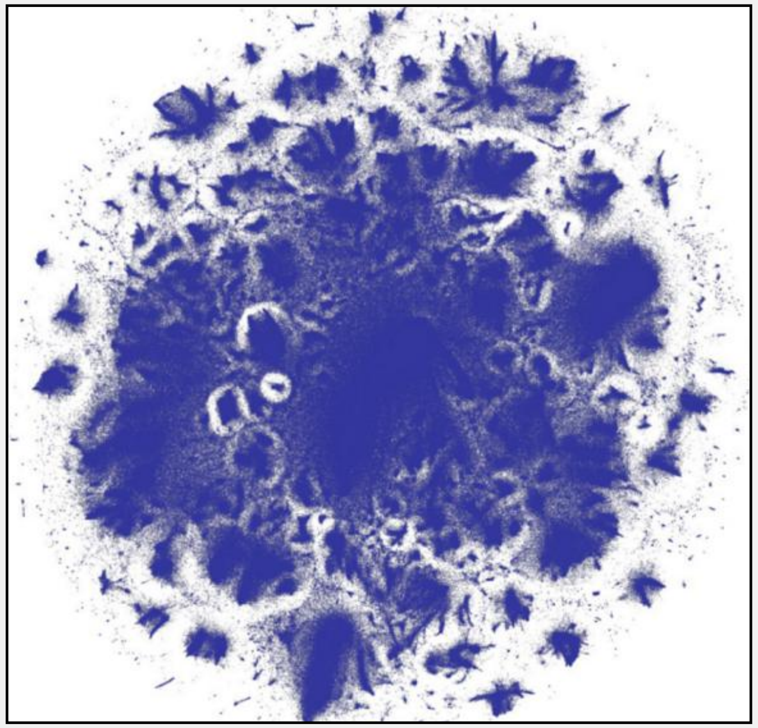
- Can we frame this as a supervised learning problem?
- Machine learning: support visual exploration.

Models

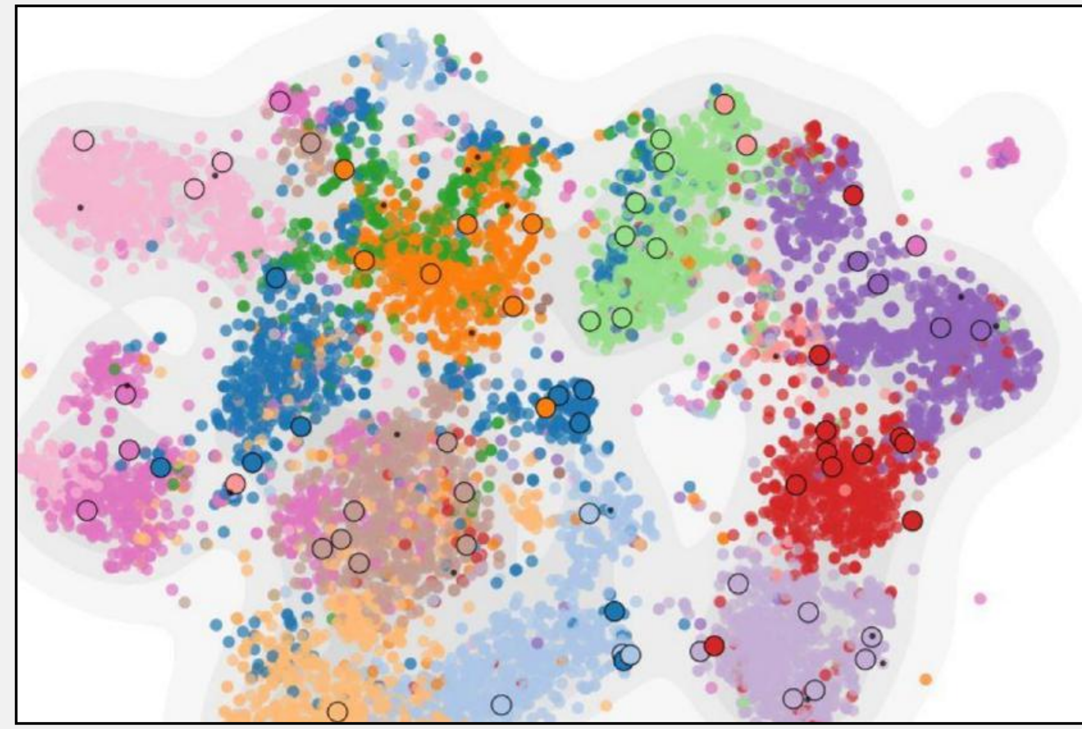


- Mixed-Initiative Visual Exploration: dimensionality reduction, clustering, topic modeling.
- Integrating a model with a visual interface: *need to know model specifics*.

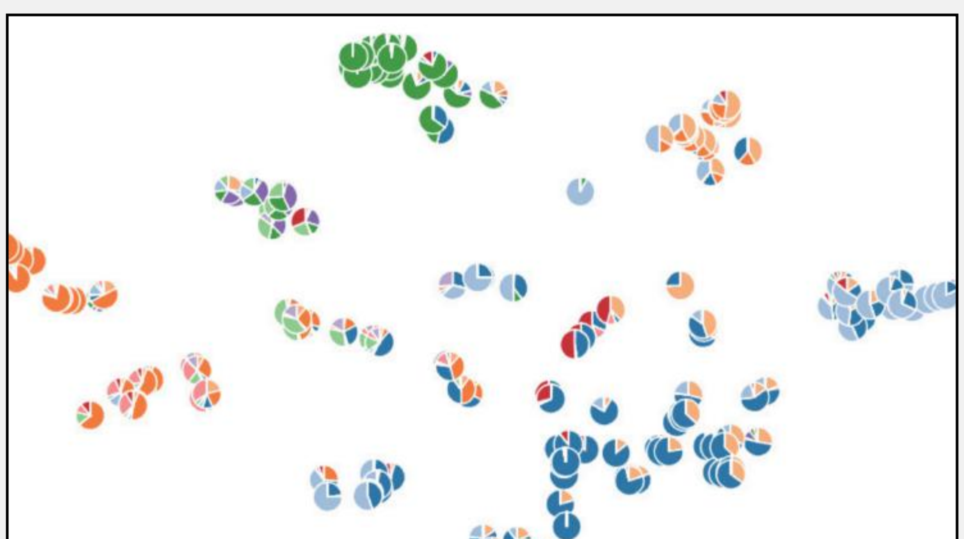
Dimensionality Reduction



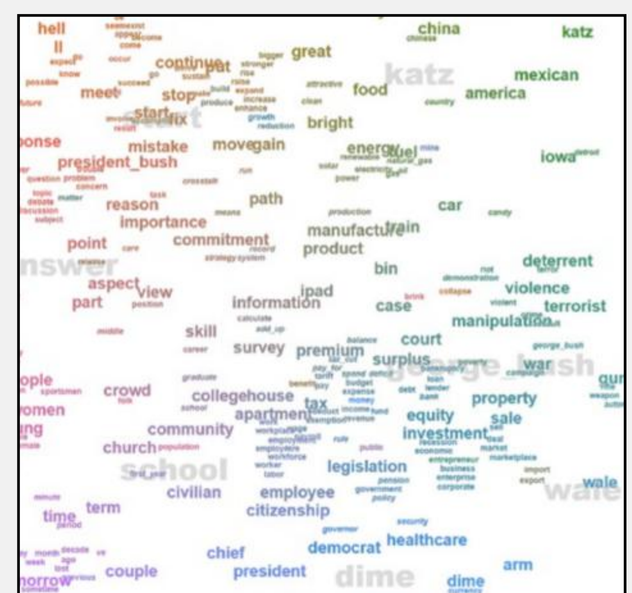
[Pezzotti et al. 2019]



[Xiang et al. 2019]



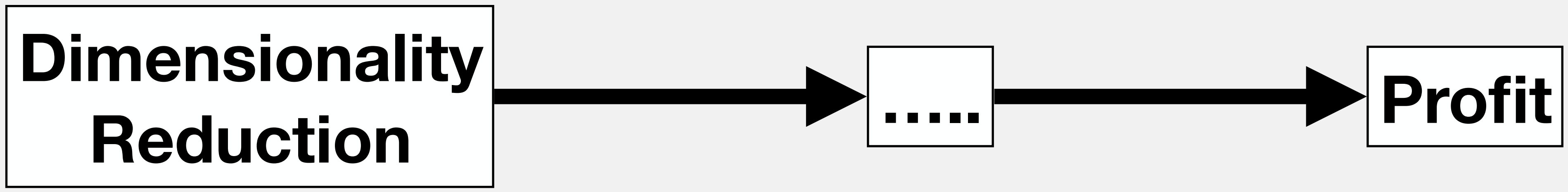
[Chen et al. 2019]



[El-Assady et al. 2019]

Dimensionality Reduction

- Arguably the most common technique in use for visual analytics approaches.
- Data: high-dimensional, typically quantitative
- Output: low-dimensional, typically 2D, projection



- What can go wrong? Why have a human in the loop?

Principal Component Analysis

- Mathematical preliminaries

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \quad \mathbf{x}_i \in \mathbb{R}^d$$

- Objective of PCA: find directions that, upon projection, *maximize variance of data*

$$\max_{\mathbf{v}_1} \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{v}_1^\intercal (\mathbf{x}_i - \mu) \right)^2$$

- We first compute the sample mean, and subtract it from our data, resulting in our data being zero-mean.

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \qquad \mathbf{x}_i \leftarrow \mathbf{x}_i - \mu$$

- Obtain the following: $\frac{1}{n} \sum_{i=1}^{n} (\mathbf{v}_i^\intercal \mathbf{x}_i)^2 = \frac{1}{n} \sum_{i=1}^{n} \mathbf{v}_1^\intercal \mathbf{x}_i \mathbf{x}_i^\intercal \mathbf{v}_1 = \mathbf{v}_1^\intercal \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\intercal \right) \mathbf{v}_1$

PCA Continued..

- Covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\intercal \longrightarrow \max_{\mathbf{v}_1} \mathbf{v}_1^\intercal C \mathbf{v}_1$ **s.t.** $\mathbf{v}_1^\intercal \mathbf{v}_1 = 1$

- Solution: eigenvalue problem $C\mathbf{v}_1 = \underline{\lambda_1}\mathbf{v}_1$ **s.t.** $\mathbf{v}_1^\intercal \mathbf{v}_1 = 1$

Lagrange multiplier, formed from constrained optimization problem

- More broadly: all components found as eigenvectors

$CV = V\Lambda$ **s.t.** $V^\intercal V = I$

$V \in \mathbb{R}^{d \times 2}$ **eigenvectors: our projection**

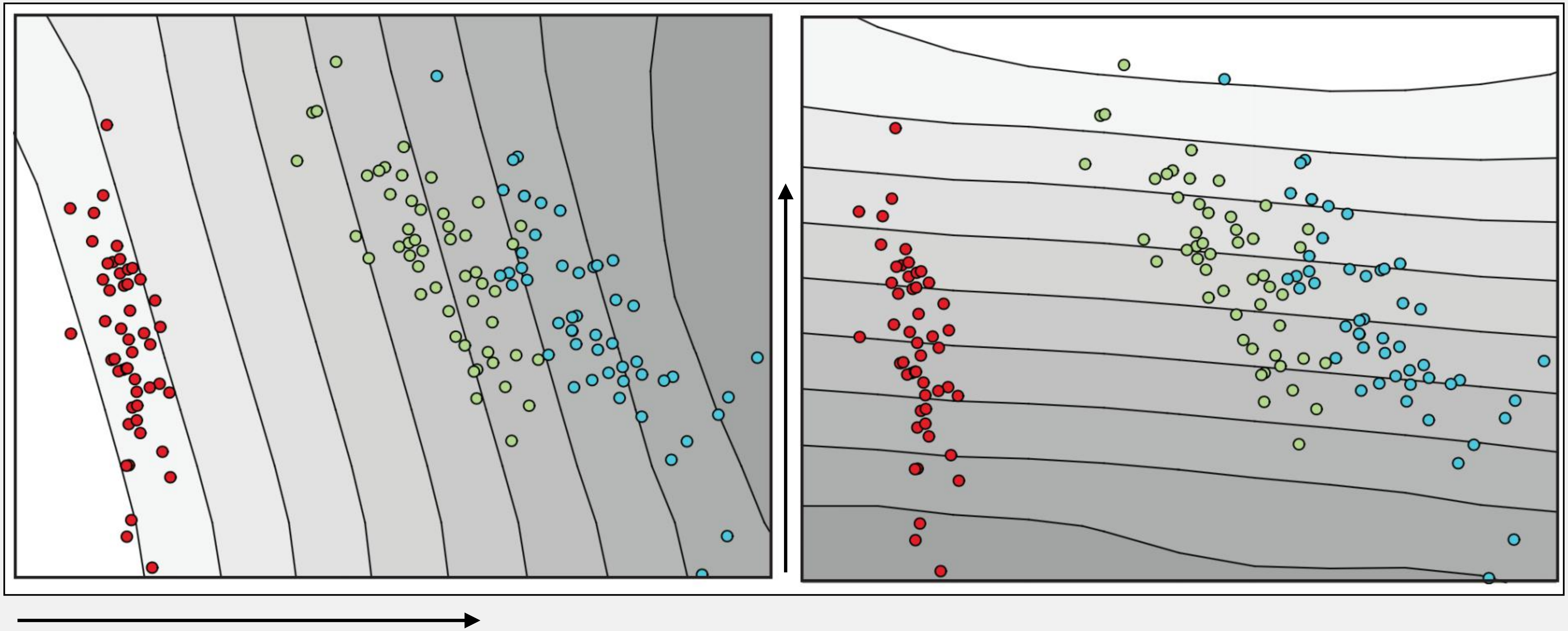
$\Lambda \in \mathbb{R}^{2 \times 2}$ **eigenvalues: measure captured variance**

- Dimensionality reduction: projection onto eigenvectors

$$\mathbf{p}_i = V^\intercal \mathbf{x}_i$$

Interpreting PCA

[Faust et al. 2018]



We perturb a data point along first eigenvector, then in 2D projection, it will move orthogonal to the y-axis

We perturb a data point along second eigenvector, then in 2D projection, it will move orthogonal to the x-axis

👍

Limitations with PCA?

- When is the projection “good”?

$$\lambda_i = \max_{\mathbf{v}} \mathbf{v}^\intercal C \mathbf{v} \textbf{ s.t. } \mathbf{v}^\intercal \mathbf{v} = 1 \ , \ \mathbf{v}^\intercal \mathbf{v}_j = 0 \ , \ j < i$$

- So eigenvalues help quantify goodness

- Another view of PCA: reconstruction

$$\min_V \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \underbrace{VV^\intercal \mathbf{x}_i}_{\textbf{linear projection}}\|^2 \quad V^\intercal V = I$$

$V^\intercal \mathbf{x}$ **low-dimensional representation**

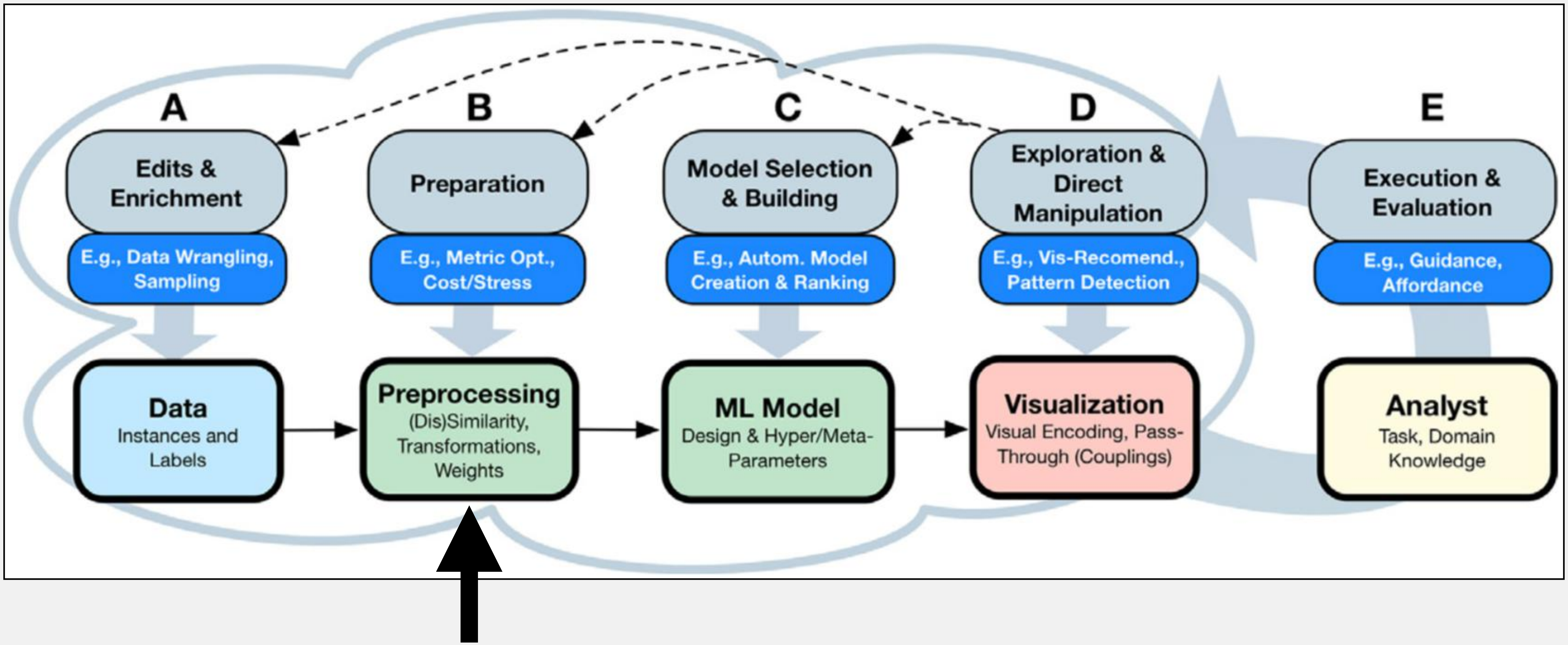
$\underbrace{VV^\intercal \mathbf{x}}_{\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2}$ **projection back to subspace**

- When data does not lie near a 2-dimensional subspace, we incur error

Visualizing PCA?

- What are potential issues in visualizing PCA projection via a scatterplot?
  - We fail to see visual patterns in the projection, but salient structures may exist in the original data!
  - We see patterns that are an illusion, do not exist in the data!
  - More broadly: a disconnect between 2D distribution and original data distribution
- Issue: does the data meet the model's assumptions?

Data Preparation



- Transformations that we apply to our data.
- Modify what the model optimizes.

Back to PCA

- How may we *modify* PCA?
  - Assign weights to data points
  - Assign weights to data attributes
  - Rescaling data attributes (e.g. unit variance)
  - Regularization (outliers!)

Weighted PCA

- Let's associate a weight with each of our data points, which tells us the *importance* of a data point

$$(\mathbf{x}_i, w_i)\ ,\ w_i \in [0,1]$$

- We can modify PCA to be sensitive to the weights: low-weighted points should have less influence on the projection

$$C = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\intercal \longrightarrow C_w = \frac{1}{n}\sum_{i=1}^{n} w_i^2 \mathbf{x}_i \mathbf{x}_i^\intercal \qquad \left(\mu_w = \frac{1}{\sum_i w_i}\sum_{i=1}^{n} w_i \mathbf{x}_i\right)$$

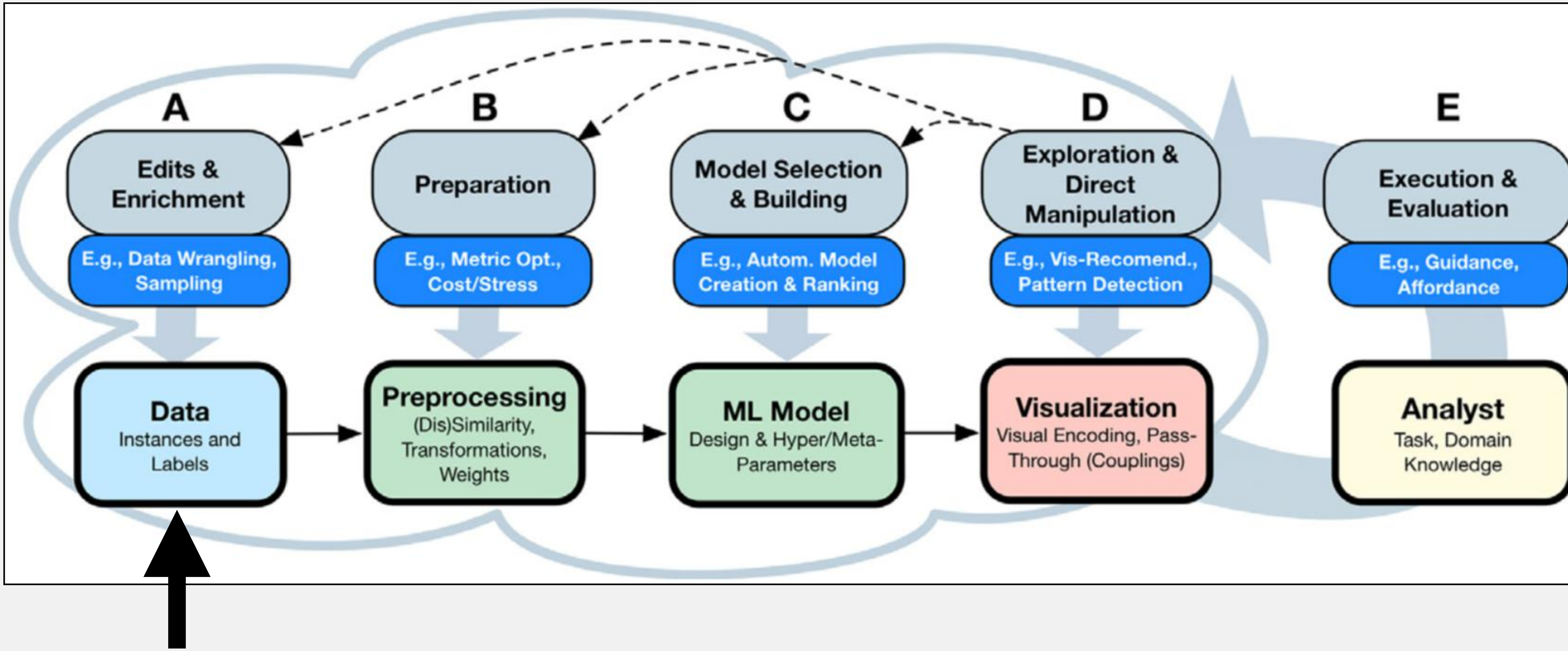
- Weighted reconstruction:

$$\min_V \frac{1}{n}\sum_{i=1}^{n} w_i^2 \|\mathbf{x}_i - VV^\intercal \mathbf{x}_i\|^2 \quad V^\intercal V = I$$

Obtaining Weights

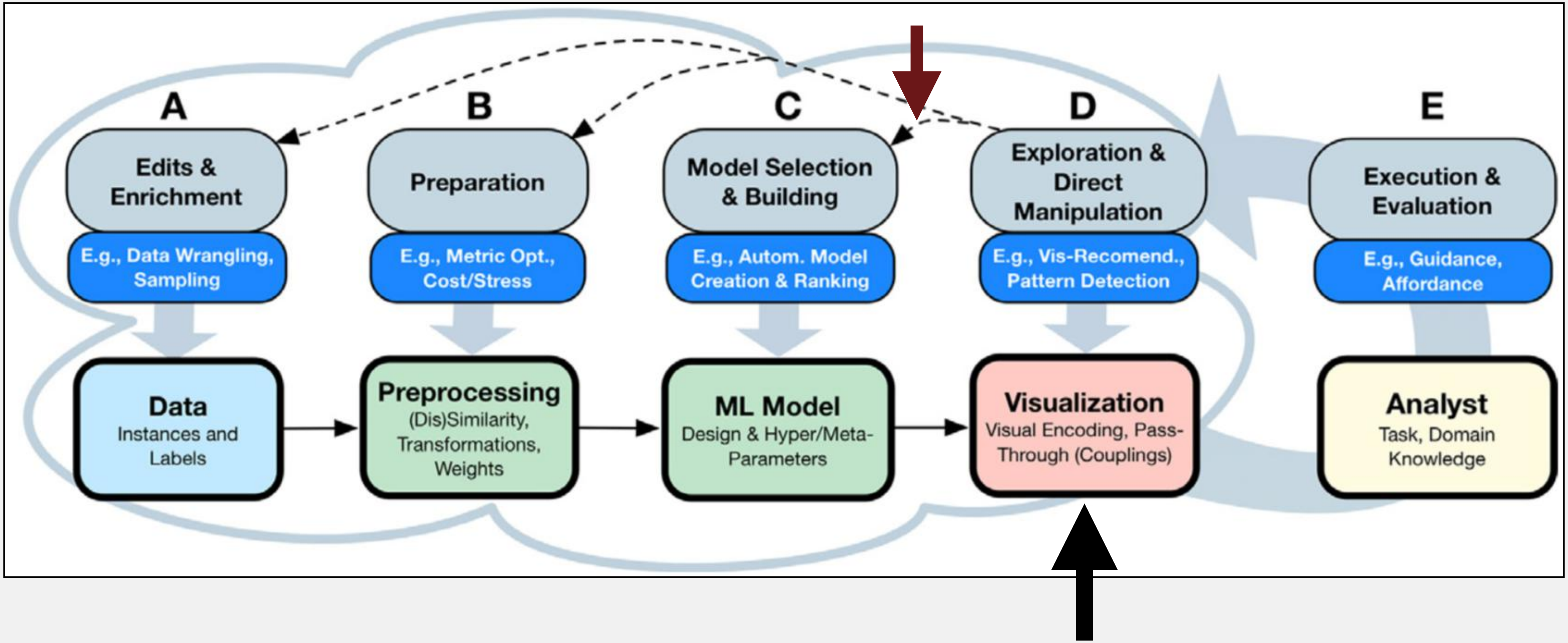
- Lots of work has considered this for **heteroscedastic noise** **[Hong et al. 2018]**
    - Automatic weighting schemes
- Should we simply use, and trust, an automatic scheme for weighting?
- User can also *specify weights*
- How does one know what to specify?

Edits & Enrichments



- Prepare our data for the model: dealing with non-quantitative data, removing data attributes, adding attributes, etc…
- (often less-studied in terms of visual exploration)

Visualization



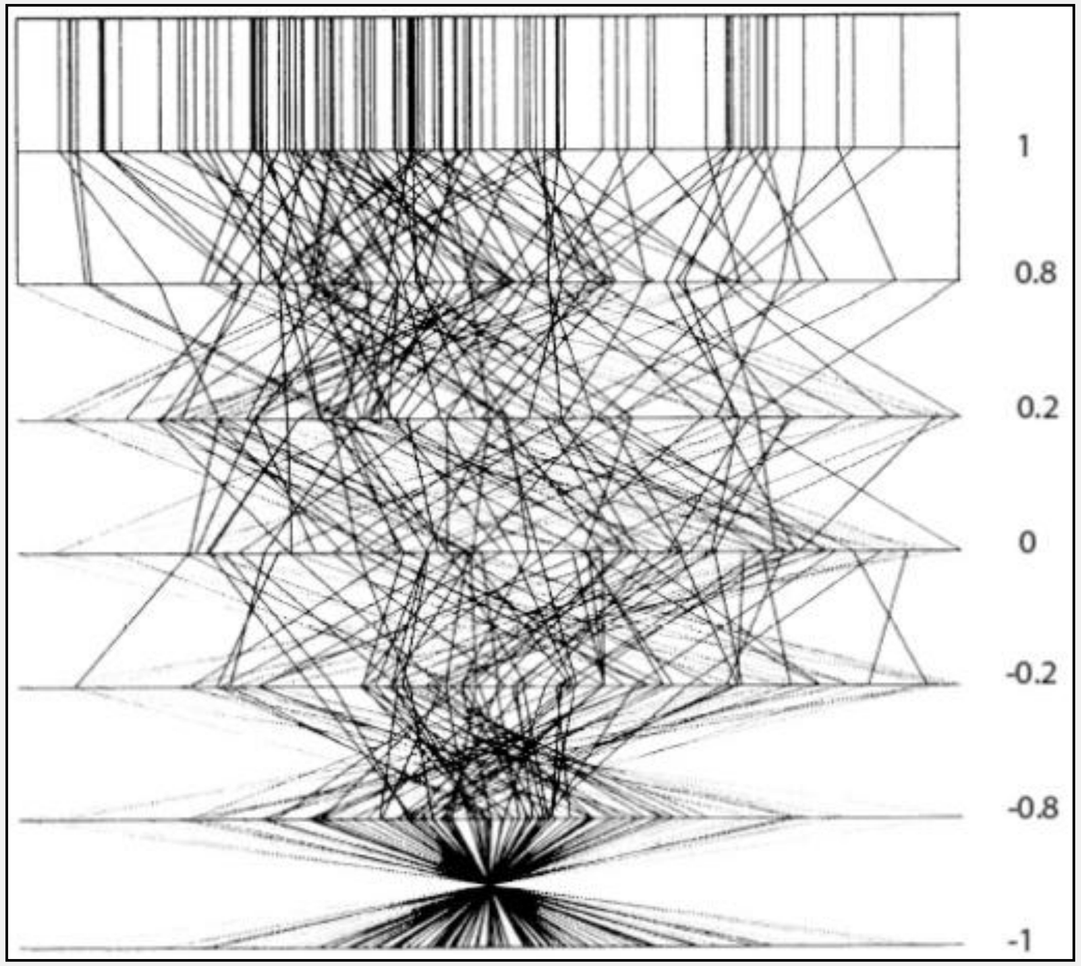
- How do we *visually encode* data + model?
- How does the user *interact* with each of the stages we just discussed?

Visualizing PCA

- What information is available to visualize?
  - Raw data
  - Covariance matrix
  - Low-dimensional projection
  - Eigenvalues
  - Eigenvectors
  - Projection quality
  - Reconstruction

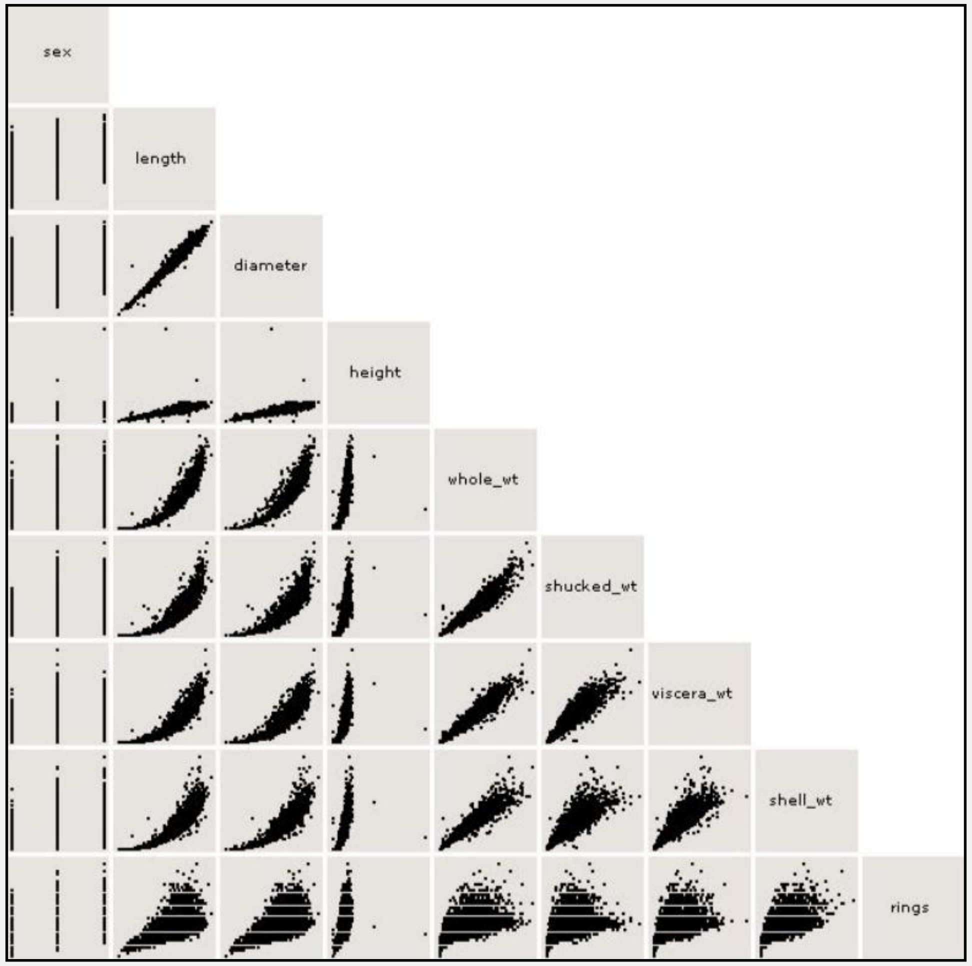
Raw Data Visualization

Parallel Coordinates



[Wegman 1990]

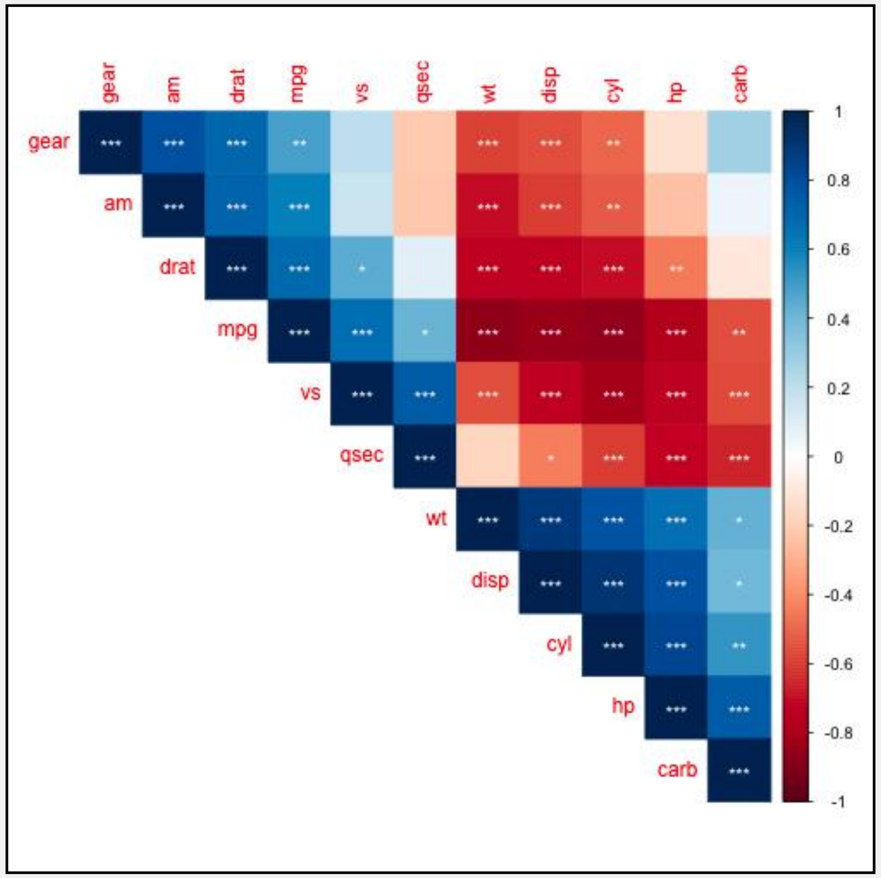
Scatterplot Matrix



[Wilkinson et al. 2005]

Covariance Matrix

- Heatmap: x-axis and y-axis domains are the same, specifically the data attributes



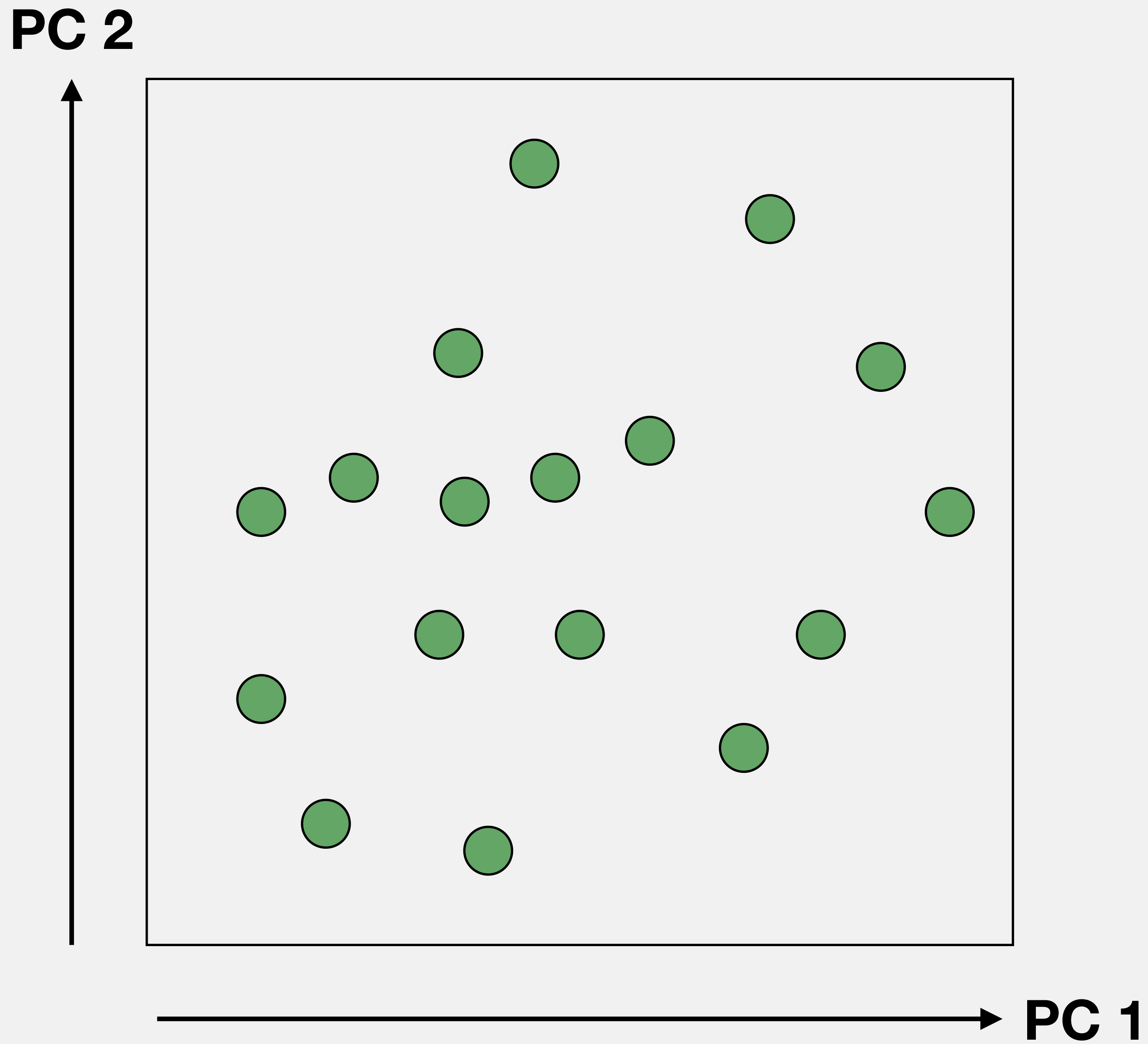
diverging colormap

https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html

Other visual channels to encode correlation?

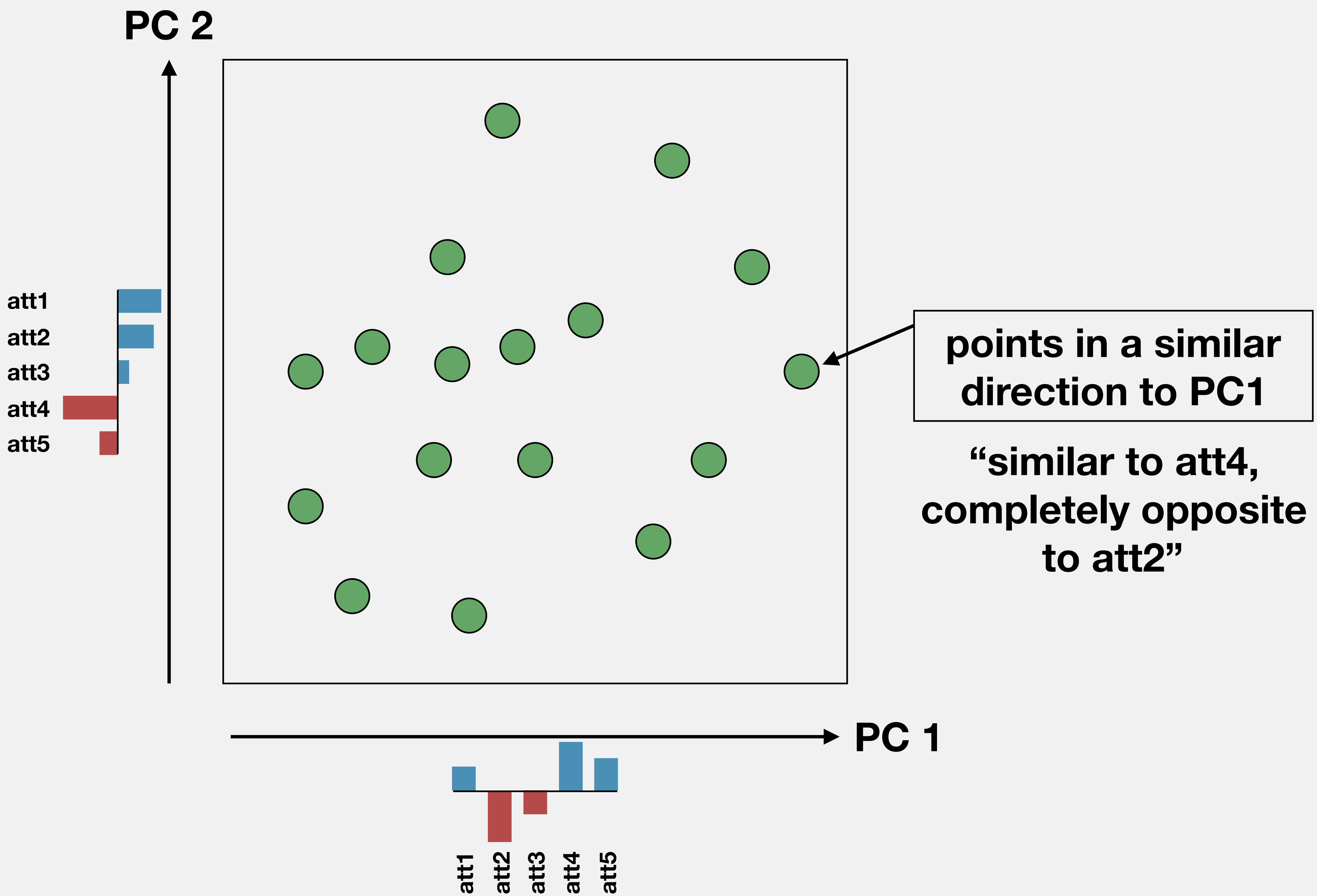
Low-Dimensional Projection

- Scatterplot: we associate each axis with a principal coordinate (the top two)



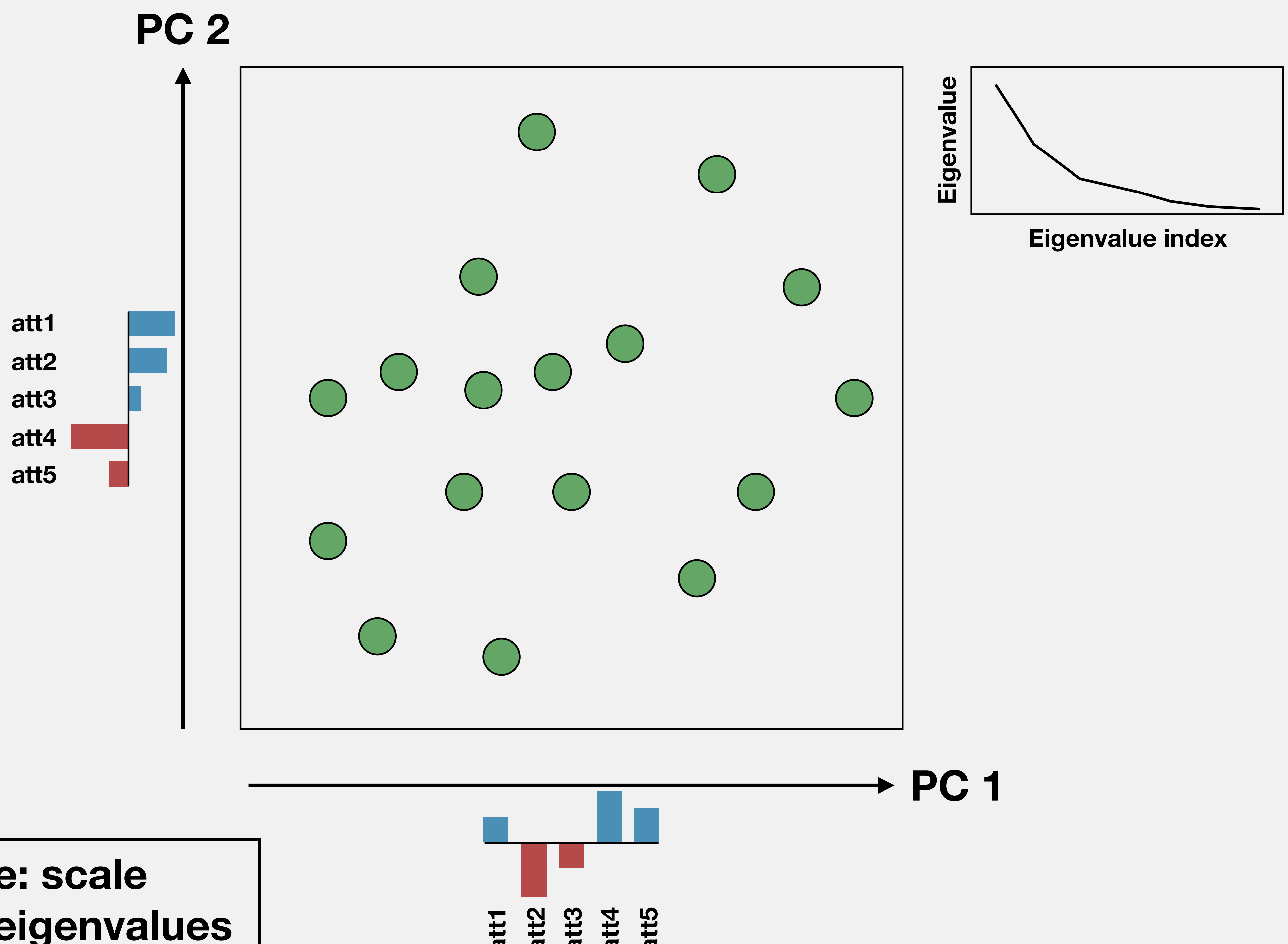
Eigenvectors

- A quantitative value per data attribute, per PC



Eigenvalues

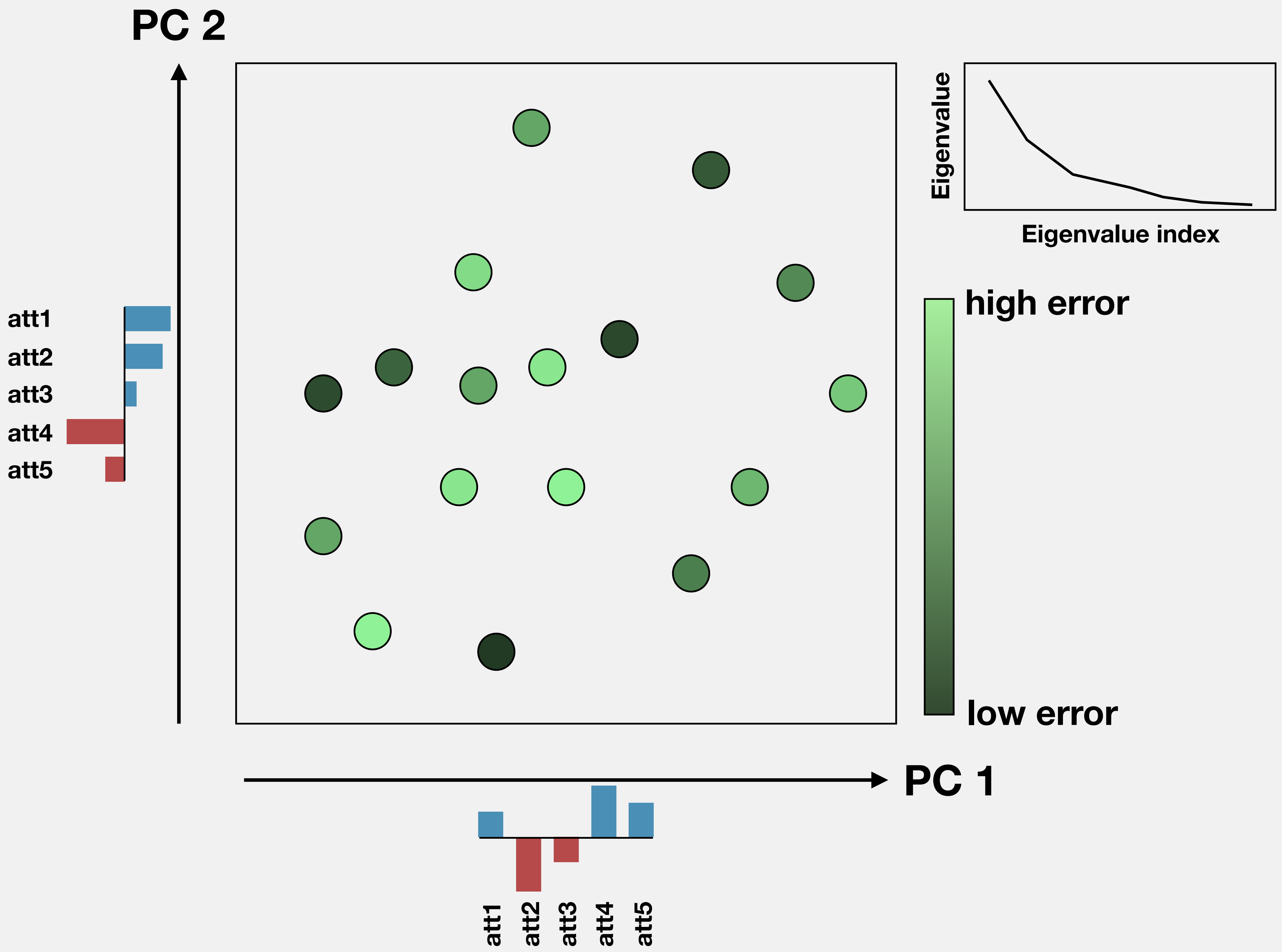
- One quantitative value per PC



Alternative: scale projection by eigenvalues

Projection Quality

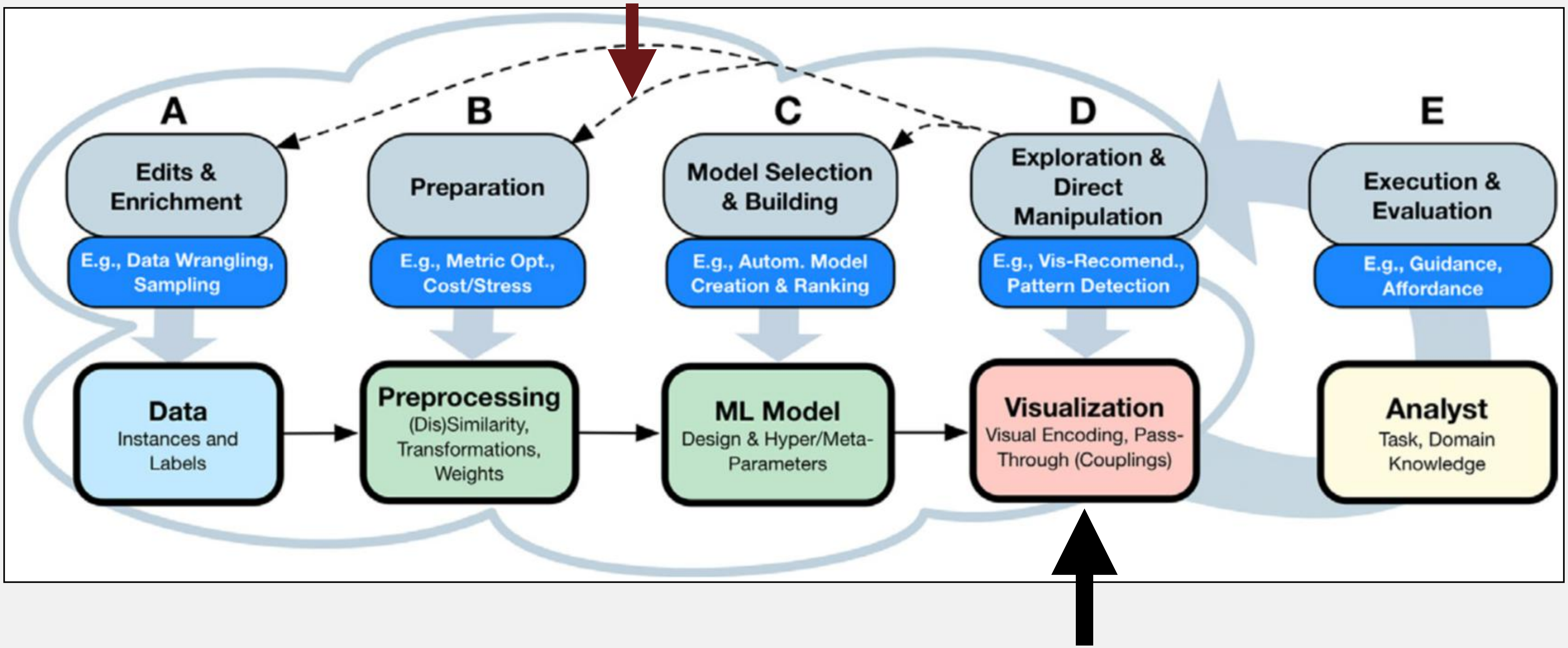
- A quantitative value per data item



Reconstruction

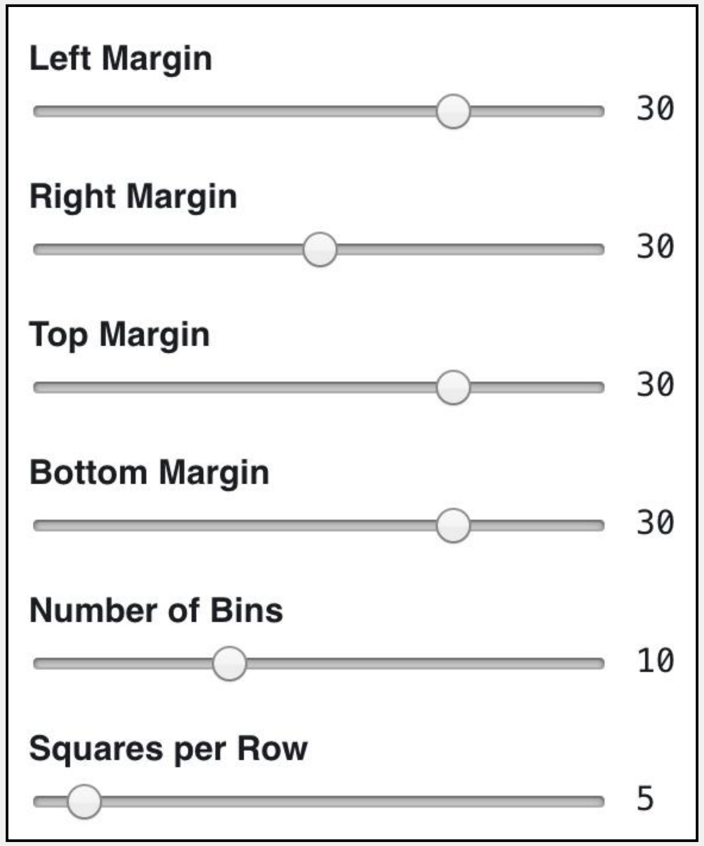
- Can go back to our raw data visualization plots: parallel coordinates and scatterplot matrices
- Want to place in context: original data, reconstructed data.
- Challenging enough to show high-dimensional data!
- (rarely done, projection quality usually shown in lieu of actual reconstructed points)

Visualization



Interaction

- Two basic types of interactions: **user interface elements**, **direct manipulation**.
- User Interface elements: sliders, buttons, menus, text areas, etc…



- Typically control global properties of the visualization: spacing, color, selecting data attributes, attribute values, etc..
- In response: a global change to the visualization…

Caution: Change Blindness

- A visible change occurs in your field of view, but you do not notice it.

- <u>Overwriting</u>: new visual representation comes in, the replacement of what you *did not attend to* goes unnoticed.

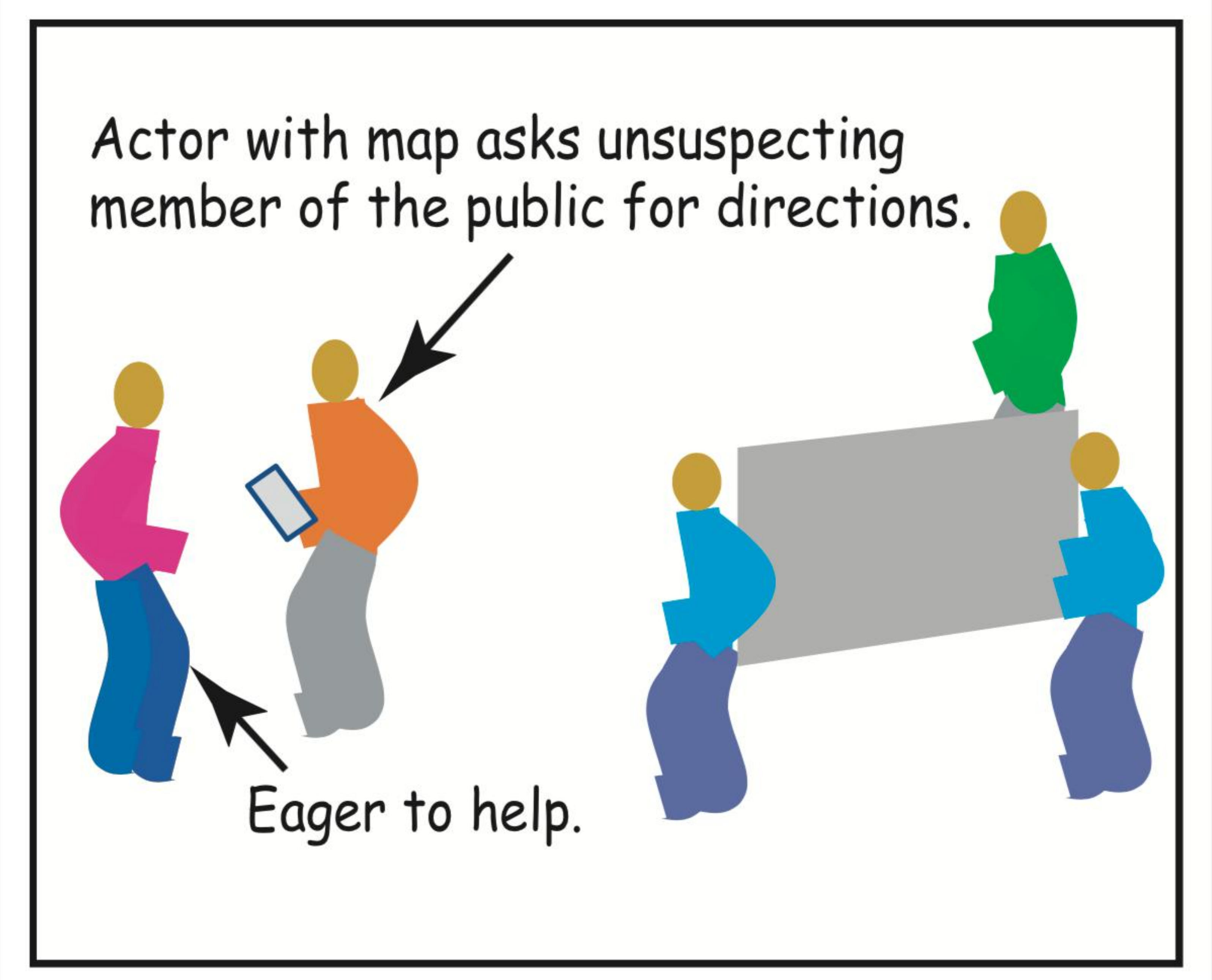
- <u>First impression</u>: you *do attend* to a visual element, but *fail to notice its change* - common if a visual change did not impact the meaning

- <u>Failure to compare</u>: no explicit comparison made between new/old

[Nowell et al. 2001]

What can we pay attention to?

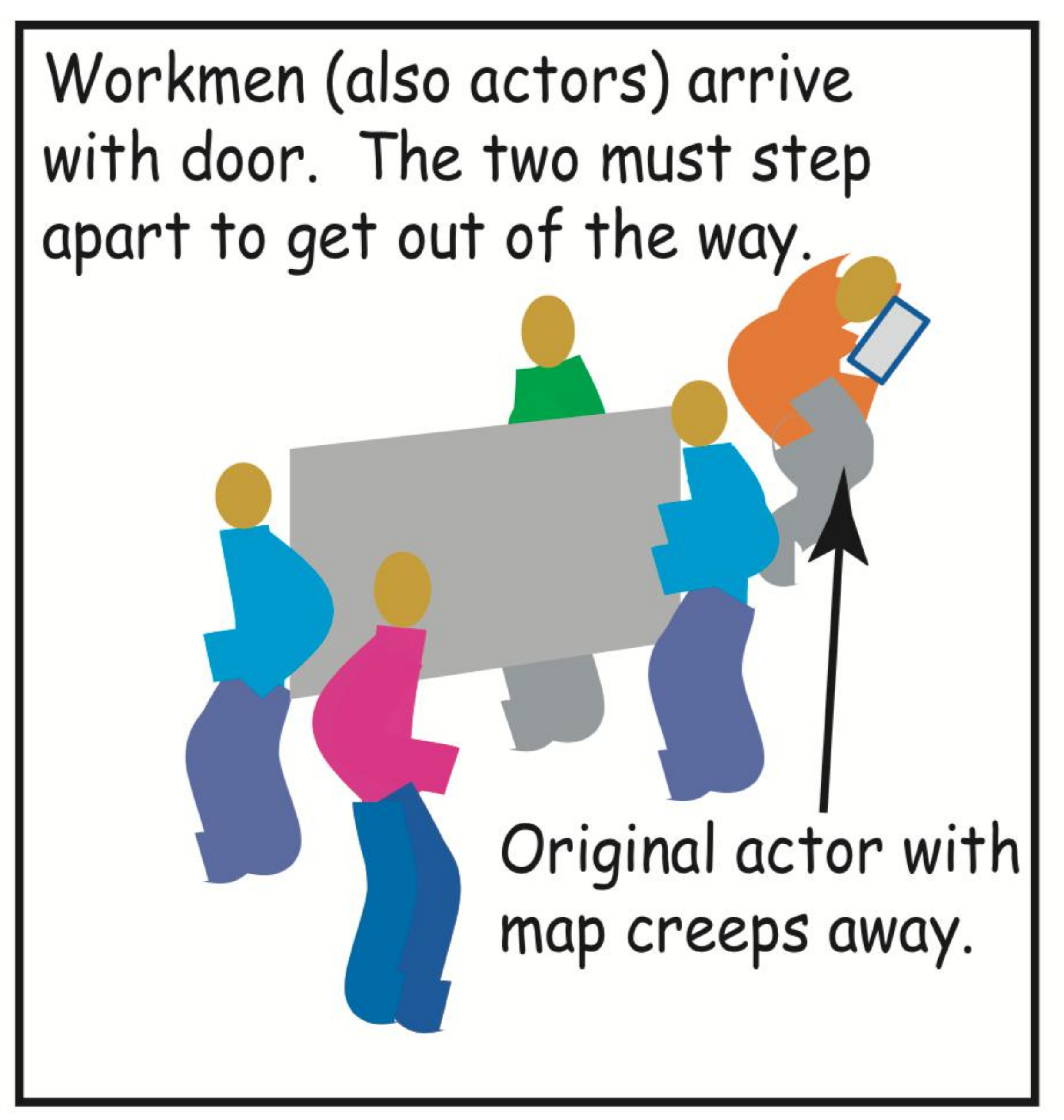
- **Visual working memory**: we can only hold about seven objects in short-term memory at one time (+/- 2).



[Ware]

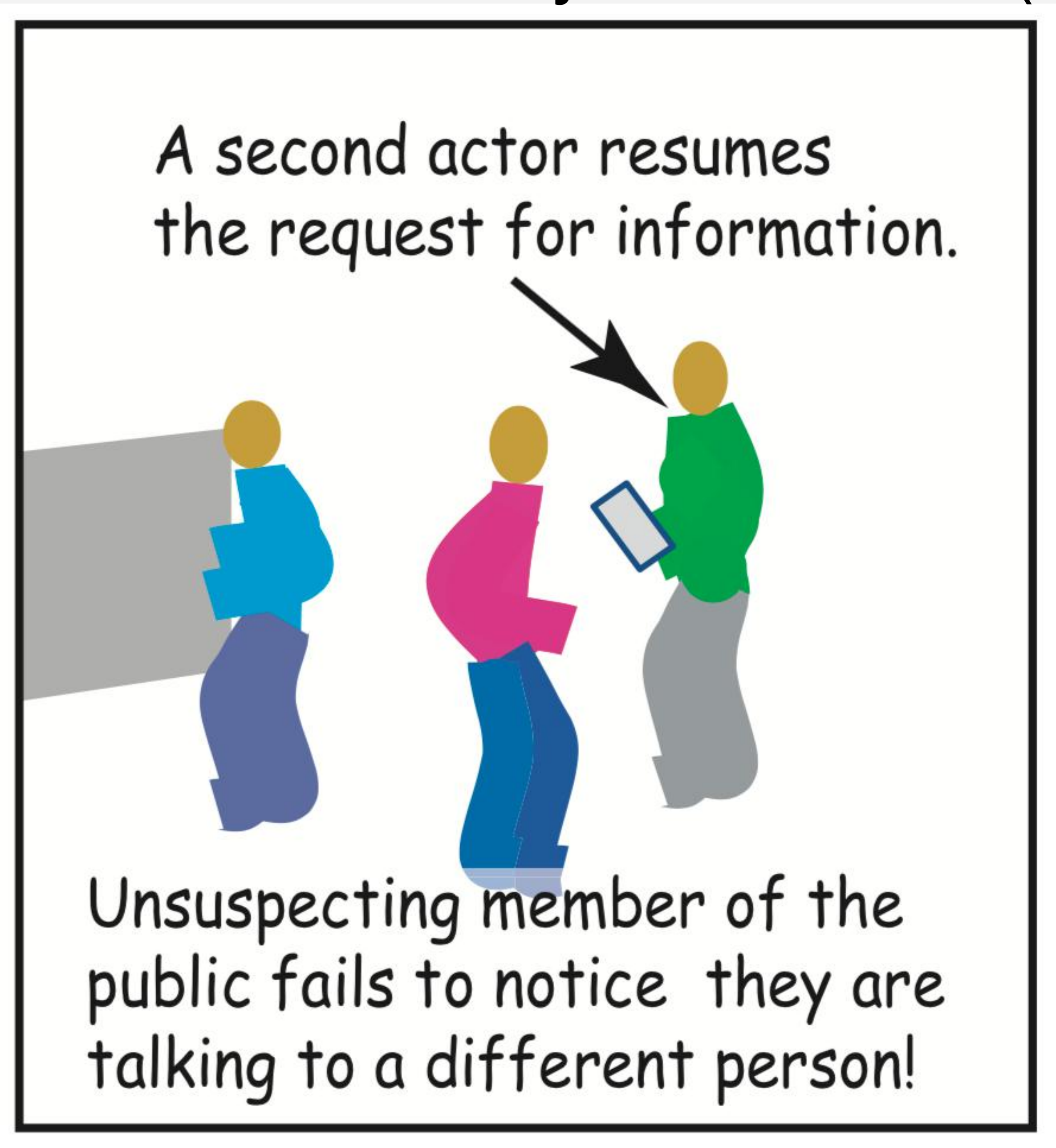
What can we pay attention to?

- **Visual working memory**: we can only hold about seven objects in short-term memory at one time (+/- 2).

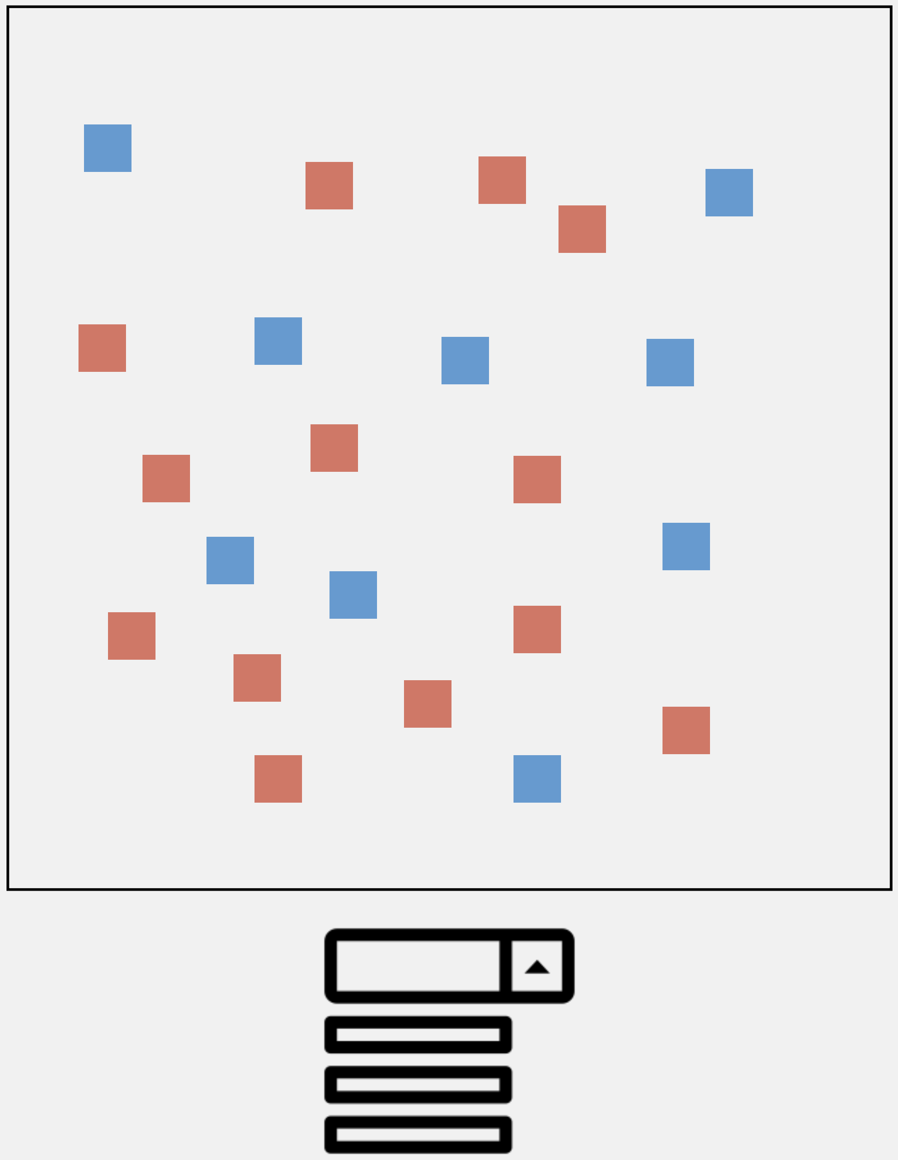


What can we pay attention to?

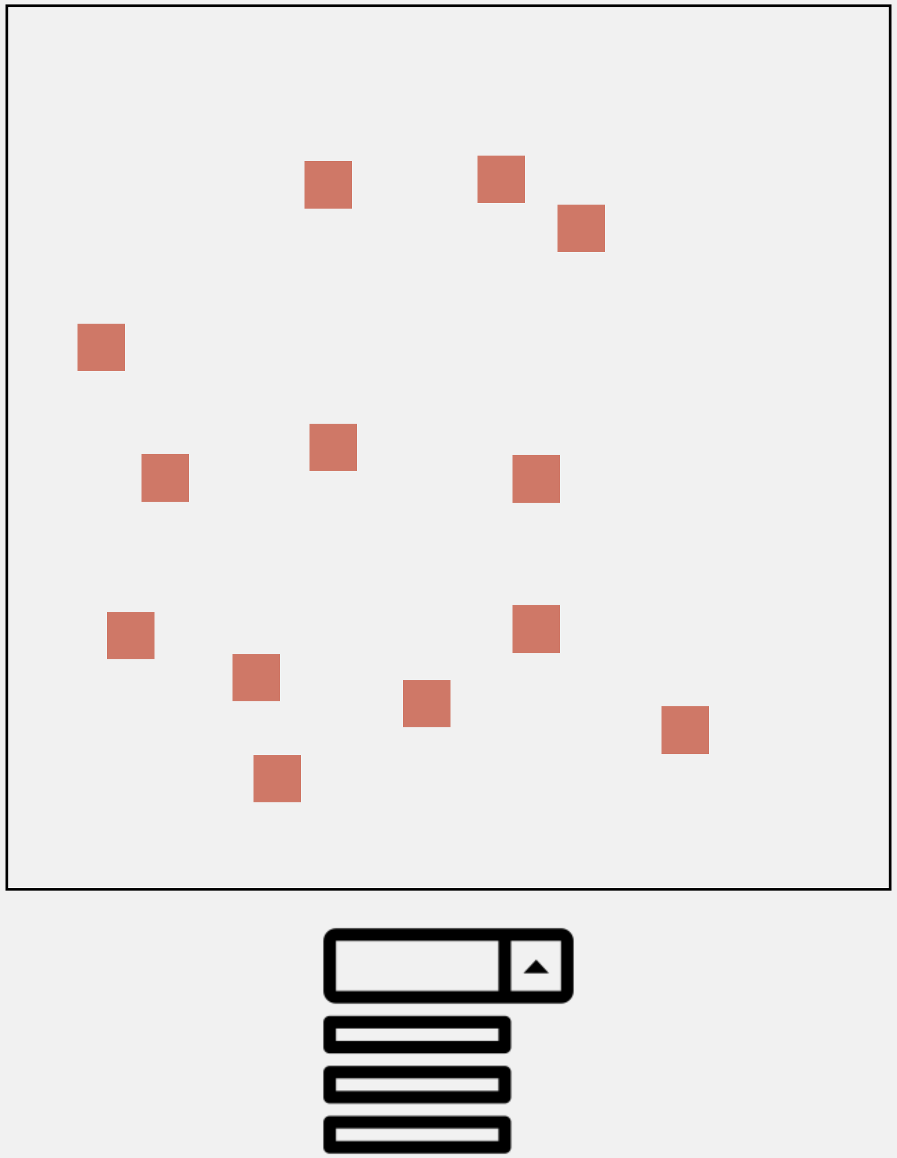
- **Visual working memory**: we can only hold about seven objects in short-term memory at one time (+/- 2).



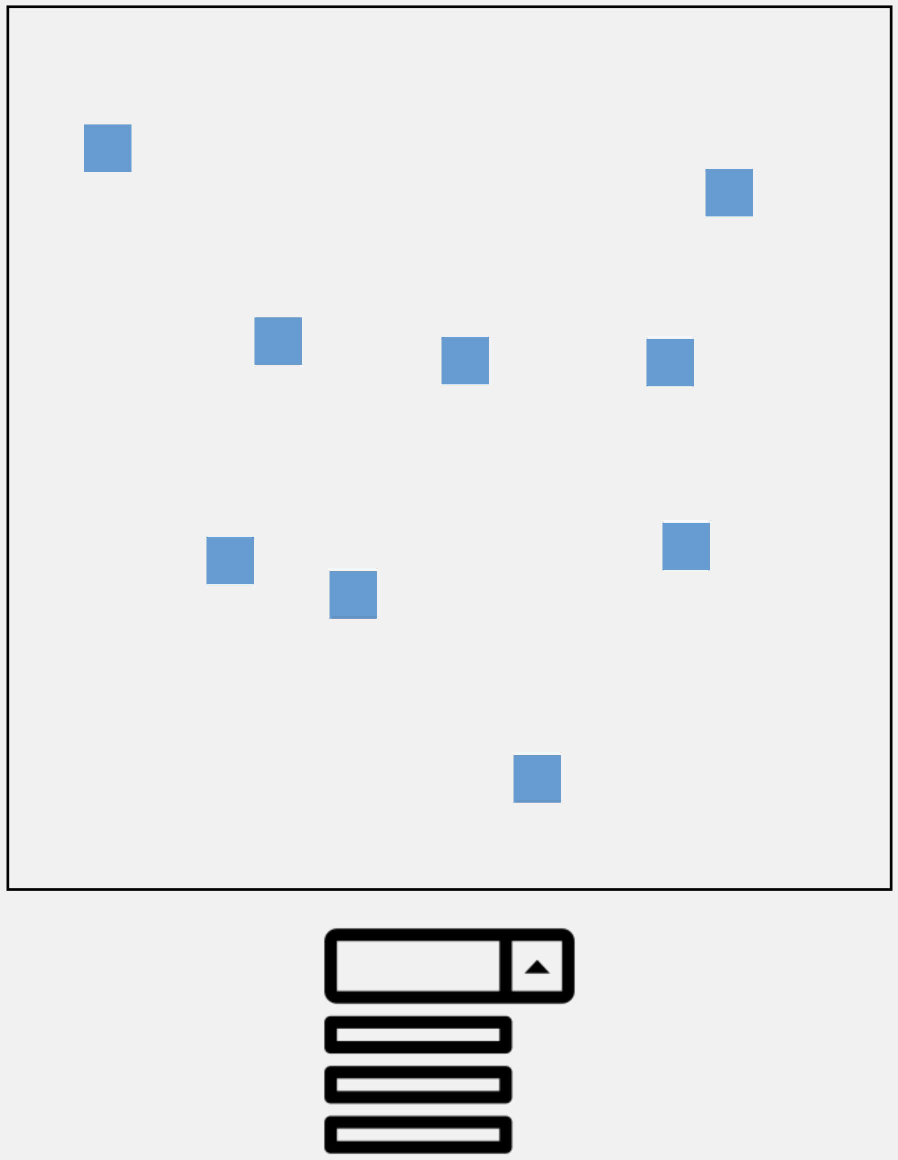
Implications



Implications

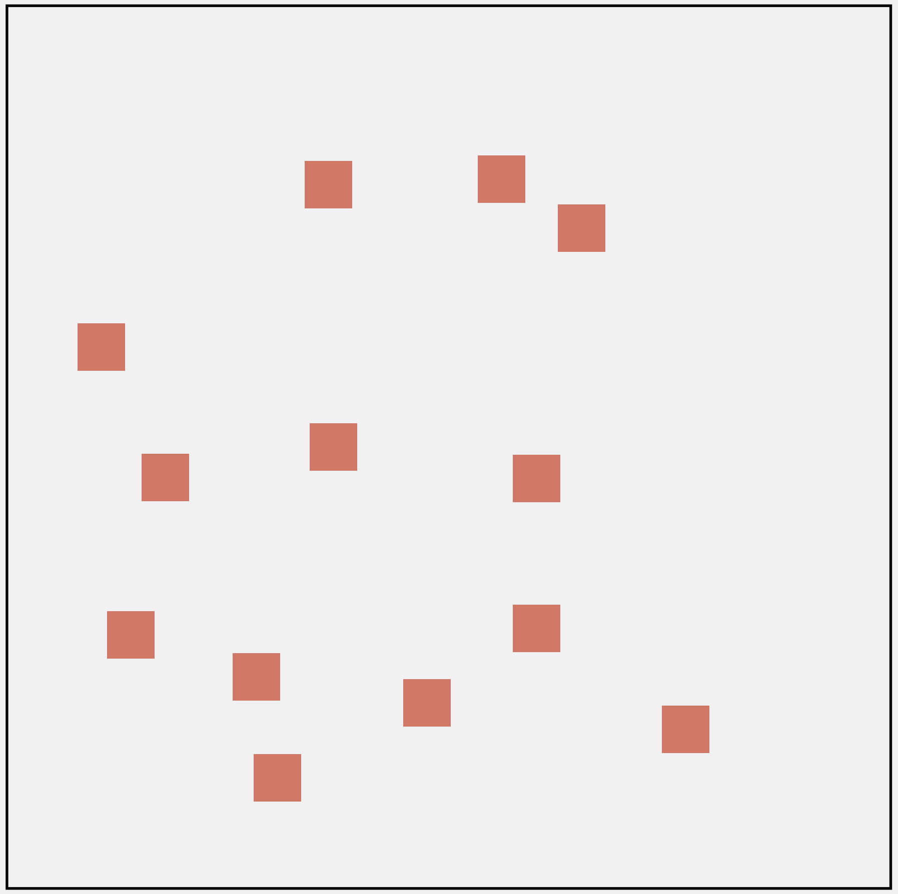


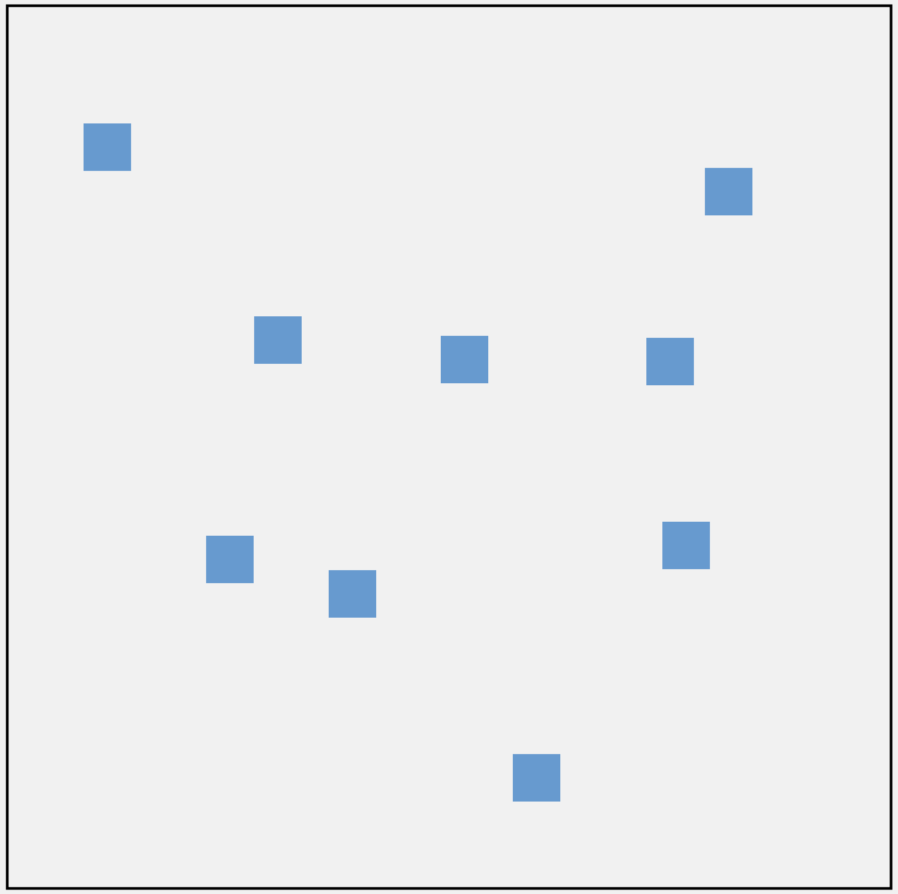
Implications



No Unjustified Interactions

- Consider multiple views



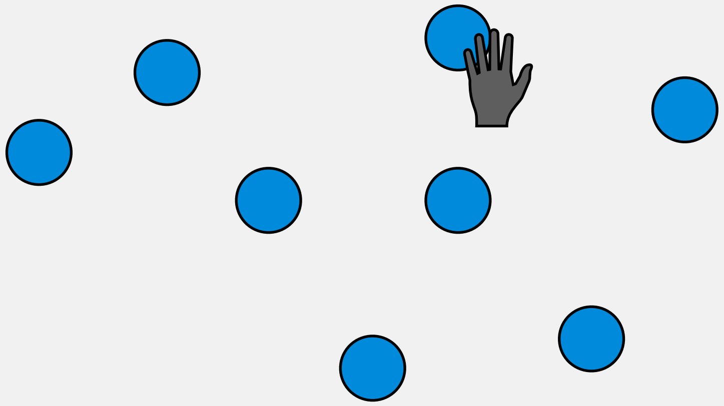


Mitigating Change Blindness

- Draw attention to elements / regions of change
- Visually distinguish old elements from new elements
  - Minimize amount of “stuff” the user needs to keep in memory
- On the other hand: need to prioritize with other design decisions…

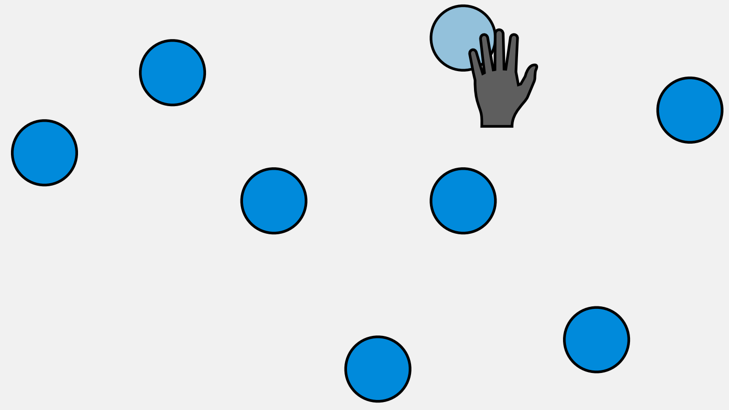
Direct Manipulation

- Interacting directly with graphical elements
- Advantages:
  - More fine-grained interactions: at the <u>data-item level</u>.
  - Another way to mitigate change blindness: what we interact with is what is (likely) to be changed.

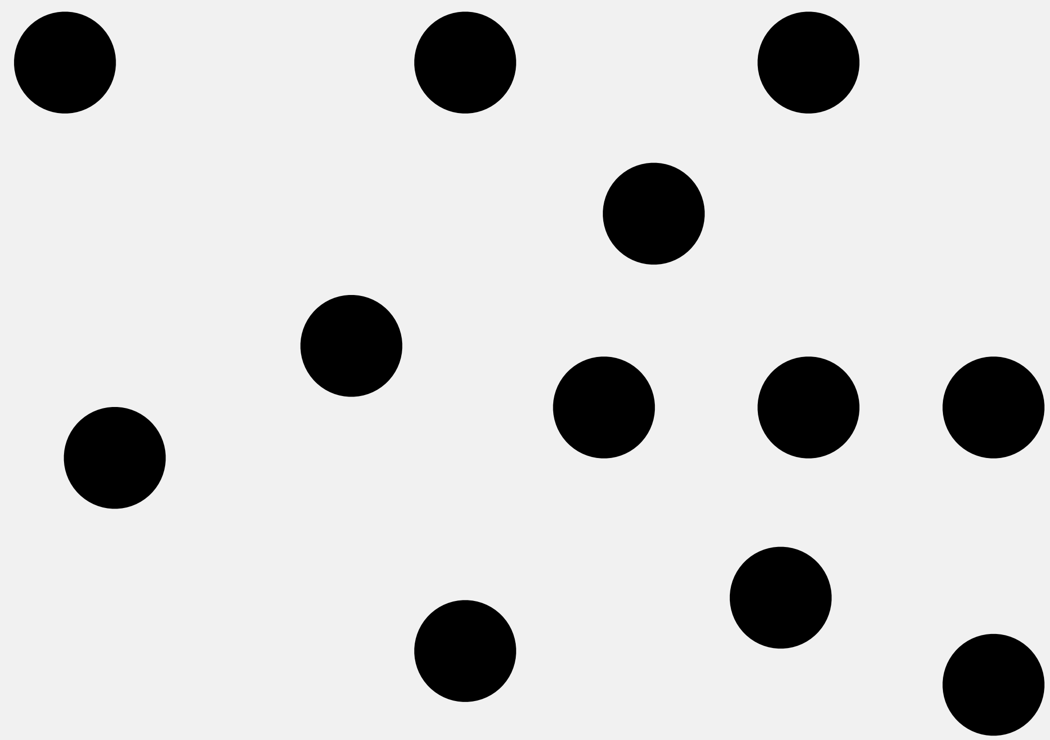


Direct Manipulation

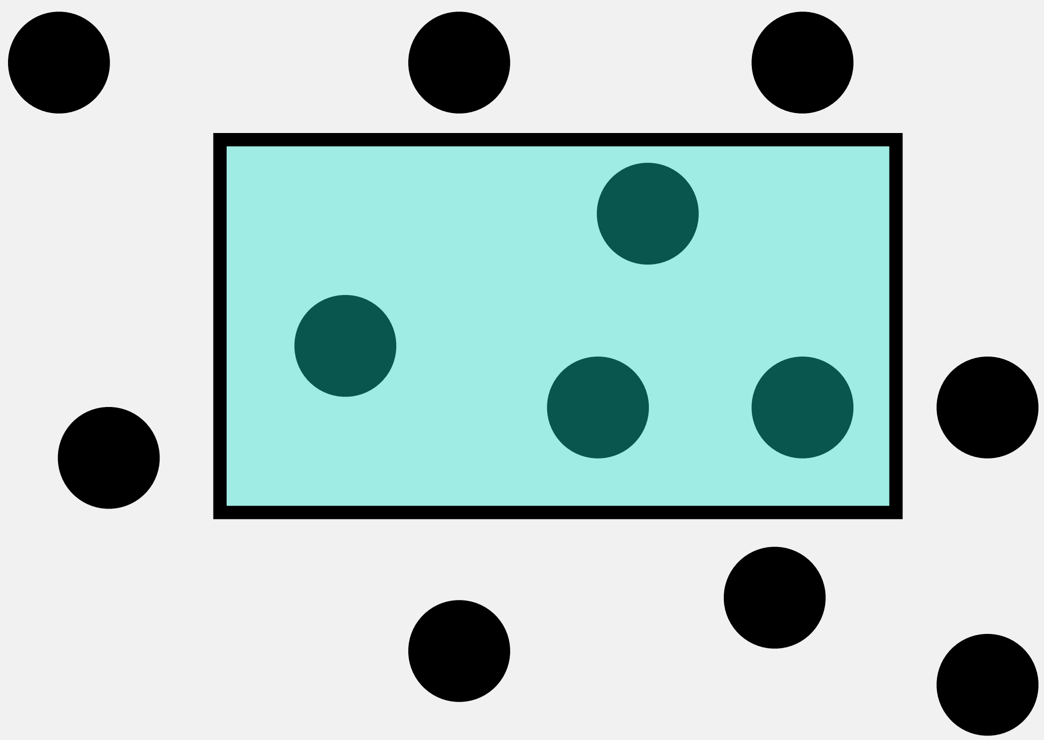
- Interacting directly with graphical elements
- Advantages:
    - More fine-grained interactions: at the data-item level.
    - Another way to mitigate change blindness: what we interact with is what is (likely) to be changed.



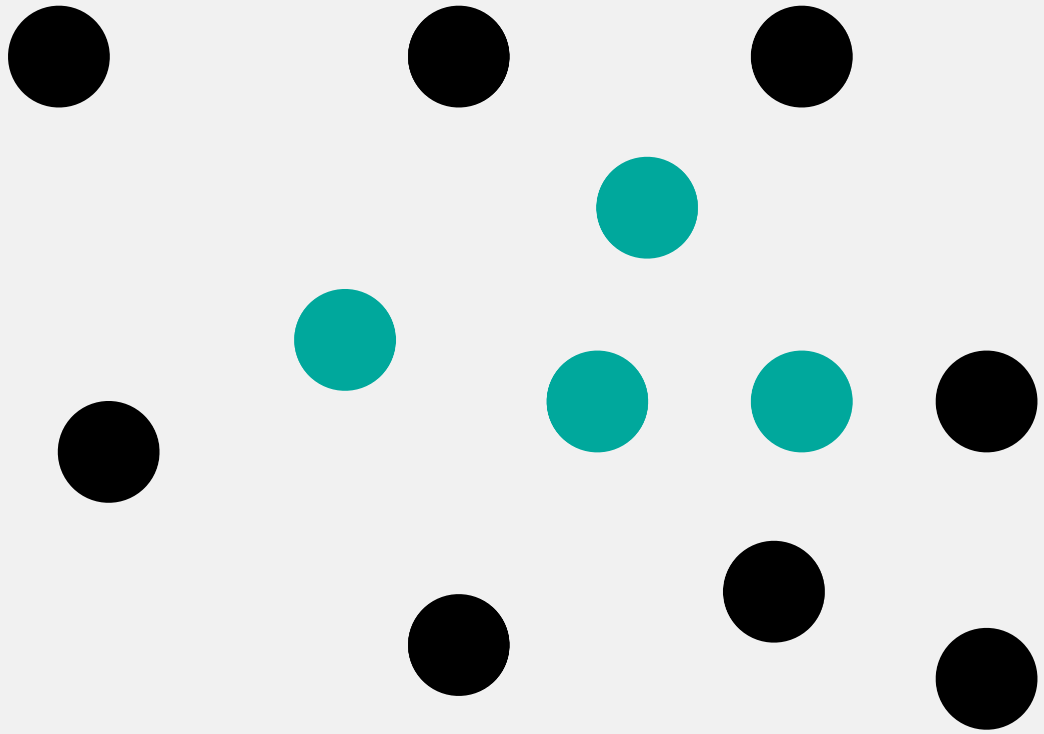
Brushing Data Items



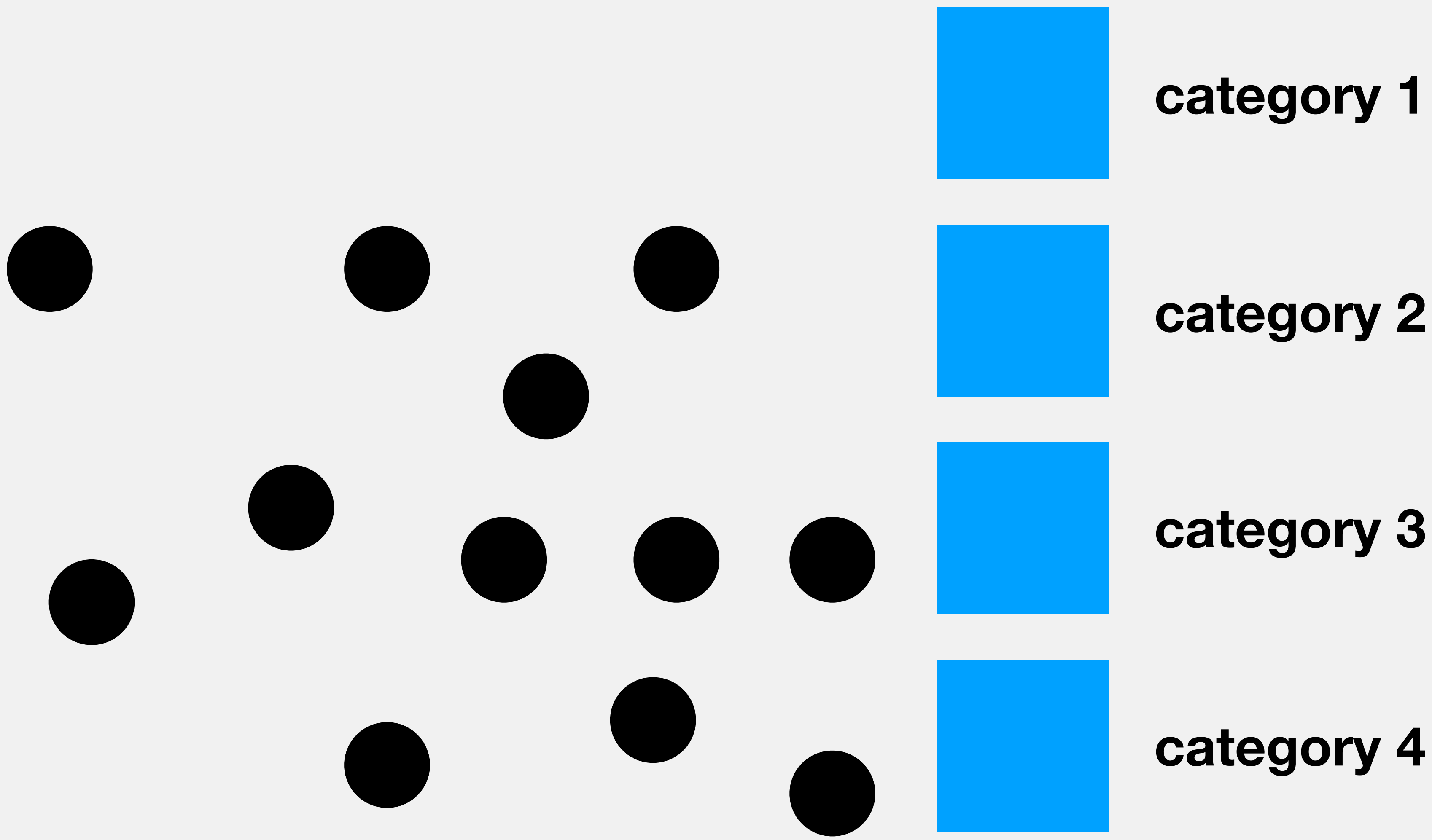
Brushing Data Items



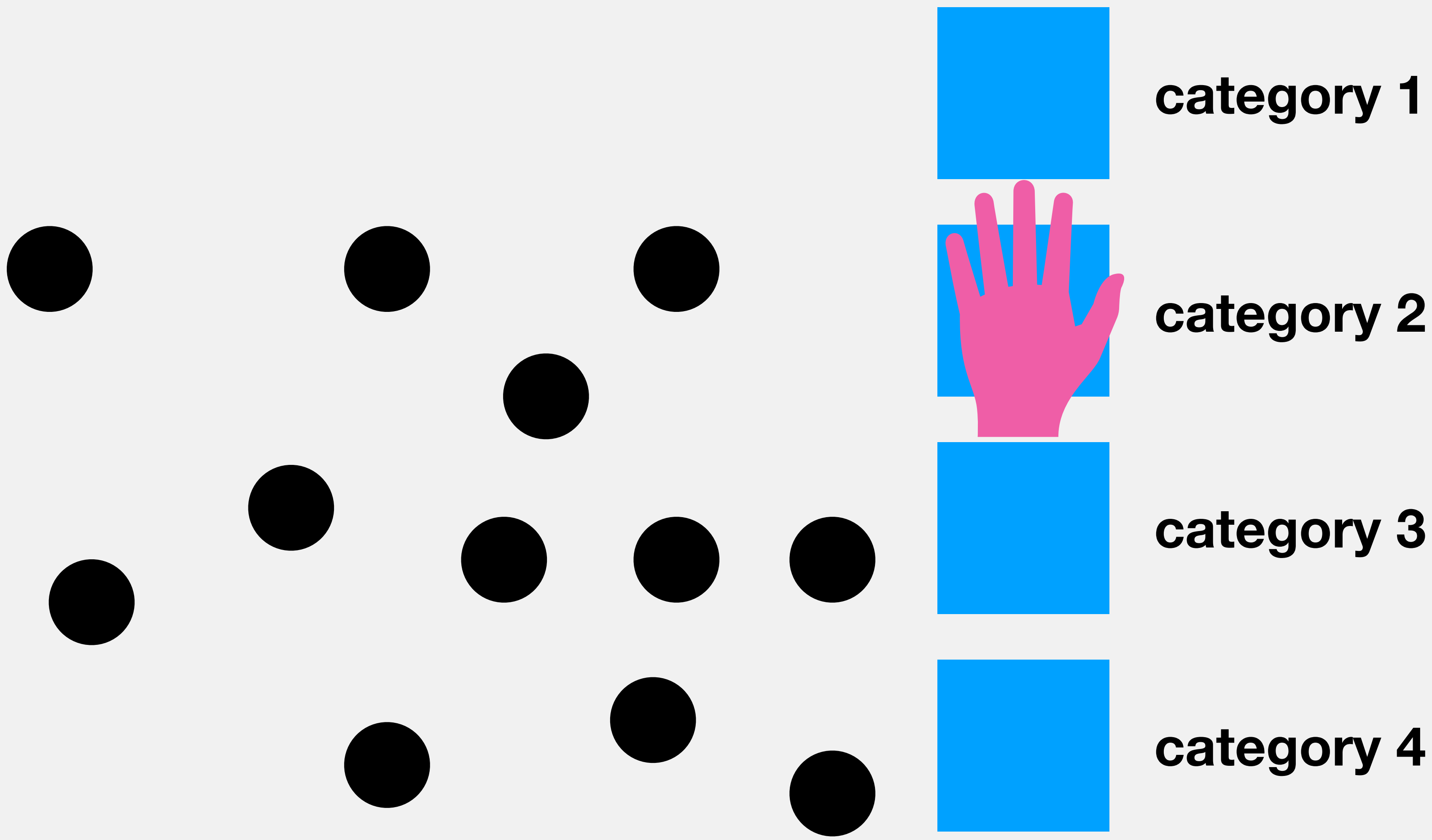
Brushing Data Items



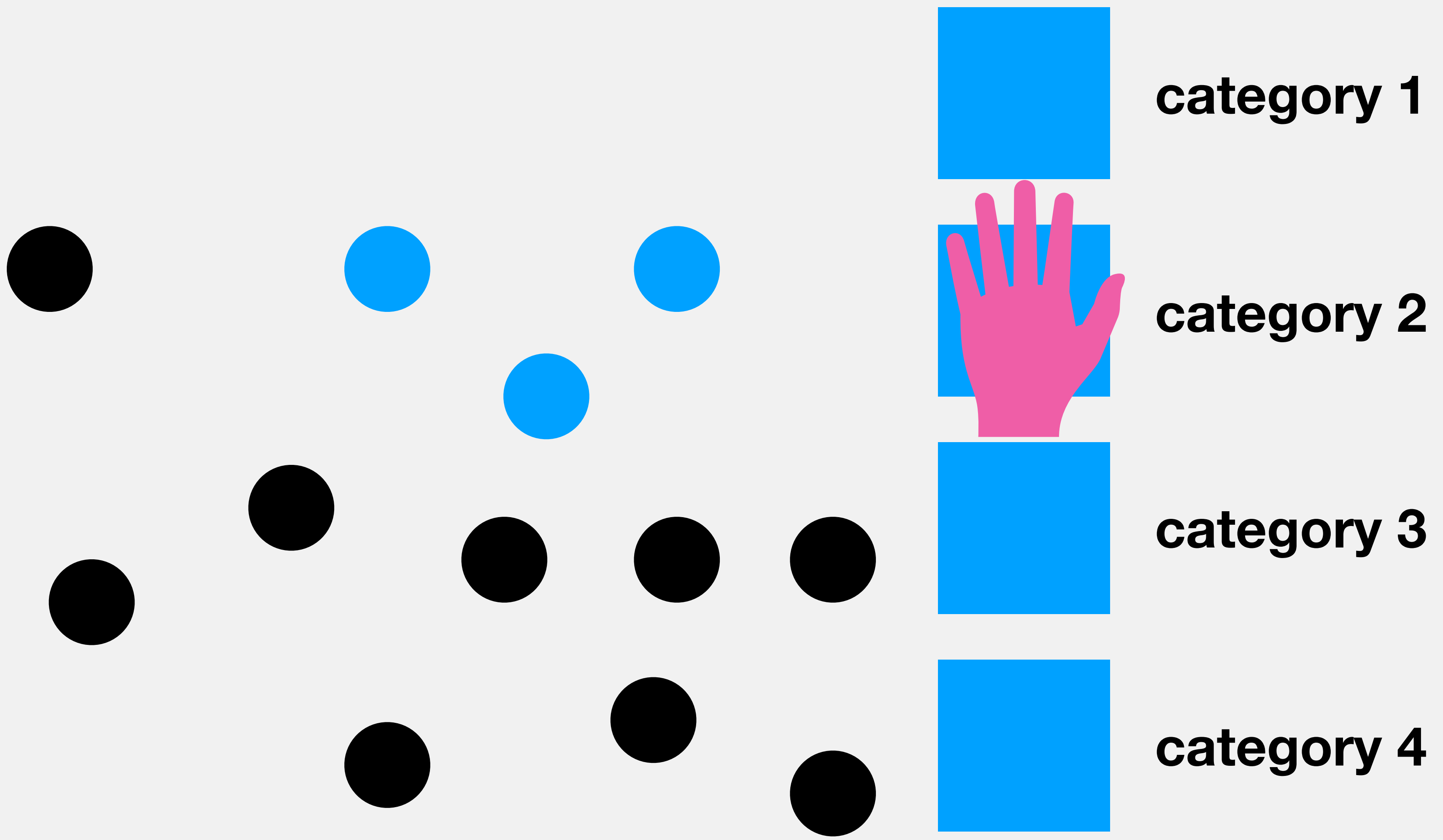
Data-Driven Selections



Data-Driven Selections



Data-Driven Selections



Back to PCA

- Interacting with graphical marks: one way to assign *weights to points*.
    - Everything inside of a brush: <u>important</u>; away from brush: <u>less important</u>
- Many visual analytics approaches, that we will cover, are focused on this very problem: <u>translating user interactions into model updates</u>.