LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Deep Learning

Chapter 9: Unsupervised Learning

Mina Rezaei

Department of Statistics – LMU Munich

Winter Semester 2020

UNSUPERVISED LEARNING

- So far, we have described the application of neural networks to supervised learning, in which we have labeled training examples.
- In these **supervised learning** scenarios, we exploit information of class memberships (or numeric values) to train our algorithm. That means in particular, that we have access to labeled data (y).
- In supervised learning model learn a function to map x to y.
- Examples are: classification, regression, object detection, semantic segmentation, image captioning,etc.

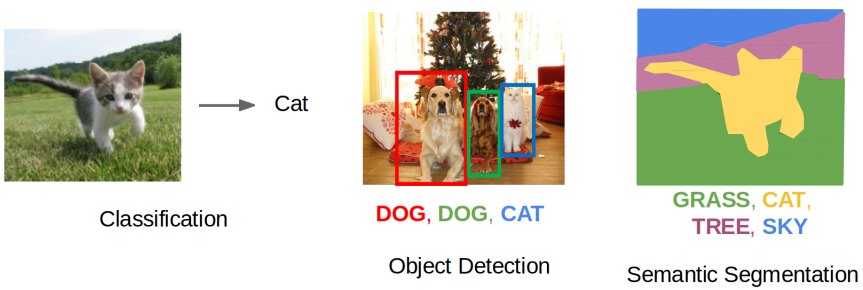


Figure: examples of supervised learning model where we have corresponding labels for training dataset

UNSUPERVISED LEARNING

- There exists another learning paradigm, **unsupervised learning**, where:
- Training data consists of unlabeled input points $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$
- our goal is to learn some underlying hidden structure of the data.
- Examples are: clustering, dimensionality reduction, feature learning, density estimation, etc

UNSUPERVISED LEARNING - EXAMPLES

1. Clustering.

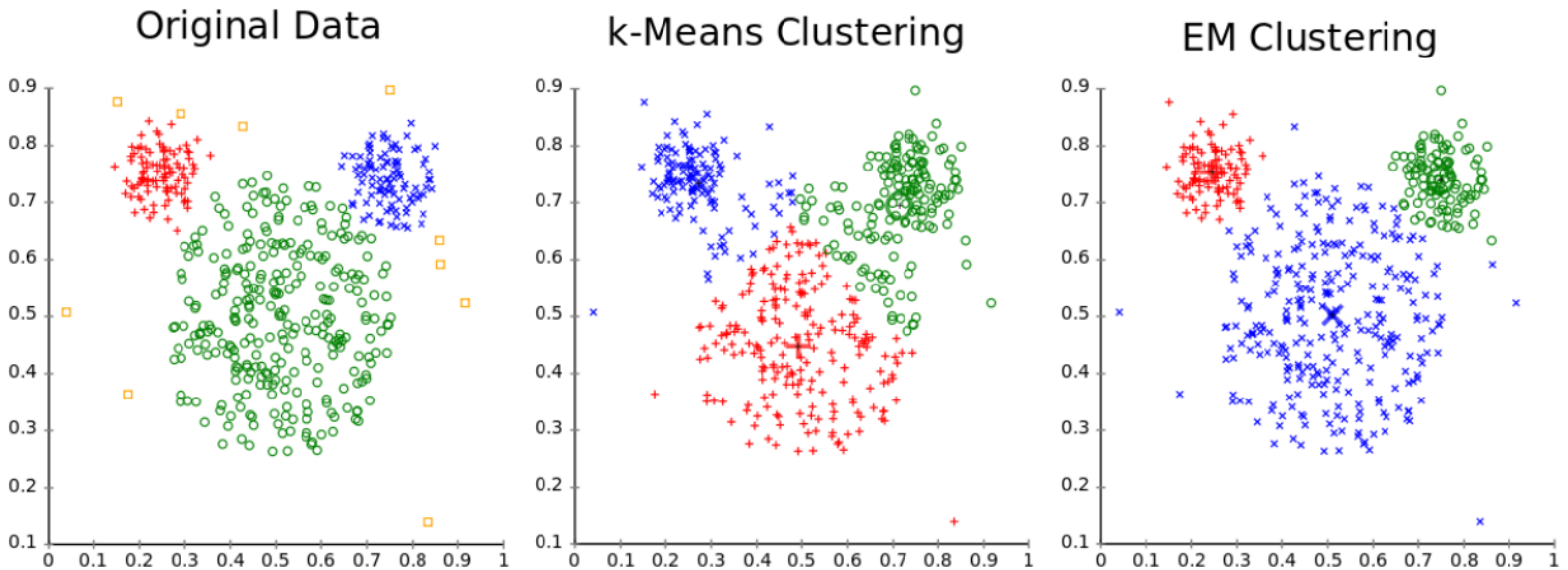


Figure: Different cluster analysis results on a dataset. True labels (the colors in the original data) are shown here but the algorithms only operate on unlabelled data. (Source : Wikipedia)

UNSUPERVISED LEARNING - EXAMPLES

2. Dimensionality reduction/manifold learning.

- E.g. for visualisation in a low dimensional space.

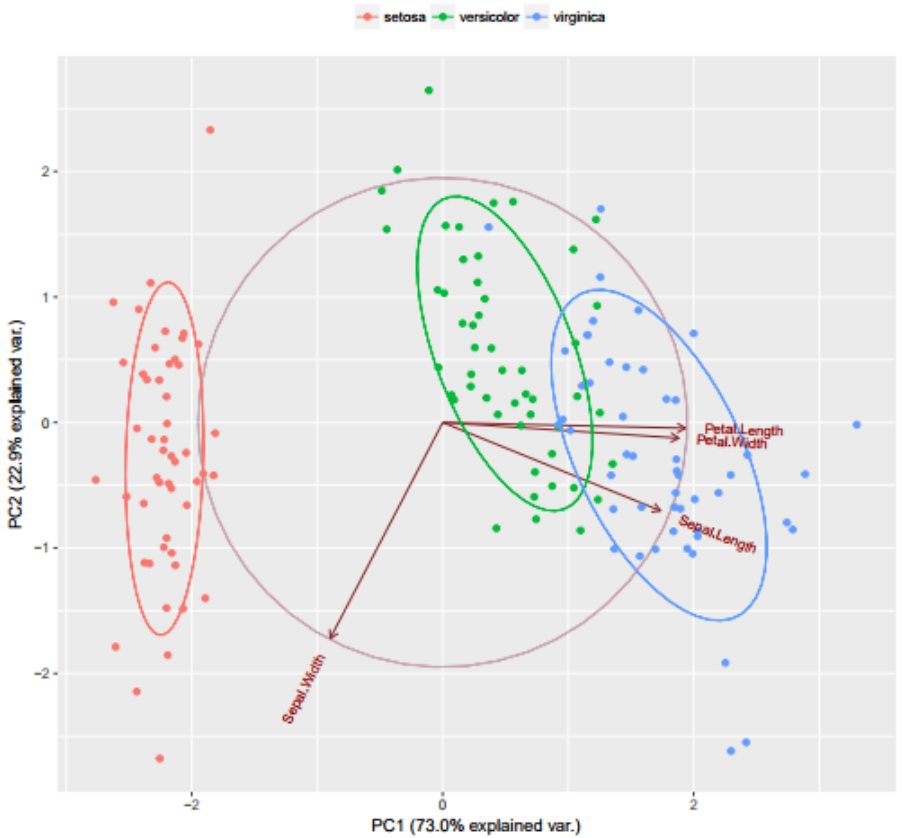


Figure: Principal Component Analysis (PCA)

UNSUPERVISED LEARNING - EXAMPLES

2. Dimensionality reduction/manifold learning.

- E.g. for image compression.

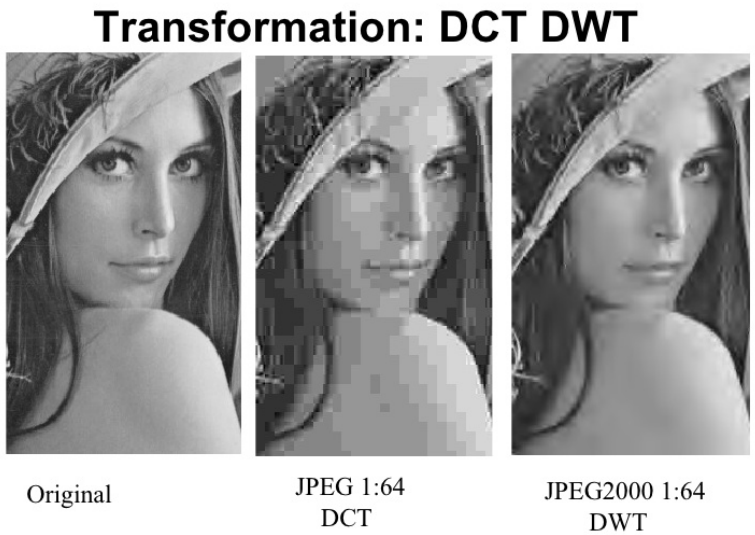


Figure: from https://de.slideshare.net/hcycon/bildkompression

UNSUPERVISED LEARNING - EXAMPLES

3. Feature extraction/representation learning.

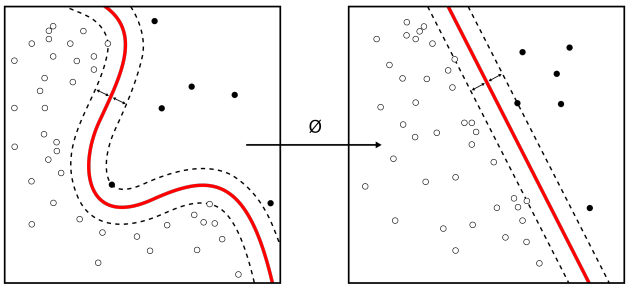


Figure: Source: Wikipedia

- E.g. for **semi-supervised learning**: features learned from an unlabeled dataset are employed to improve performance in a supervised setting.

UNSUPERVISED LEARNING - EXAMPLES

4. Density fitting/learning a generative model.

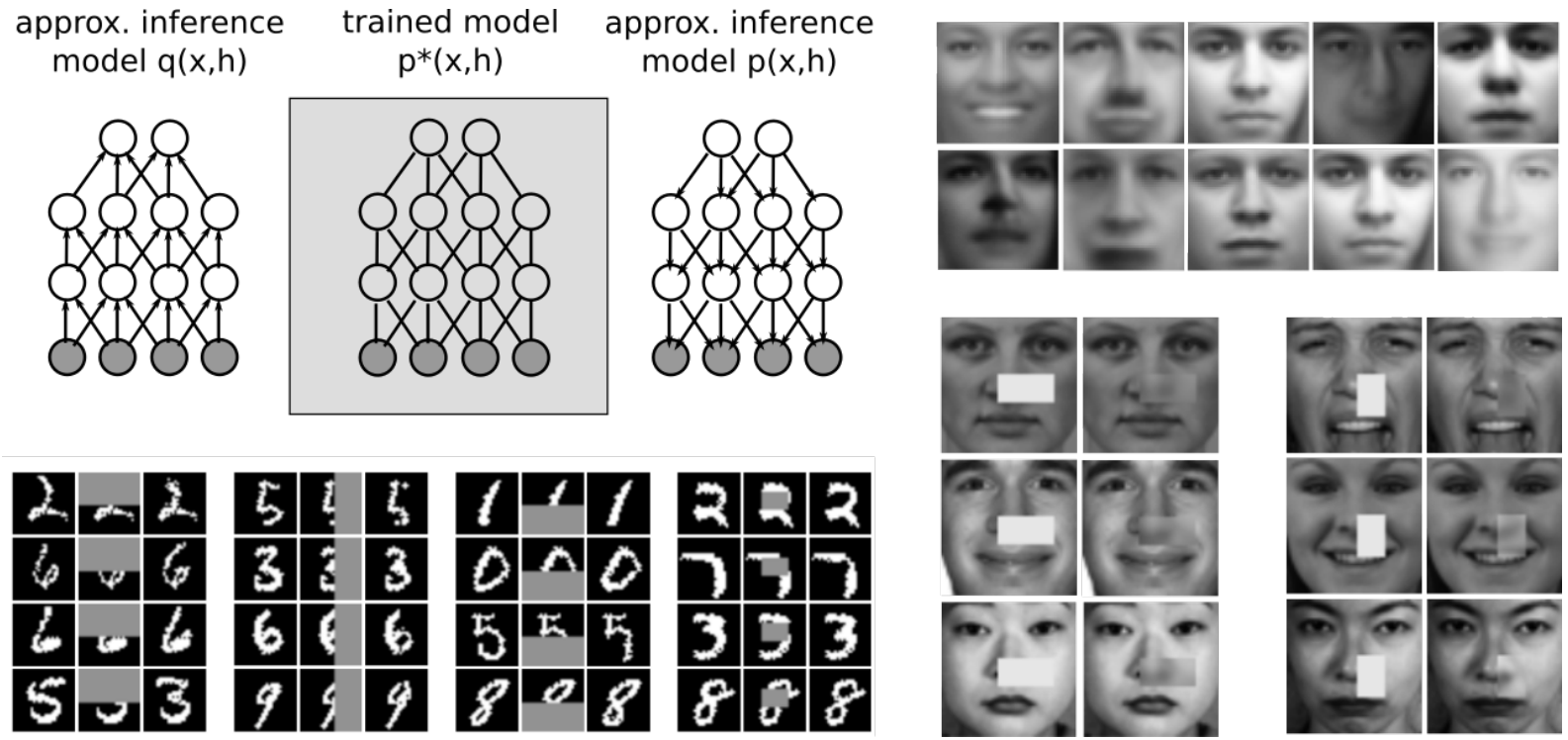


Figure: A generative model can reconstruct the missing portions of the images. (Bornschein, Shabanian, Fischer & Bengio, ICML, 2016)

MANIFOLD LEARNING

- **Manifold hypothesis**: Data of interest lies on an embedded non-linear manifold within the higher-dimensional space.
- A **manifold**:
  - is a topological space that locally resembles the Euclidean space.
  - in ML, more loosely refers to a connected set of points that can be approximated well by considering only a small number of dimensions.

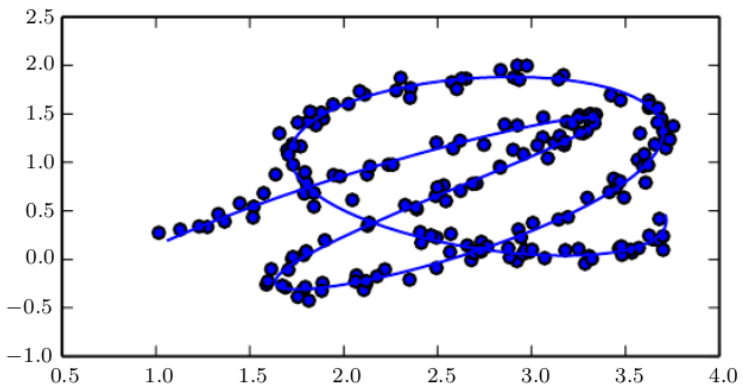


Figure: from Goodfellow et. al

MANIFOLD LEARNING

- An important characterization of a manifold is the set of its tangent planes.
- **Definition**: At a point $\boldsymbol{x}$ on a d-dimensional manifold, the **tangent plane** is given by d basis vectors that span the local directions of variation allowed on the manifold.

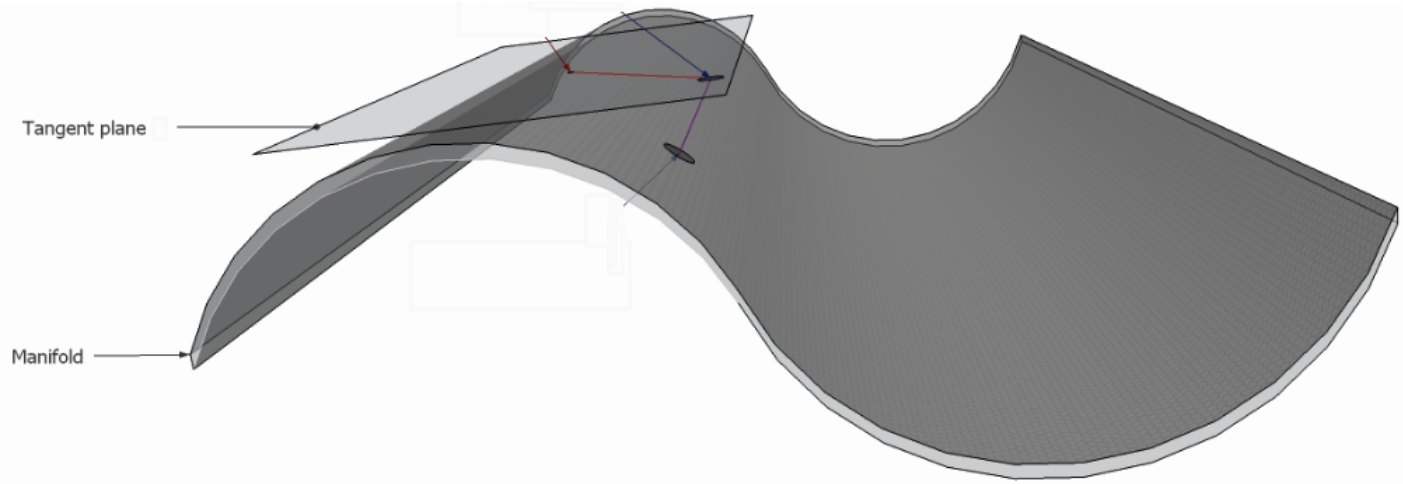


Figure: A pictorial representation of the tangent space of a single point, $\boldsymbol{x}$, on a manifold (Goodfellow et al. (2016)).

MANIFOLD LEARNING

- Manifold hypothesis does not need to hold true.
- In the context of AI tasks (e.g. processing images, sound, or text) it seems to be at least approximately correct, since :
  - probability distributions over images, text strings, and sounds that occur in real life are highly concentrated (randomly sampled pixel values do not look like images, randomly sampling letters is unlikely to result in a meaningful sentence).
  - samples are connected to each other by other samples, with each sample surrounded by other highly similar samples that can be reached by applying transformations (E.g. for images: Dim or brighten the lights, move or rotate objects, change the colors of objects, etc).

REVISION OF PCA

- The purpose of PCA is to project the data $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ onto a lower-dimensional subspace (e.g. to save memory).
- For each point $\mathbf{x}^{(i)} \in \mathbb{R}^p$ we need to find a corresponding code vector $\mathbf{c}^{(i)} \in \mathbb{R}^l$ with $l < p$. That step is accomplished by the encoding function which produces the code for an input:

$$f(\mathbf{x}) = \mathbf{c}$$

- Additionally, we need a decoding function to produce the reconstruction of the input given its code:

$$\mathbf{x} \approx g(f(\mathbf{x}))$$

- PCA is then defined by the choice of the encoding and decoding function.
- We can choose matrix multiplication to map the data back into $\mathbb{R}^p$: $g(\mathbf{c}) = \mathbf{Dc}$, with $\mathbf{D} \in \mathbb{R}^{p \times l}$, defining the decoding.

REVISION OF PCA

- To keep the encoding problem easy, PCA constrains the columns of $\mathbf{D}$ to be orthogonal.
- We begin with the optimal code $\mathbf{c}^*$ for each input. We could achieve this by minimizing the distance between the input $\mathbf{x}$ and its reconstruction $g(\mathbf{c})$ (PCA is a linear transformation with minimum reconstruction error)
- One way to obtain the optimal code $\mathbf{c}^*$ is to minimize the distance between the input $\mathbf{x}$ and its reconstruction $g(\mathbf{c})$ (that means, linear transformation with minimum reconstruction error):

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} ||\mathbf{x} - g(\mathbf{c})||_2^2$$

- Solving this optimization problem leads to

$$\mathbf{c} = \mathbf{D}^T\mathbf{x}$$

REVISION OF PCA

- Thus, to encode a vector, we apply the encoder function

$$f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$$

REVISION OF PCA

- We can also define the PCA as the reconstruction operation:

$$r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$$

- To find the encoding matrix $\mathbf{D}^*$, we minimize the Frobenius norm of the matrix of errors computed over all dimensions and points:

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \sqrt{\sum_{i,j} \left(x_j^{(i)} - r(x^{(i)})_j\right)^2}, \text{ subject to } \mathbf{D}^\mathbf{T}\mathbf{D} = \mathbf{I}_l$$

- for $l = 1$, $\mathbf{D}^*$ collapses to a single vector and we can rewrite the equation as

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} ||\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T||_F^2, \text{ subjected to } \mathbf{d}^T\mathbf{d} = 1$$

- The optimal $\mathbf{d}^*$ is given by the eigenvector of $\mathbf{X}^T\mathbf{X}$ corresponding to the largest eigenvalue.

REVISION OF PCA

- In general, for $l = k$ (with $k < p$) , the optimal reconstruction $\mathbf{X}^*$, by the Eckart-Young-Mirsky Theorem, is the truncated **Singular Value Decomposition (SVD)** of $\mathbf{X}$:

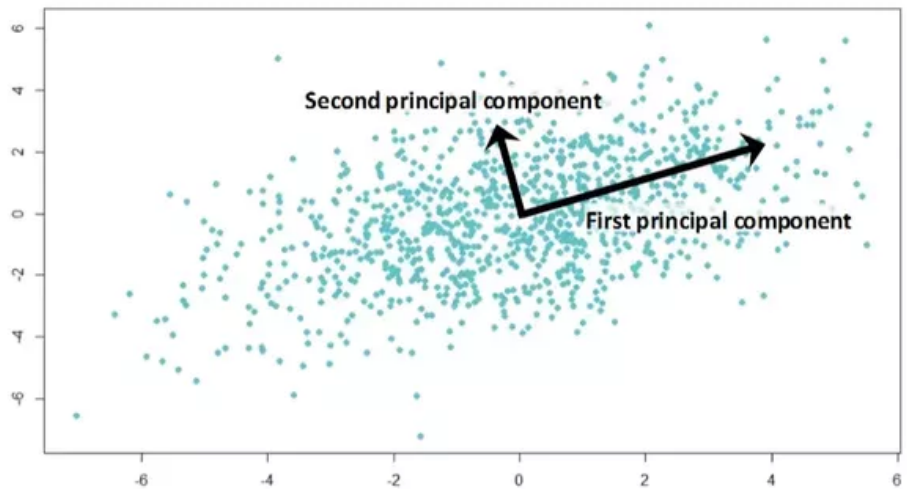
$$\mathbf{X}^* = \mathbf{U}_k \Sigma_k \mathbf{V}_k^\top$$

  where, the diagonal matrix Σ_k contains the k largest **singular values** and the columns of the matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ are the corresponding **right singular vectors** and **left singular vectors**, respectively.

- Here, the optimal encoding matrix $\mathbf{D}^*$ consists of the k left singular vectors as columns.

REVISION OF PCA

- The first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible).
- Each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.



credit:Syed Nazrul

Figure: The vectors shown are the (scaled) eigenvectors. Keeping only the first principal component results in dimensionality reduction.

UNSUPERVISED DEEP LEARNING

Given i.i.d. (unlabeled) data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \sim p_{\text{data}}$, in unsupervised deep learning, one usually trains :

- an autoencoder (a special kind of neural network) for **representation learning** (feature extraction, dimensionality reduction, manifold learning, ...), or,

UNSUPERVISED DEEP LEARNING

Given i.i.d. (unlabeled) data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \sim p_{\text{data}}$, in unsupervised deep learning, one usually trains :

- an autoencoder (a special kind of neural network) for **representation learning** (feature extraction, dimensionality reduction, manifold learning, ...), or,
- a **generative model**, i.e. a probabilistic model of the data generating distribution p_{data} (data generation, outlier detection, missing feature extraction, reconstruction, denoising or planning in reinforcement learning, ...).