Semi-supervised Learning

A MSIAM M2 Modelling Seminar Project

Supervisor:

Vasilii Feofanov

Authors:

Anastasia Petrova

Dmitrii Borisov

Margarita Kotova

Sofiia Rodina

Vladislav Shlenskii

February, 2020

Contents

List of Algorithms

List of Figures

List of Tables

1 Introduction

Semi-supervised learning is the branch of machine learning concerned with using labelled as well as unlabelled data to perform certain learning tasks. Conceptually situated between supervised and unsupervised learning, it permits harnessing the large amounts of unlabelled data available in many use cases in combination with typically smaller sets of labelled data.

This paper is focused on semi-supervised learning analysis for machine learning applications. We explore different semi-supervised algorithms, its principles and specifics. As the major study domain of this article semi-supervised learning assumptions were chosen: we discuss each assumption in details and conduct the experiments to examine the practical meaning of all of them. We also discuss additional factors that influence the performance of semi-supervised algorithms. Causal direction between features and labels in machine learning applications was also chosen for deep experimental analysis in this paper.

Semi-supervised learning as a machine learning paradigm is based on a set of underlying hypothesis which make the existence of semi-supervised algorithms possible.

1.1 Low density assumption

The key to semi-supervised learning problems is the prior assumption of *low density separation*, which means: the boundaries between classes should lie in low-density regions of the input space [1]. The low-density separation assumption arises automatically once we assume smoothness and the presence of classes. Semi-supervised learning algorithms tend to draw this boundary exactly in low-density regions. However, there are datasets, where decision boundary can be situated in high-density region. We can suggest, that semi-supervised learning does not work for such types of data.

1.2 Clustering assumption

The semi-supervised learning algorithms may use *cluster hypothesis*, which implies the data tend to form discrete regions. It is assumed that points in the same cluster are most likely have the common label, although there may be multiple clusters forming a single class (data that shares a label may spread across multiple clusters).

1.3 Manifold assumption

The *manifold assumption* is that data lie approximately on a manifold of much lower dimension than the input space. It usually assumes a Reimannian (i.e., locally Euclidean) structure, with data points "sufficiently" densely sampled, not some cluster or point cloud. The manifold assumption is practical when high-dimensional data are generated by some process that may be hard to model directly, but which has only a few degrees of freedom.

1.4 Causal and anticausal learning

In addition to major assumptions there are complementary factors that can influence the SSL models quality. Mey and Loog [2] explored a lot of significant elements in their article, e.g. data generation process, principal impossibility to outperform a supervised learner, causal direction and others.

Causality is a term used to discover the direction between two different components. We say that one component causes another component ($C \rightarrow E$) when the cause C influences the generation of effect E in some way. A simple causal model was presented by Schölkopf et al.[3] and is shown below. The effect E is caused by C with a deterministic mapping ϱ. Both E and C are influenced by noises N_E and N_C respectively(Figure 2).

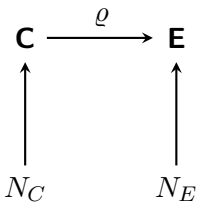


Figure 2: Simple causal model

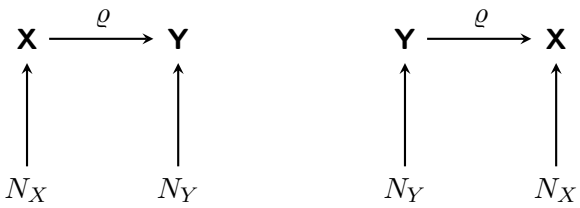


Figure 3: Causal models for machine learning applications

The approach can be projected to machine learning applications, where causal direction between features and labels is examined. Let X be a feature set and Y a set of labels. We consider two probabilities $P(Y|X)$ and $P(X)$. Assuming that X is the cause of Y these probabilities are independent. The situation changes when Y causes X, then the probability $P(X)$ contains information about the labels and such features are considered more helpful for SSL applications. Causal models are shown on Figure 3.

2 Semi-supervised algorithms

In this section we concisely describe several papers on topic of semi-supervised learning applicable to a classification task. To do so, we consider the notations as in *Table 1.*

Designation	**Explanation**
$\mathcal{X} \subset \mathbb{R}^n$	Feature space
$\mathcal{L} \subset \mathbb{Z}$	Set of labels
$\mathbf{x} \in \mathcal{X}$	Feature vector of a sample
$y \in \mathcal{L}$	Class a sample belongs to
$X_L \subset \mathcal{X} \times \mathcal{L}$	Set of labeled data
$X_U \subset \mathcal{X}$	Set of unlabeled data
$X_\Sigma = X_L \cup X_U$	Total set of data
$h \colon \mathcal{X} \to \mathcal{L}$	Classifier
$\mathcal{H}$	Hypothesis space
$\mathbb{J}(\bullet)$	Indicator function
$p(\bullet)$	Probability of a random variable
$\mathbb{E}(\bullet)$	Expectation of random variable
$\#A$	Cardinality of set A
$m(\bullet)$	Classification margin function
$\mathbb{R}^{n \times m}$	Space of matrices of size $n \times m$
$\mathbb{R}_+^{n \times m}$	Space of matrices of size $n \times n$ with nonnegative entries

Table 1: Mathematical notations

Some of the overviewed papers deal with binary classification problem, in this case we consider set of labels to be $\mathcal{L} = \{-1, 1\}$; some of them examine approaches to multi-class classification problem, here, $\mathcal{L} = \{1, \ldots, K\}$. We also use $n_L = \#X_L$ and $n_U = \#X_U$ to indicate number of labeled and unlabeled separately; and $n_\Sigma = \#X_\Sigma = n_L + n_U$ to denote number of samples in total.

2.1 Semi-supervised classification under the cluster assumption

In the paper "Generalization Error Bounds in Semi-supervised Classification Under the Cluster Assumption" by Rigollet, authors define cluster assumption and its mathematical interpretation.

Definition 2.1. Two points $\mathbf{x}, \mathbf{x}_0 \in \mathcal{X}$ should have the same label y if there is a path between them which passes only through regions of relatively high $P_{\mathcal{X}}$.

It explains two major flaws of modified mathematical formulation of the cluster assumption in terms of connected components of set C.

Definition 2.2. Classes of equivalence of the binary relation R defined on C such that two points $x, y \in C$ satisfy xRy if and only if there exists a continuous map $f : [0, 1] \to C$, such that $f(0) = x$ and $f(1) = y$.

These flaws are as follows:

1. A situation may arise where a set is one connected component, but there is a low density region, so two clusters are desired like on*Figure 3*.

2. A set may have two connected components, but they are too close to each other in a certain sense, so they desired to be identified as a single cluster like on *Figure 4*.

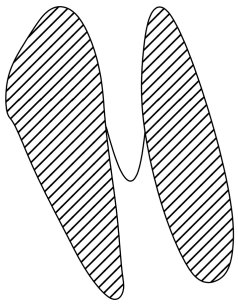


Figure 3: A set with one connected component [4]

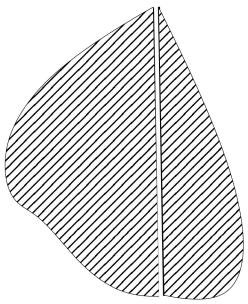


Figure 4: A set with two connected components [4]

Therefore, it provides another definition in terms of a density level sets, i.e. a λ-level set of density p is defined by $\Gamma(\lambda) = \{\mathbf{x} \in \mathcal{X} \colon p(\mathbf{x}) \geq \lambda\}$.

Algorithm

1. We use the unlabeled data X_U. We have to set two parameters: λ, it characterizes a λ-level set of density p, and s_0, it characterizes the scale at which two closely spaced regions can be considered as one cluster, i.e. that two sets C_1 and C_2 are s_0-separated if $d_\infty(C_1, C_2) > s_0$ for some $s_0 > 0$. Using this data, we want to split all the data into clusters.

   The article said that the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm implements this method. DBSCAN has two input parameters: $\epsilon > 0$ — the maximum neighborhood radius from a given point and $M \geqslant 1$ — the minimum number of points required to form a dense region. Those can be expressed through λ and s_0. If a set of points is given in a certain space, the algorithm groups together points that are closely spaced, marking as outliers points that are lonely in areas of low density.

   DBSCAN starts with an arbitrary starting point that has not been visited, and if it contains sufficiently many points, a cluster is initialized. Otherwise, the point is labeled as noise,but later this point could probably be found in ϵ-neighborhood of another point and hence become a part of a cluster. If a point is found to be a dense part of a cluster, its ϵ-neighborhood is also part of that cluster. Hence, all points that are found within the ϵ-neighborhood are added. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

2. After we split the set of unlabeled data into clusters, we assign a single label to each estimated homogeneous region by a majority vote on labeled data.

2.2 Likelihood estimation for semi-supervised classification

Guarantees of improvements for semi-supervised classifiers may currently only be given under restrictive conditions on the data. In the paper *Contrastive Pessimistic Likelihood Estimation for Semi-Supervised Classification* by Loog, authors propose the general way to perform a semi-supervised parameter estimation for likelihood-based classifiers for which, on a full

training set, the estimates are never worse than the supervised solution in terms of the log-likelihood.

In order to show the potential improvements semi-supervised classifiers can deliver, they introduce a novel, generally applicable estimation principle that extends likelihood estimation to the semi-supervised case in a consistent way. In particular, the method is contrastive, which refers to the fact that the objective function takes into account the original supervised solution in an explicit way. This enables the semi-supervised solution to explicitly control the potential improvements over the supervised solution.

In addition, the method is pessimistic, which refers to the fact that the unlabeled data is treated as if it behaves in the worst kind of way, i.e. such that the semi-supervised estimates benefit the least from it. It makes the estimates conservative, but resilient to any possible state in which the unlabeled data can be encountered. This principle is called a maximum contrastive pessimistic likelihood (MCPL) estimation.

The article is based on maximization log-likelihood:

$$L(\theta \mid X_L) = \sum_{i=1}^{n_\Sigma} \log\big(p(\mathbf{x}_i, y_i \mid \theta)\big) = \sum_{k=1}^{K} \sum_{j=1}^{n_{\Sigma_k}} \log\big(p(x_{kj}, k \mid \theta)\big), \tag{1}$$

where $\theta \in \Theta$ denotes all possible model's parameters.

The principal result obtained in this paper is that, for likelihood-based classifiers, semi-supervised parameter estimates $\hat{\theta}_{\text{semi}}$ obtained by means of MCPL are essentially in between the corresponding supervised and the optimal estimates:

$$L(\hat{\theta}_{\text{sup}} \mid X_{V^*}) \leq L(\hat{\theta}_{\text{semi}} \mid X_{V^*}) \leq L(\hat{\theta}_{\text{opt}} \mid X_{V^*}). \tag{2}$$

Let's denote an element of unlabeled data as $\mathbf{u} \in X_U$ for convenience. And the set $X_{V^*} = X_L \cup \big\{(\mathbf{u}_i, v_i^*)\big\}_{i=1}^{n_U}$, assuming that V^* contains the true labels v_i^* belonging to the feature vectors in X_U. Then $\hat{\theta}_{\text{opt}}$ and $\hat{\theta}_{\text{sup}}$ are formally defined as:

$$\hat{\theta}_{\text{opt}} = \arg\max_{\theta \in \Theta} L(\theta \mid X_{V^*}). \tag{3}$$

$$\hat{\theta}_{\text{sup}} = \arg\max_{\theta \in \Theta} L(\theta \mid X_L). \tag{4}$$

Let's define q_{ki} to be the hypothetical posterior $p(k \mid \mathbf{u}_i)$ of observing a particular label k given the feature vector $\mathbf{u}_i$. We may interpret the q_{ki} as soft labels for every $\mathbf{u}_i$. Provided that these posteriors are given, we can express the log-likelihood on the complete data set for any θ as:

$$L(\theta \mid X_L, X_U, q) = L(\theta \mid X_L) + \sum_{i=1}^{n_U} \sum_{k=1}^{K} q_{ki} \log p(\mathbf{u}_i, k \mid \theta). \tag{5}$$

For a given q, the relative improvement of any semi-supervised estimate θ over the supervised solution can now be expressed as follows:

$$CL(\theta, \hat{\theta}_{\text{sup}} \mid X_L, X_U, q) = L(\theta \mid X_L, X_U, q) - L(\hat{\theta}_{\text{sup}} \mid X_L, X_U, q). \tag{6}$$

Objective function becomes:

$$CPL(\theta, \hat{\theta}_{\text{sup}} \mid X_L, X_U) = \min_{q \in \Delta_{K-1}^{M}} CL(\theta, \hat{\theta}_{\text{sup}} \mid X_L, X_U, q), \tag{7}$$

where $\Delta_{K-1}^{M} = \prod_{i=1}^{M} \Delta_{K-1}, \Delta_{K-1} = \left\{(\rho_1, \ldots, \rho_K)^T \in \mathbb{R}^K \mid \sum_{i=1}^{K} \rho_i = 1, \rho_i \geq 0\right\}$.

We are now ready to define MCPL estimation, which extends general likelihood estimation for supervised learners to the general semi-supervised case:

$$\hat{\theta}_{\text{semi}} = \arg\max_{\theta \in \Theta} CPL(\theta, \hat{\theta}_{\text{sup}} \mid X_L, X_U). \tag{8}$$

Authors also remark that the relation between likelihood and classification error rate is not necessarily monotonic and a higher likelihood does not necessarily lead to a lower error.

2.3 Learning with Local and Global Consistency

In the paper "Learning with Local and Global Consistency" by Zhou et al., authors are «designing a classifying function which is sufficiently smooth with respect to the intrinsic structure collectively revealed by known labeled and unlabeled points». Main result of the paper is iteration algorithm allowing to construct such function w.r.t. multi-class classification.

The key to this problem is the prior assumption of consistency, or cluster assumptions:

- nearby points are likely to have the same label.

- points on the same structure are likely to have the same label.

To describe the algorithm, there are introduced two matrices $F \in \mathbb{R}_{+}^{n_\Sigma \times K}$ and $Y \in \mathbb{R}^{n_\Sigma \times K}$. The matrix $F = [F_1^T, \ldots, F_{n_\Sigma}^T]^T$ corresponds to a classification on the dataset X_Σ by labeling each point $\mathbf{x}_i$ as $y_i = \arg\max_{j \leq K} F_{ij}$. The matrix Y is defined as follows:

$$Y_{ij} = \begin{cases} 1, \text{if } \mathbf{x}_i \text{ is labeled as } y_i = j, \\ 0, \text{otherwise.} \end{cases} \tag{9}$$

Finally, the proposed algorithm may be described with pseudo-code as *Algorithm 1.*

Algorithm 1 Algorithm for learning with local and global consistency

Require: Labeled and unlabeled training sets X_L and X_U.

1: Form the affinity matrix W defined as:

$$W_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \text{ if } i \neq j, \\ 0, \text{ otherwise.} \end{cases}$$

2: Construct the matrix $S = D^{-\frac{1}{2}} W D^{\frac{1}{2}}$ where D is a diagonal matrix such that D_{ii} is sum of the i-th row of W.

3: Iterate $F(t+1) = \alpha S F(t) + (1-\alpha) Y$ until convergence, where $\alpha \in (0, 1)$.

4: Let F^* be a limit of sequence $\{F(t)\}_{t=1}^{T}$, then each point $\mathbf{x}_i$ is labeled with:

$$y_i = \arg\max_{j \leq K} F_{ij}^*.$$

This algorithm converges to $F^\star = (\mathbb{I} - \alpha S)^{-1} Y$. However, *Algorithm 1* may be extended with regularisation framework, to do so, authors define a cost function asssosiated with F:

$$Q(F) = \frac{1}{2}\left(\underbrace{\sum_{i,j=1}^{n_\Sigma} W_{ij} \left\| \frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}} \right\|^2}_{\text{smoothness constraint}} + \mu \underbrace{\sum_{i=1}^{n_\Sigma} \|F_i - Y_i\|^2}_{\text{fitting constraint}} \right), \tag{10}$$

where μ is a regularisation parameter. In this case the classification function also may be written in a closed form:

$$F^\star = \arg\min_{F \in \mathbb{R}_{+}^{n_\Sigma \times K}} Q(F) = \frac{\mu}{1-\mu}(\mathbb{I} - \alpha S)^{-1} Y. \tag{11}$$

2.4 Semi-supervised random forests

In the paper *Semi-Supervised Random Forests* by Leistner et al., authors introduce a new approach to the random forest (RF) algorithm [8] for *Semi-Supervised Learning (SSL)* which improves the prediction accuracy with the unlabelled data. RF is an interesting candidate as it can easily be parallelized, solves multi-classes tasks and insensitive to label noise.

RF is an ensemble of decision trees. It uses *bootstraping*, the subsampling with replacement from the origin training data, to train separate trees. Those samples that were not chosen for the traning of the tree are called *Out-Of-Bag* (OBB) examples and used to evaluate the *Out-Of-Bag-Error* (OOBE). ring training, each decision node of the tree creates a set of random test and then selects the best among them using some metrics (e.g. Gini index).

We denote the entire forest as $F = \{f_1, \ldots, f_N\}$, where N is the number of trees in a forest. The estimated probability for predicting class k:

$$p(k \mid \mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} p_n(k \mid \mathbf{x}), \tag{12}$$

where $p_n(k \mid \mathbf{x})$ is the estimated density of class labels of the leaf of the n-th tree. The final multi-class decision $C(X) = \arg\max_{k \in \mathcal{Y}} p(k \mid \mathbf{x})$. Breiman classification *margin* of a labeled sample is denoted as:

$$m_l(\mathbf{x}, y) = p(y \mid \mathbf{x}) - \max_{k \in \mathcal{Y}, k \neq y} p(k \mid \mathbf{x}) \tag{13}$$

For a correct classification $m_l(\mathbf{x}, y) > 0$, then the generalisation error is:

$$GE = E_{(\mathbf{X},Y)}(m_l(\mathbf{x}, y) < 0). \tag{14}$$

Semi-Supervised Learning with Random Forests

Regularization approach. Unlabelled data is used to maximize the margin over the entire random forest.

Margin for the unlabelled data. As unlabelled data doesn't have true labels, we can't use (1) for its margin. We are going to use the margin vector $\mathbf{g_i}(\mathbf{x_u})$ where the coordinates are the margins to the i-th class, then the the predicted label is $i^* = \arg\max_{i \in \mathcal{Y}} g_i(\mathbf{x}_u)$.

Learning. Overall loss:

$$L(g) = \frac{1}{n_L} \sum_{(\mathbf{x},y)\in X_L} l\big(g_y(\mathbf{x})\big) + \frac{\alpha}{n_U} \sum_{\mathbf{x}\in X_U} l\big(m_u(\mathbf{x})\big). \tag{15}$$

This function is non-convex with unlabelled data. To minimize this function we need to introduce additional probability distribution $\hat{\mathbf{p}}$ over unlabelled samples and apply the process of Deterministic Annealing Optimization.

Algorithm

Algorithm 2 Semi-supervised Random Forest

Require: A set of labeled X_L, and unlabeled data X_U.
Require: The size of the forest: N.
Require: A starting heat parameter T_0 and a cooling functions $c(T, m)$.
Train the RF: $F \leftarrow \text{trainRF}(X_L)$.
Compute the OOBE: $e_F^0 \leftarrow \text{oobe}(F, X_L)$.
Set the epoch: $m = 0$.
repeat
  Get the temperature: $T_{m+1} \leftarrow c(T_m, m)$.
  Set $m \leftarrow m + 1$.
  $\forall \mathbf{x}_u \in X_U, y \in Y$: Compute $p^*(k \mid \mathbf{x}_u)$.
  for n from 1 to N **do**
    $\forall \mathbf{x}_u \in X_U$: Draw a random label $\hat{y}_u$ from $p^*(\cdot \mid \mathbf{x}_u)$ distribution.
    Set $X_n = X_L \cup \{(\mathbf{x}_u, \hat{y}_u) \mid \mathbf{x}_u \in X_U\}$.
    Re-train the tree: $f \leftarrow \text{trainTree}(X_n)$.
  end for
  Set $e_F^m \leftarrow \text{oobe}(F, X_L)$.
until Stopping condition
if $e_F^m > e_F^0$ **then**
  Reset the RF: $F \leftarrow \text{trainRF}(X_L)$.
end if
Output the forest F.

2.5 Self-learning algorithm

In the paper "A Transductive Bound for the Voted Classifier with an Application to Semi-supervised Learning" by Amini, Usunier, and Laviolette, authors are investigating a transductive binary classification problem solved by a majority voting classifier, or an ensemble. Main academic result of the paper is a theorem stating a closed form of a tight bound

over a joint transductive Bayes risk, which is further applied to a margin-based algorithm for pseudo-labeling. The idea of margin thresholding is not new, but the novelty of the algorithm is that value of threshold is chosen automatically.

Speaking formally, authors are considering learning algorithms that work in a fixed hypothesis space $\mathcal{H}$. After observing both labeled data and unlabeled data, the goal is to fit posterior distribution of classifier weights Q over $\mathcal{H}$ so that ensemble B_Q has the smallest possible risk on unlabeled examples:

$$B_Q = \operatorname{sign}\left[\mathbb{E}_{h\sim Q} h(\mathbf{x}_u)\right] \ \forall \mathbf{x}_u \in X_U. \tag{16}$$

Let $y_u \in \mathcal{L}$ be a real pseudo-label of $\mathbf{x}_u \in X_U$ then a transductive joint Bayes risk is:

$$R_{u,\theta}(B_Q) \stackrel{def}{:=} \frac{1}{\#X_U} \sum_{\mathbf{x}_u \in X_U} \mathbb{J}\left(B_Q(\mathbf{x}_u) \neq y_u \wedge m_Q(\mathbf{x}_u) > \theta\right), \tag{17}$$

where $m_Q(\bullet) = |\mathbb{E}_{h\sim Q} h(\bullet)|$ denotes an unsigned margin function and θ stands for margin threshold. Margin is used as an indicator of confidence that a sample lies far enough from inter-class boundary, correspondingly, a Bayes classifier should make errors mostly on low margin regions.

Consider also $\mathbb{E}_{\text{uf}}(\bullet)$ to be an expectation w.r.t. uniformly distributed variable over X_U and $p_{\text{uf}}(\bullet)$ to be a probability distribution over X_U, then *Theorem 1* gives the bound for joint Bayes risk.

Theorem 1 (Bound on transductive joint Bayes risk)**.** *Suppose B_Q is as in* (16)*. Then for all Q, all $\delta \in (0, 1]$, all $\theta \geq 0$ with probability at least $1 - \delta$.*

$$R_{u,\theta}(B_Q) \leq \inf_{\gamma\in(0,1]} \left\{ p_{\text{uf}}\big(\theta < m_Q(\mathbf{x}_u) < \gamma\big) + \frac{1}{\gamma} \lfloor K_u^\delta(Q) + M_Q^{\leqslant}(\theta) - M_Q^{<}(\gamma) \rfloor_+ \right\}. \tag{18}$$

Here, $K_{\text{uf}}^\delta(Q) \leq \mathbb{E}_{\text{uf}}\left[m_Q(\mathbf{x}_u)\right]$, $M_Q^{\lhd}(t) = \mathbb{E}_{\text{uf}}\Big\{ m_Q(\mathbf{x}_u) \mathbb{J}\big[m_Q(\mathbf{x}_u) \lhd t\big] \Big\}$ and $\lfloor \mathsf{x} \rfloor_+ = \mathsf{x} \cdot \mathbb{J}(\mathsf{x} > 0)$.

Then, the most margin-confident unlabeled samples are the ones that have a small conditional Bayes error:

$$R_{u|\theta}(B_Q) \stackrel{def}{:=} p_{\text{uf}}\big(B_Q(\mathbf{x}_u) \neq y \mid m_Q(\mathbf{x}_u) > \theta\big) = \frac{R_{u,\theta}(B_Q)}{p_{\text{uf}}(m_Q(\mathbf{x}_u) > \theta)}. \tag{19}$$

Finally, the self-learning algorithm may be written as a pseudo-code as *Algorithm 3*:

Algorithm 3 Self-learning algorithm

Require: Labeled and unlabeled training sets X_L and X_U.

Train classifier H on X_L
Set $X_{\not\subset} \leftarrow \varnothing$
repeat
  Compute the margin threshold $\theta^\star$ minimising (19) from (18).
  $S \leftarrow \big\{(\mathbf{x}, y) \mid \mathbf{x} \in X_U; m_Q \geq \theta^\star, y = \operatorname{sign}\big[H(\mathbf{x})\big]\big\}$.
  $X_{\not\subset} \leftarrow X_U \cup S, X_U = X_U \setminus S$.
  Learn a classifier H by optimising a global loss function on X_L and $X_{\not\subset}$.
until X_U is empty or there are no adds to $X_{\not\subset}$
Output the final classifier H.

3 Experimental setup

In this section we firstly describe all the sets of data we are using in our experiments, both real and synthetic; and motivate a synthetic data construction for a particular semi-supervised assumption. Secondly, we briefly indicate and link the models for experiments to papers discussed in the previous *Section 2*.

The source code of the project can be found on our GitHub [10], we refer to our implemented package for semi-supervised learning as `sslearn`. Also consider a common naming `sklearn` for machine-learning package in python [11].

3.1 Datasets

3.1.1 Real

MNIST

Original MNIST [12] dataset is a grayscale image collection of handwritten digits of size 28×28 pixels. It has 60000 images for training and 10000 images for testing, image labels are distributed uniformly both in train and in test sets. For our experiments we extracted subset of images from a test set with labels 4 and 9. Further, we will be referring to this subset as `mnist_4_9`.

Fashion MNIST

Fashion MNIST [13] is a dataset of Zolando's article images, its characteristics are the same as MNIST's. In the very similar manner we create an experimental subset `fashion_mnist_4_9` (4 stands for «coat», 9 stands for «ankle boot»).

Pendigits

Pendigits [14] is another dataset of grayscale images with total number of 10992 samples. Each image is represented as 16×1 vector. Again, for experiments we extract only images labeled as 4 and 9 resulting a `pendigits_4_9` subset.

MAGIC Gamma Telescope

This dataset «simulates registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique» [15]. Total number of samples is 19020 and each sample has 10 features. For experiments we, traditionally, use a reduced subset of 2000 samples which we refer to as `telescope_2000`.

Breast Cancer Wisconsin

Breast Cancer Wisconsin dataset [16] contains information about patients' monitoring. There are 699 samples in total, each sample has 9 features. In our notation this dataset set is marked as `breast_w`.

Banknote authentication

From the description [17], «data were extracted from images that were taken from genuine and forged banknote-like specimens. Wavelet Transform tool were used to extract features from images». In total, this dataset has 1372 samples of 4 features each and has `banknotes` naming in further narration.

King-Rook vs. King-Pawn

This dataset [18] contains a description of a chess board for king rook vs king pawn endgame as well as the game's result — win or not win for whites. In total, there are 3196 samples having 36 features. Consider `kr_vs_kp` abbreviation.

Balance scale

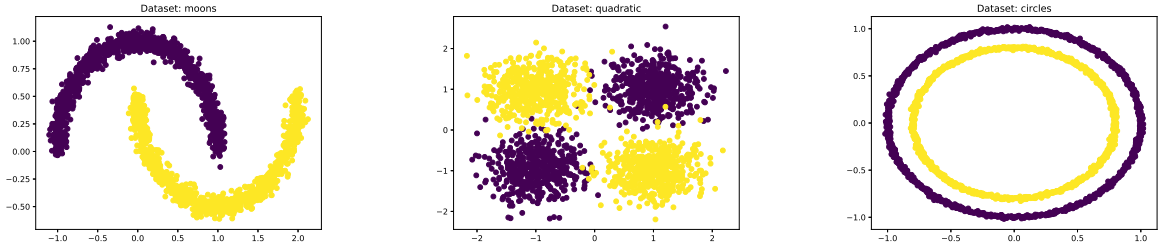
Originally, it was a multiclass dataset «generated to model psychological experimental results» [19] containing 625 samples with 4 features. However, as some of our models are initially were designed to handle only binary problems, we binarise it w.r.t two first classes resulting `balance_scale` subset.

3.1.2 Synthetic

3.1.2.1 Cluster assumption

Goal: check the operation of our algorithms on the data that form separate clusters, with points in the same cluster most likely having a common label.

To solve this problem, we generated three different data options, which are presented on the *Figure 5*. For this we used standard functions of free software machine learning Scikit-learn library for the Python programming language, namely: `sklearn.datasets.make_moons`, `sklearn.datasets.make_blobs` and `sklearn.datasets.make_circles`. In our notation these datasets are marked as `moons`, `quadratic`, `circles`.



(a) Moons *(b)* Quadratic *(c)* Circles

Figure 5: Data visualisation for cluster assumption

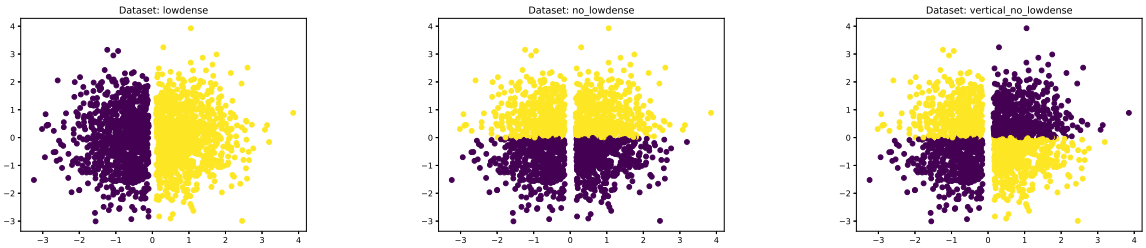
3.1.2.2 Low density separation assumption

Goal: consider such datasets, when this assumption does not perform, i.e. boundaries between classes do not lie in low-density regions. Semi-supervised algorithm must draw a boundary in low-density region, thus we will prove that SSL algorithms are not suitable for such data.

To solve this problem we generate three datasets, which are presented in the Figure 6. The dataset in the Figure 6 (a) demonstrates the case when low density separation assumption is performed. Figures 6 (b) and 6 (c) are examples of non-compliance of the assumption.

3.1.2.3 Manifold assumption

Goal: check the manifold assumption, where data are located in lower dimensional manifolds. For this purpose datasets of `spirals`(two twisted spirals in two-dimensional space)



(a) Lowdense *(b)* No_lowdense *(c)* Vertical_no_lowdense

Figure 6: Data visualisation for low density separation assumption

and `overlapping_planes`(two overlapping planes in three-dimensional space) have been constructed.

(a) Spirals *(b)* Overlapping_planes

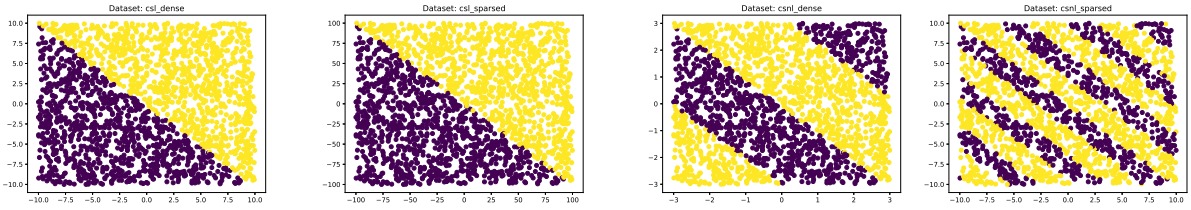
Figure 7: Data visualisation for manifold assumption

3.1.2.4 Causality

Goal: check the hypothesis behind causal and anticausal learning. If the features are caused by the labels ($Y \rightarrow X$) then we suggest improvements in performance as the information about labels is contained in features, otherwise ($X \rightarrow Y$) no improvements is expected.
To conduct an experiment we have generated a set of pure causal and anticausal data. To observe the behaviour of the generated datasets at the scale sparse and dense approaches are used. Sparse datasets consider better separability between classes as the number of points is kept constant (2000 samples), but the feasible set of possible values in increased. For Gaussian distributions, sparsity means larger distances between the means and smaller intersections between the classes (influenced by variances).

Causal data: to generate the independent labels we used an approach of mathematical functions with additional noise. The features - point coordinates - don't gain any information

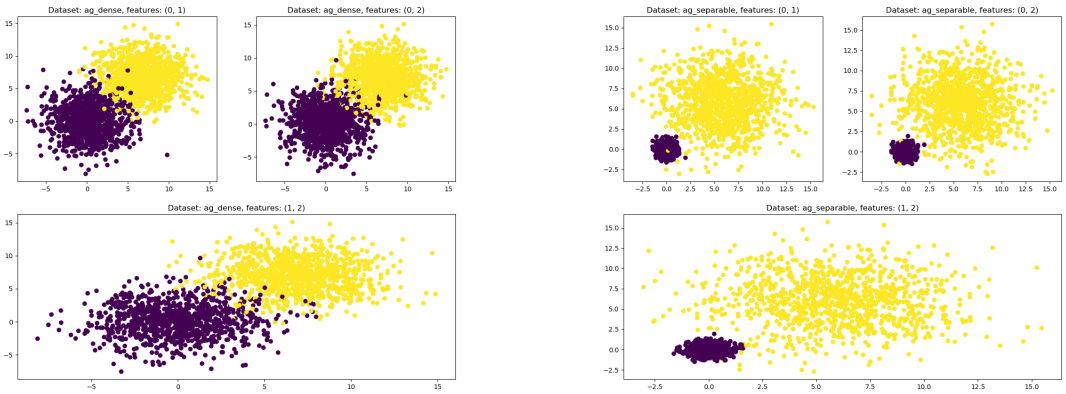
about the labels - function values. Gaussian noise was applied to avoid overfitting, the linear and sine functions were binarized with a sign extractor.



(a) Linear dense *(b)* Linear sparse *(c)* Non-linear dense *(d)* Non-linear sparse

Figure 8: Data visualisation for causal data

Anticausal data: to generate datasets with features storing the information about the labels Gaussain distribution was used. When the point is generated it already keeps the information about the distribution it comes from.



(a) Two dense Gaussians *(b)* Two separable Gaussains

Figure 9: Data visualisation for anticausal data

3.2 Models

In *Table 2* is presented information about models that were used in the experiments.

Column «Model Identifier» stands for model's abbreviation, «Baseline Model» describes a classifier that was used as a supervised model upon which semi-supervised techniques were applied, «Baseline Model Source» signifies the source package the baseline classifier implementation is adopted from. The long dash means that no baseline model and, hence, baseline source exists for model.

Paper	Model Identifier	Baseline Model	Baseline Model Source
[4]	CLUSTERING	Random Forest	`sklearn`
[5]	LDA	Linear Discriminant Analysis	`sslearn`
[7]	RF	Random Forest	`sslearn`
[9]	SLA	Random Forest	`sklearn`
[6]	LGC	—	—

Table 2: Information about experimental models

3.3 Evaluation

Main evaluation score in our experiments was *accuracy* metrics. In order not to produce more clean results and to be less dependent on standard random number generator, each experiment is conducted for 20 times with fixed random seed for every model, and then averaging is performed.

4 Experimental results

In this section we present and discuss obtained results. The data used in our experiments is a mix of real world datasets and synthetic datasets described in *Section 3*. Results are represented in two forms: graphs and tables. In the captions of each graph, e.g. *Figure 10*, depicted datasets are listed in the same order as drawn on the graph, (S) and (R) stand for real world and synthetic dataset correspondingly. Each point represents one algorithm, if the point is green, then semi-supervised modification is helpful in comparison with the baseline algorithm, otherwise baseline model is better, black points mean that accuracy for both cases was the same. The value $lsize$ is the percentage of labelled examples in the whole set of training instances.

4.1 Real datasets

The section is devoted to testing performances of our algorithms real datasets that we would not able to distribute between assumptions. Which is demonstrated on *Table 7*, *Table 3*, *Table 4*, *Table 5* and *Table 6*. Even though that we do not know exactly the assumption these datasets follow we may notice that semi-supervised approaches like SLA, RF — which performs better than supervised baseline on every imaged-dataset — and LDA work better than supervised ones on datasets with images (`pendigits_4_9`, `fashion_mnist_4_9`, `mnist_4_9`), especially in the zone $n_U \gg n_L$. On the contrary, images classification was a really difficult task for LGC algorithm even on `fashion_mnist_4_9` dataset that happened to be a piece of a cake for the other models.

4.2 Low density assumption

In this section we want to check that semi-supervised learning algorithms work worse for non-low density datasets than for datasets, satisfying low-density separation assumption. On *Table 22*, *Table 23*, *Table 24* the comparison analysis for different algorithms is shown. We can see that for small $lsize$ (0.5% and 1%) of labeled data SSL clustering and lda algorithms clearly show worse results on non-lowdensity datasets. With increase the number of labelled data difference between accuracy SSL algorithms and their baselines is not observed, except for SSL clustering algorithm (it works much worse on non-lowdensity data). Thus, we

can conclude that only some of semi-supervised learning algorithms work worse for non-low density datasets when we have a small amount of labeled data, and no low-density data is not particularly sensitive to semi-supervised paradigm when we have more labeled data.

4.3 Clustering assumption

This section is devoted to testing the performance of our algorithms on data that form discrete regions. We want to check it since some of the considering algorithms are based on the cluster hypothesis, so they must work better for such types of datasets.

On *Table 13*, *Table 18*, *Table 19* the comparison analysis of different models for clustering datasets is shown. We can see that for small $lsize$ (0.5% and 1%) semi-supervised approach is helpful for the majority of datasets as evidenced by the fact that most algorithms show better results then their baselines. Moreover, it can be seen that with a small amount of labeled data, the clustering algorithm works more than 4% better than its baseline and has the greatest accuracy relative to other algorithms. With increase the number of labelled data the absolute difference between the accuracy of baselines and the SSL-algorithms tends to decrease. Thus, the SSL seems to be very helpful when we have a small amount of labeled data and neutral otherwise since the accuracy of SSL algorithms and their baselines is almost the same.

4.4 Manifold assumption

This section deals with datasets with manifold assumption. We want to test the algorithms for the possibility of selecting small dimensional manifolds located in space with larger dimensions.

A well-known problem of many statistical methods and learning algorithms is the so-called curse of dimensionality. It is related to the fact that volume grows exponentially with the number of dimensions, and an exponentially growing number of examples is required for statistical tasks such as the reliable estimation of densities. This is a problem that directly affects generative approaches that are based on density estimates in input space. A related problem of high dimensions, which may be more severe for discriminative methods, is that pairwise distances tend to become more similar, and thus less expressive. If the data happen to lie on a low-dimensional manifold, however, then the learning algorithm can essentially

operate in a space of corresponding dimension, thus avoiding the curse of dimensionality.

On *Table 20* (spirals) and *Table 21* (overlapping planes) the comparison analysis of different models for manifold assumption datasets is shown.

Spirals: For all $lsize$'s clustering algorithm shows approximately the same quality, but for small $lsize$ (0.5% and 1%) this quality is noticeably higher than other algorithms. Starting from $lsize$ equals to 5%, RF and SLA algorithms start to improve the quality of classification bringing it to almost 1. LDA and LGC algorithms did not show good results on this dataset. It can be concluded that linear algorithms are not suitable for this dataset, but Random Forest (it is a baseline of clustering, RF and LSA algorithms) has proved to be a good one.

Overlapping planes: In this dataset the algorithms showed good results from the beginning and with the increase of $lsize$ only improved the quality of classification. However, the algorithms SLA (Baseline) and LGC have always been in outsiders and have not shown any improvement with the increase in $lsize$.

4.5 Causality

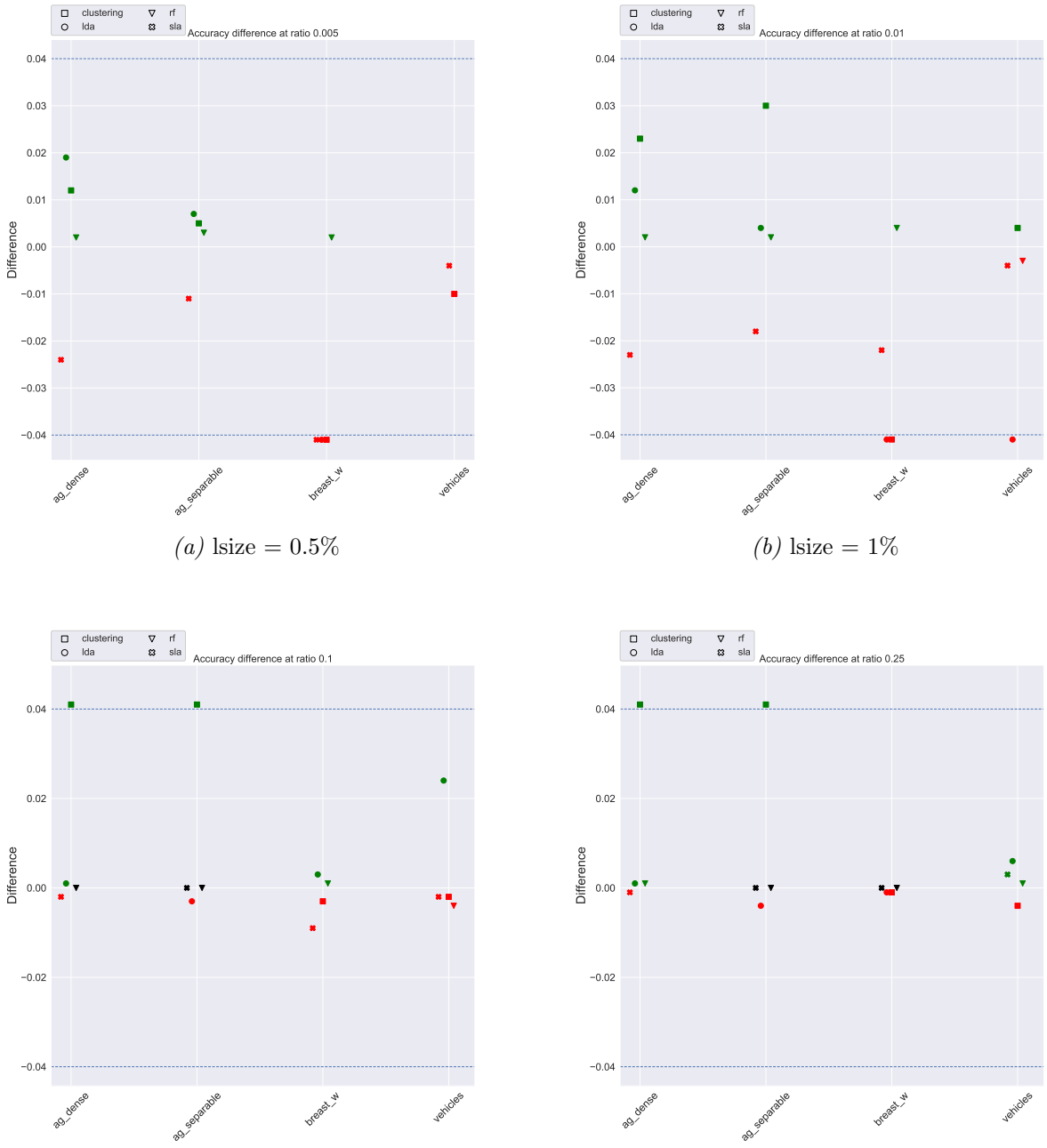
In this section we check the hypothesis, that semi-supervised learning is more helpful for datasets with anticausal features then for causal data. Real world datasets used in this section were analyzed for causal directions in [3]. `breast_w` and `vehicles` stand for anticausal datasets and `balance_scale` and `kr_vs_kp` are used as causal examples. *Statistical* analysis can be done using *Figure 10* and *Figure 11*, for accurate analysis on each dataset at Appendix A.

Anticausal: On *Figure 10* the comparison analysis of different models for anticausal datasets is shown. For small $lsize$ we see that the difference in accuracy is green and semi-supervised approach is helpful for the majority of datasets. As we increase the number of labelled values the accuracy increase tends to converge to a neighbourhood of zero. It means that SSL is helpful when the number of labelled samples is small, but at some point the increase of labelled points is redundant as it doesn't improve the accuracy. We can also notice that the threshold value of the maximum number of helpful examples is different: synthetic data needs less examples to benefit from the SSL approach, real world data needs more.

Causal: The same comparison graphs for causal data are shown on *Figure 11*. The first

thing to notice is the statistical distribution of green and red points: the density of red points is greater. For 0.5% of labelled data the majority of points can be found in the neighbourhood of zero, but there are some outliers. As we increase the number of labelled points the average accuracy difference decreases and more and more points get red. It confirms the hypothesis that causal data is not particularly sensitive to semi-supervised paradigm. However, for real world data there is at least one exception with the dataset `balance_scale`: SSL algorithms increase their accuracy and outperform baselines for big numbers of $lsize$. It seems to suggests that the real world data is complicated by itself and causality assumption is not the only one to be taken into account when we discuss semi-supervised learning pros and cons.

To summarize, causality hypothesis looks approved for synthetic data, but for real world datasets it cannot be applied directly because of the data complexity and possible hidden connections which are not considered by pure causality assumption.



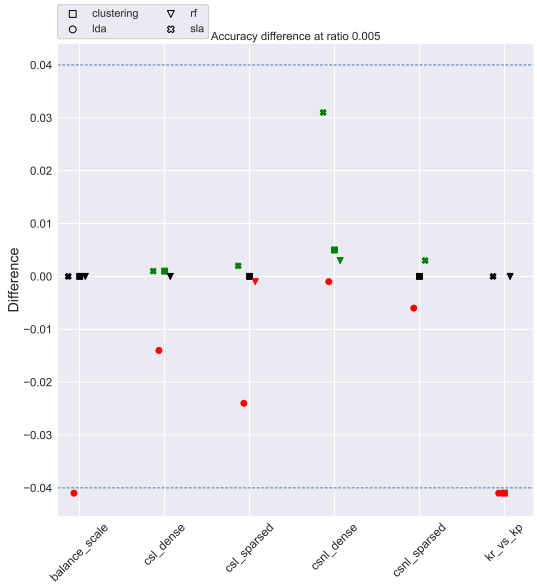
(a) lsize = 0.5%

(b) lsize = 1%

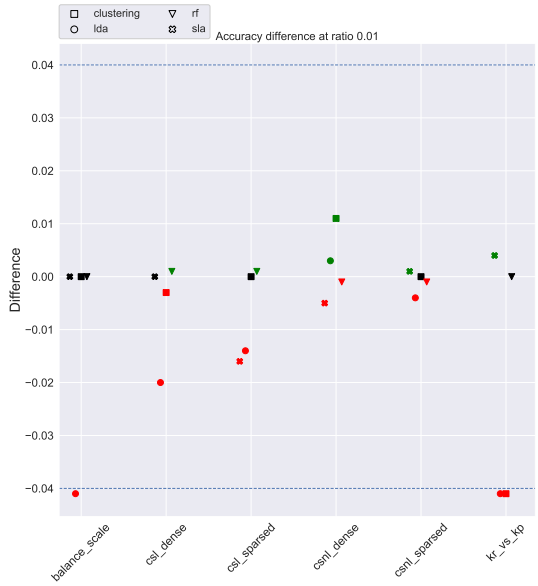
(c) lsize = 10%

(d) lsize = 25%

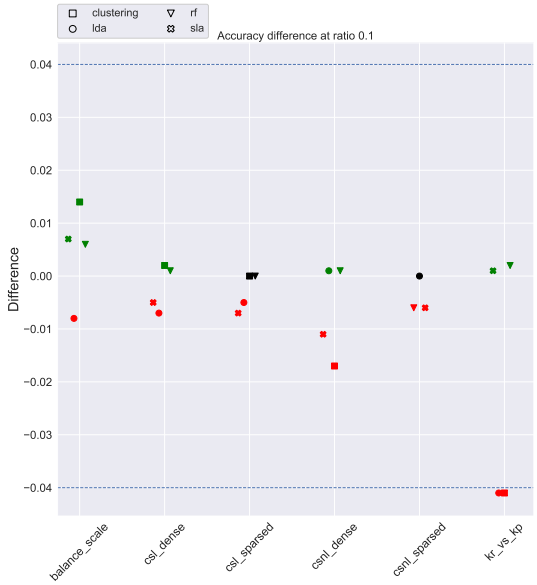
Figure 10: Anticausal results: ag_dense (S), ag_separable (S), breast_w (R), vehicles (R)



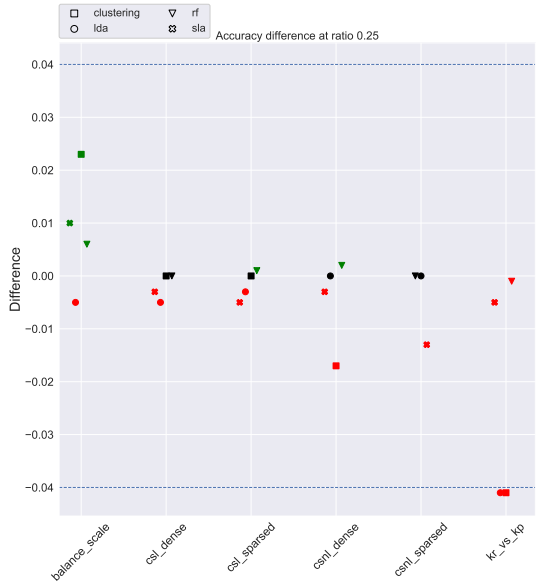
(a) lsize = 0.5%



(b) lsize = 1%



(c) lsize = 10%



(d) lsize = 25%

Figure 11: Causal results: balance_scale (R), csl_dense (S), csl_sparsed (S), csnl_dense (S), csnl_sparsed (S), kr_vs_kp (R)

5 Conclusion

In this paper we observed several semi-supervised learning algorithms and discussed and checked semi-supervised learning major assumptions: clustering, low density and manifold hypothesis. We also examined how SSL algorithms based on concrete assumption behave with different datasets.

As expected, semi-supervised CLUSTERING algorithm showed the best results on datasets based on the *clustering hypothesis*. Moreover, other considered algorithms improved accuracy of prediction in comparison with the baseline on a small amount of labelled data.

What concerns the *low density hypothesis* the results highly depend on the amount of labelled data. CLUSTERING and LDA algorithms worked worse than the baseline for non-low density datasets with a small amount of labelled samples. For huge labelled subsamples the majority of SSL algorithms showed the same results as baseline.

Manifold as it turned out is a very strong assumption. It's a complicated problem by itself to automatically detect this property for a dataset. The variety of manifolds is huge and complementary research is needed to analyze the data for this assumption. Moreover, the results for a manifold assumption are good and the majority of algorithms showed an increase in performance for semi-supervised modifications.

In this paper we also denoted the importance of datasets structure analysis and discovered the problem of causal direction between features and labels in machine learning applications. *Causality hypothesis* was clearly confirmed with synthetic datasets without internal dependencies, but for real world data the situation looks more complicated and causality assumption while still being correct in many cases can not be applied standalone.

References

[1] O. Chapelle and A. Zien. "Semi-supervised classification by low density separation". In: *AISTATS* (2005), pp. 57–64.

[2] Alexander Mey and Marco Loog. "Improvability Through Semi-Supervised Learning: A Survey of Theoretical Results". In: *ArXiv* abs/1908.09574 (2019).

[3] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris M. Mooij. "On causal and anticausal learning". In: *ICML*. 2012.

[4] Philippe Rigollet. "Generalization Error Bounds in Semi-supervised Classification Under the Cluster Assumption". In: *J. Mach. Learn. Res.* 8 (Dec. 2007), pp. 1369–1392.

[5] Marco Loog. *Contrastive Pessimistic Likelihood Estimation for Semi-Supervised Classification*. 2015. arXiv: 1503.00269 [stat.ML].

[6] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Scholkopf. "Learning with Local and Global Consistency". In: (2004).

[7] Christian Leistner, Amir Saffari, Jakob Santner, and Horst Bischof. *Semi-Supervised Random Forests*. Sept. 2009. DOI: 10.1109/ICCV.2009.5459198.

[8] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[9] Massih Amini, Nicolas Usunier, and François Laviolette. "A Transductive Bound for the Voted Classifier with an Application to Semi-supervised Learning". In: *Advances in Neural Information Processing Systems 21*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Curran Associates, Inc., 2009, pp. 65–72. URL: http://papers.nips.cc/paper/3444-a-transductive-bound-for-the-voted-classifier-with-an-application-to-semi-supervised-learning.pdf.

[10] Shlenskii V., Rodina S., Borisov D., Kotova M., and Petrova A. and. *MSIAM M2 Modelling Seminar: Semi-Supervised Learning Research, Algorithms Comparison*. URL: https://github.com/SSL-GRENOBLE/SSL.

[11] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[12] Yann LeCun et al. *The Mnist Database of handwritten digits*. URL: http://yann.lecun.com/exdb/mnist/.

[13] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].

[14] Fevzi. Alimoglu E. Alpaydin. *Pen-Based Recognition of Handwritten Digits Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits.

[15] P. Savicky R. K. Bock. *MAGIC Gamma Telescope Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope.

[16] Olvi Mangasarian William H. Wolberg. *Breast Cancer Wisconsin (Original) Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original).

[17] Volker Lohweg. *Banknote authentication Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/banknote+authentication.

[18] Peter Clark Alen Shapiro. *Chess (King-Rook vs. King-Pawn) Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King-Pawn%29.

[19] Tim Hume. *Balance Scale Data Set*. URL: http://archive.ics.uci.edu/ml/datasets/balance+scale.

Appendices

A Joint result tables

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	6	13	68	137	343	686
CLUSTERING	.869	.871	.894	.894	.897	.904
CLUSTERING (Baseline)	.78	.81	.934	.967	.985	.99
LDA	**.912**	**.962**	.969	.969	.969	.972
LDA (Baseline)	.883	.958	**.974**	.975	.975	.977
RF	.785	.855	.972	**.988**	**.996**	**.998**
RF (Baseline)	.782	.853	.968	.987	.996	.998
SLA	.786	.803	.922	.956	.982	.99
SLA (Baseline)	.78	.81	.934	.967	.985	.99
LGC	.444	.444	.445	.445	.444	.445

Table 3: Accuracy score for `banknotes` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	9	19	99	199	497	995
CLUSTERING	.709	.831	.911	.938	.962	**.974**
CLUSTERING (Baseline)	.716	.834	.913	.944	.964	.973
LDA	.0	.0	.0	.0	.0	.904
LDA (Baseline)	.629	.722	.876	.877	.652	.888
RF	**.73**	.847	.923	**.95**	**.966**	.973
RF (Baseline)	.727	.829	.916	.946	.965	.973
SLA	.716	**.86**	**.931**	.949	.966	.974
SLA (Baseline)	.716	.834	.913	.944	.964	.973
LGC	.493	.493	.493	.493	.493	.493

Table 4: Accuracy score for `mnist_4_9` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	.982	.99	.998	**.999**	**.999**	**.999**
CLUSTERING (Baseline)	.984	.994	.998	.999	.998	.999
LDA	.0	.991	.995	.997	.998	.99
LDA (Baseline)	.945	.952	.989	.993	.988	.975
RF	**.994**	.997	**.999**	.999	.999	.999
RF (Baseline)	.992	.995	.998	.999	.999	.999
SLA	.994	**.999**	.999	.998	.998	.998
SLA (Baseline)	.984	.994	.998	.999	.998	.999
LGC	.5	.5	.5	.5	.5	.5

Table 5: Accuracy score for `fashion_mnist_4_9` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	21	109	219	549	1099
CLUSTERING	.885	.954	.988	.993	.994	.996
CLUSTERING (Baseline)	.921	.961	.985	.988	.993	.996
LDA	.0	.835	**.995**	**.996**	.996	.996
LDA (Baseline)	.875	.893	.993	.996	.996	.997
RF	**.939**	**.977**	.993	.996	.997	**.998**
RF (Baseline)	.93	.972	.992	.995	.997	.998
SLA	.939	.976	.982	.988	.993	.995
SLA (Baseline)	.921	.961	.985	.988	.993	.996
LGC	.901	.937	.977	.992	**.998**	.997

Table 6: Accuracy score for `pendigits_4_9` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	.649	.667	.698	.714	.731	.743
CLUSTERING (Baseline)	.65	.698	**.79**	**.808**	**.833**	**.846**
LDA	.0	.691	.759	.767	.78	.778
LDA (Baseline)	.617	.677	.764	.776	.787	.785
RF	.663	.71	.781	.799	.823	.834
RF (Baseline)	**.666**	**.712**	.783	.799	.824	.836
SLA	.642	.691	.773	.791	.826	.838
SLA (Baseline)	.649	.698	.78	.801	.829	.844

Table 7: Accuracy score for `telescope_2000` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	2	3	16	33	84	168
CLUSTERING	**.854**	**.853**	**.854**	.855	.854	**.852**
CLUSTERING (Baseline)	.854	.853	.839	.841	.831	.831
LDA	.0	.0	.81	.85	.857	.85
LDA (Baseline)	.854	.853	.825	**.858**	**.862**	.847
RF	.854	.853	.848	.853	.844	.844
RF (Baseline)	.854	.853	.843	.847	.838	.836
SLA	.854	.853	.846	.848	.841	.844
SLA (Baseline)	.854	.853	.839	.841	.831	.831
LGC	.854	.853	.826	.836	.838	.84

Table 8: Accuracy score for `balance_scale` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	3	6	34	68	170	341
CLUSTERING	.739	.829	.944	.957	.963	.97
CLUSTERING (Baseline)	.819	.912	.945	.96	.964	.971
LDA	.0	.0	.933	.951	.958	.961
LDA (Baseline)	.751	.885	.933	.948	.959	.962
RF	**.944**	**.95**	**.966**	**.968**	**.971**	**.973**
RF (Baseline)	.942	.946	.965	.967	.971	.973
SLA	.69	.89	.932	.951	.964	.97
SLA (Baseline)	.819	.912	.945	.96	.964	.971
LGC	.65	.65	.65	.65	.649	.649

Table 9: Accuracy score for `breast_w` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	15	31	159	319	799	1598
CLUSTERING	.522	.522	.522	.522	.522	.522
CLUSTERING (Baseline)	.709	.778	.933	.956	**.977**	**.984**
LDA	.0	.0	.0	.0	.186	.374
LDA (Baseline)	.63	.638	.896	.924	.936	.939
RF	.688	.773	.931	**.957**	.973	.979
RF (Baseline)	.688	.773	.931	.955	.974	.983
SLA	.709	**.782**	**.938**	.957	.972	.982
SLA (Baseline)	.709	.778	.933	.956	.977	.984
LGC	**.72**	.733	.789	.829	.899	.946

Table 10: Accuracy score for `kr_vs_kp` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	.72	.78	.817	.817	.819	.82
CLUSTERING (Baseline)	.708	.757	.757	.753	.753	.753
LDA	.979	**.99**	**.992**	**.992**	**.993**	**.994**
LDA (Baseline)	.96	.978	.99	.991	.992	.994
RF	**.987**	.988	.991	.991	.992	.993
RF (Baseline)	.985	.986	.99	.991	.991	.993
SLA	.952	.951	.982	.984	.989	.992
SLA (Baseline)	.976	.974	.984	.986	.99	.993
LGC	.986	.985	.989	.991	.989	.988

Table 11: Accuracy score for `ag_dense` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	.827	.859	.885	.889	.891	.889
CLUSTERING (Baseline)	.822	.829	.836	.841	.838	.83
LDA	.934	.966	.971	.971	.973	.974
LDA (Baseline)	.927	.962	.974	.974	.977	.977
RF	**.992**	**.995**	.998	**.999**	**.999**	**1.0**
RF (Baseline)	.989	.993	.998	.999	.999	1.0
SLA	.975	.971	**.999**	.999	.999	.999
SLA (Baseline)	.986	.989	.997	.999	.999	1.0
LGC	.5	.5	.5	.5	.5	.5

Table 12: Accuracy score for `ag_separable` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.629**	**.723**	.919	.955	.963	.967
CLUSTERING (Baseline)	.57	.661	.867	.904	.943	.96
LDA	.501	.503	.499	.498	.494	.485
LDA (Baseline)	.5	.503	.5	.498	.494	.485
RF	.603	.691	**.952**	**.996**	**1.0**	**1.0**
RF (Baseline)	.603	.691	.952	.996	1.0	1.0
SLA	.587	.698	.9	.948	.978	.991
SLA (Baseline)	.589	.679	.905	.95	.982	.992
LGC	.5	.5	.5	.5	.5	.5

Table 13: Accuracy score for `circles` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.95**	.969	**1.0**	**1.0**	**.999**	**1.0**
CLUSTERING (Baseline)	.949	**.972**	.996	.998	.999	1.0
LDA	.881	.912	.955	.965	.979	.988
LDA (Baseline)	.895	.932	.964	.972	.984	.992
RF	.873	.898	.947	.959	.968	.97
RF (Baseline)	.873	.897	.946	.958	.968	.97
SLA	.823	.862	.925	.947	.967	.979
SLA (Baseline)	.822	.862	.935	.952	.97	.981
LGC	.69	.743	.759	.739	.74	.758

Table 14: Accuracy score for `csl_dense` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.982**	**.986**	.996	**.999**	**1.0**	**1.0**
CLUSTERING (Baseline)	.982	.986	**.998**	.999	1.0	1.0
LDA	.857	.918	.945	.973	.98	.991
LDA (Baseline)	.881	.932	.957	.978	.983	.99
RF	.839	.896	.95	.967	.978	.98
RF (Baseline)	.84	.895	.949	.967	.977	.98
SLA	.818	.84	.926	.945	.967	.979
SLA (Baseline)	.816	.856	.934	.952	.972	.98
LGC	.865	.879	.964	.971	.979	.978

Table 15: Accuracy score for `csl_sparsed` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	.504	.504	.493	.493	.494	.494
CLUSTERING (Baseline)	.504	.504	.498	.49	.492	.498
LDA	.517	.514	.544	.546	.559	.567
LDA (Baseline)	**.523**	**.518**	.546	.546	.559	.567
RF	.497	.503	.588	.654	**.791**	.854
RF (Baseline)	.497	.504	**.591**	**.66**	.791	.854
SLA	.522	.51	.56	.625	.753	.85
SLA (Baseline)	.519	.509	.564	.631	.766	**.857**

Table 16: Accuracy score for `csnl_sparsed` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.721**	**.867**	.959	.962	.975	.978
CLUSTERING (Baseline)	.716	.856	**.967**	**.979**	**.992**	**.996**
LDA	.533	.522	.509	.524	.533	.504
LDA (Baseline)	.534	.519	.508	.523	.533	.504
RF	.576	.673	.876	.906	.924	.931
RF (Baseline)	.573	.674	.875	.905	.922	.929
SLA	.582	.628	.859	.902	.942	.963
SLA (Baseline)	.551	.633	.871	.913	.945	.966
LGC	.461	.514	.513	.477	.52	.494

Table 17: Accuracy score for `csnl_dense` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.978**	**.979**	.979	.983	.986	.988
CLUSTERING (Baseline)	.874	.888	.944	.971	.983	.988
LDA	.844	.856	.87	.873	.875	.878
LDA (Baseline)	.847	.859	.876	.88	.883	.883
RF	.898	.961	**.998**	**.999**	**1.0**	**1.0**
RF (Baseline)	.896	.96	.997	.999	1.0	1.0
SLA	.856	.896	.97	.985	.994	.996
SLA (Baseline)	.852	.918	.977	.988	.995	.997
LGC	.5	.5	.5	.5	.5	.5

Table 18: Accuracy score for `moons` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.841**	**.944**	**.978**	**.98**	**.982**	.981
CLUSTERING (Baseline)	.668	.799	.972	.978	.982	.981
LDA	.591	.549	.519	.515	.504	.493
LDA (Baseline)	.597	.552	.52	.515	.504	.493
RF	.763	.905	.976	.978	.981	.983
RF (Baseline)	.759	.901	.975	.976	.981	.983
SLA	.67	.82	.97	.975	.979	**.985**
SLA (Baseline)	.652	.81	.968	.978	.98	.984
LGC	.5	.5	.5	.5	.5	.5

Table 19: Accuracy score for `quadratic` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	10	20	100	200	500	1000
CLUSTERING	**.886**	**.886**	**.886**	.886	.885	.889
CLUSTERING (Baseline)	.544	.606	.834	.928	.984	.997
LDA	.567	.564	.587	.597	.604	.599
LDA (Baseline)	.569	.564	.587	.597	.604	.6
RF	.552	.611	.872	**.967**	**.995**	.998
RF (Baseline)	.552	.612	.87	.963	.994	**1.0**
SLA	.556	.609	.845	.942	.982	.993
SLA (Baseline)	.561	.606	.833	.931	.982	.995
LGC	.5	.569	.569	.567	.56	.565

Table 20: Accuracy score for `spirals` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	20	40	200	400	1000	2000
CLUSTERING	.77	.773	.816	.823	.828	.826
CLUSTERING (Baseline)	.763	.771	.802	.809	.812	.813
LDA	**.794**	.797	.823	.825	.828	.827
LDA (Baseline)	.786	.79	.815	.819	.821	.823
RF	.764	**.808**	**.831**	**.832**	**.832**	**.832**
RF (Baseline)	.763	.771	.802	.809	.812	.813
SLA	.727	.5	.5	.819	.825	.824
SLA (Baseline)	.637	.616	.592	.565	.538	.541
LGC	.642	.614	.59	.564	.538	.541

Table 21: Accuracy score for `overlapping_planes` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	9	19	95	191	479	958
CLUSTERING	.956	**.998**	**1.0**	**1.0**	**1.0**	**1.0**
CLUSTERING (Baseline)	.957	.984	1.0	1.0	1.0	1.0
LDA	.95	.975	.997	.999	1.0	1.0
LDA (Baseline)	.937	.965	.994	.998	.999	1.0
RF	.933	.967	.997	.999	.999	1.0
RF (Baseline)	.933	.964	.996	.998	.999	1.0
SLA	**.961**	.979	1.0	1.0	1.0	1.0
SLA (Baseline)	.957	.984	1.0	1.0	1.0	1.0
LGC	.484	.484	.484	.484	.484	.484

Table 22: Accuracy score for `lowdense` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	9	18	92	184	460	920
CLUSTERING	.81	.85	.937	.939	.944	.944
CLUSTERING (Baseline)	**.934**	**.971**	**.995**	**.998**	.998	.999
LDA	.542	.517	.508	.553	.49	.543
LDA (Baseline)	.544	.518	.508	.553	.49	.543
RF	.873	.912	.977	.986	.991	.995
RF (Baseline)	.87	.91	.976	.985	.991	.995
SLA	.579	.733	.962	.998	.998	**1.0**
SLA (Baseline)	.572	.716	.969	.998	**1.0**	1.0
LGC	.503	.504	.504	.503	.503	.504

Table 23: Accuracy score for `no_lowdense` dataset

Ratio	0.5%	1%	5%	10%	25%	50%
Lsizes	9	18	92	184	460	920
CLUSTERING	.631	.735	.923	.932	.94	.954
CLUSTERING (Baseline)	.605	.725	**.972**	**.99**	**.997**	**.998**
LDA	.554	.538	.526	.54	.504	.521
LDA (Baseline)	.55	.539	.526	.54	.504	.52
RF	**.71**	**.823**	.957	.977	.99	.993
RF (Baseline)	.704	.822	.954	.975	.989	.994
SLA	.623	.72	.967	.99	.997	.997
SLA (Baseline)	.605	.725	.972	.99	.997	.998
LGC	.505	.505	.505	.505	.505	.505

Table 24: Accuracy score for `vertical_no_lowdense` dataset