Robust Machine Learning

Nan Ye

School of Mathematics and Physics
The University of Queensland

Where Are We Heading to?

How to build good ML models

- Making use of a crowd $\Rightarrow$ Week 7 Ensemble methods

  each of us is a biological prediction model trained on different datasets...
- Using a neural network $\Rightarrow$ Week 8 and 9 Neural networks

  brain-inspired models, some are good for images...
- Making a robust model $\Rightarrow$ Week 10 Robust machine learning

  malicious users, outliers,...
- Asking for explanations $\Rightarrow$ Week 11 Interpretable machine learning

  ...let's ask the machines for explanations...
- Exploiting prior beliefs $\Rightarrow$ Week 12 Bayesian methods

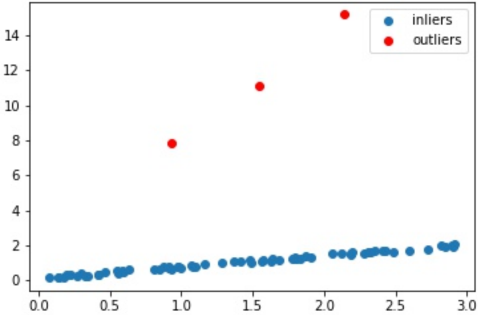
Robust Machine Learning

What is robustness

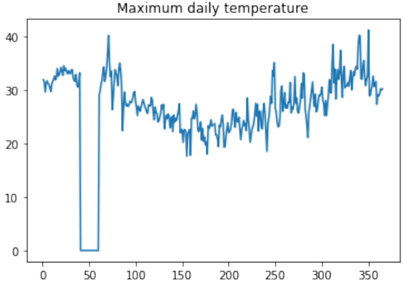
- In theory, we often assume that data is independently drawn from the data generation mechanism that we are interested in.
- In practice, the data that we get is seldom so clean
  - e.g. outliers due to measurement errors, wrong units
  - e.g. maliciously modified data by attackers
- Robust machine learning methods aim to make machine learning work robustly in these undesirable situations.

Outliers

- Outliers are unusual or atypical observations



- Outliers may indicate errors in a (reasonably good) dataset
  - e.g. sensor failures, mistakes in data entry
- Anything wrong with the plot below?

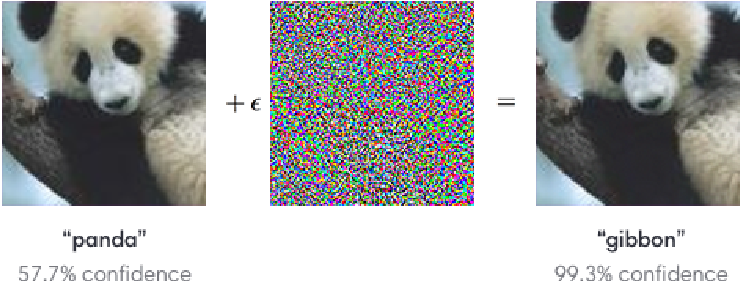


possibly, sensors failed around day 50

- In general, outliers are
  - much less frequent than inliers (i.e. normal observations)
  - differ significantly from the inliers
- "you know it when you see it"
  - there isn't a single precise and agreed-upon definition for outliers
  - different specific definitions are often used in different contexts

Adversarial examples

- Can you see any difference between the two panda images?



- Adversarial examples are imperceptibly different from examples correctly classified by a model, but they are incorrectly classified.
- There are algorithms for generating adversarial examples
- An adversary can use adversarial examples to trick your system.

Goodfellow, Shlens, and Szegedy, Explaining and harnessing adversarial examples, 2015

Robust methods

- We focus on algorithms that are less affected by outliers and adversarial examples in this course.
- Outliers and adversarial examples present very different challenges
  - Outliers are considered as misleading data points and thus best to be removed.
  - Adversarial examples are similar to regular observations, and the algorithms are expected to be able to give correct predictions on them.

Checking Your Understanding

Which of the following statement is correct? (Multiple choice)

(a) All machine learning algorithms can produce good models on datasets with outliers.

(b) Adversarial examples are special types of outliers.

(c) Outliers are examples that are infrequent and differ significantly from normal examples.

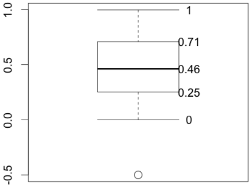
Learning with Outliers

Approaches

- Filter outliers first, then build a model
- Subsampling methods
  - make use of multiple random subsamples to find a robust model
  - we cover Theil-Sen estimators and RANSAC
- Robust loss methods (aka *M*-estimators in statistics)
  - make use of a loss function which is robust against outliers
  - we cover ℓ_1 regression and Huber regression

Outlier Detection

- For one dimensional data, we can use the box-plot to visualize the distribution of the data and check whether there are outliers



- One common rule is to classify points outside $[Q_1 - 1.5IQR, Q_3 + 1.5IQR]$ as outliers
  - Q_1 and Q_3 are the 1st and 3rd quartiles respectively
  - $IQR = Q_3 - Q_1$ is the interquartile range

- In higher dimensional spaces, we need to use more sophisticated methods for detecting the outliers
- E.g. isolation forest, one-class SVMs (not covered)
- In general, if outliers in a dataset are errors, filtering outliers first before training a model leads to a better model.

Subsampling Methods

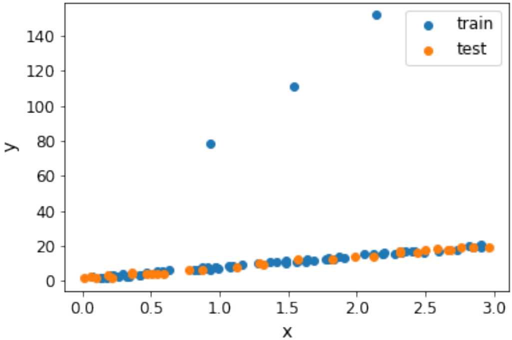
- Recall: outliers are rare and very different from inliers.
  - $\Rightarrow$ a small random subset may not include an outlier, or includes just a few outliers.
- A single small random subset, though possibly free from outliers, usually doesn't contain all information from inliers from the entire dataset.
- Subsampling methods consider multiple small random subsets, and aggregate results obtained using them in some way.

Theil-Sen Estimators

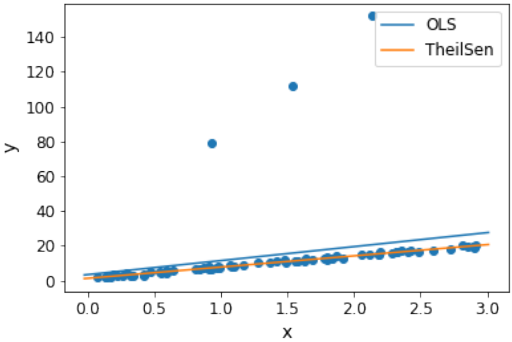
Univariate problems

- Consider a training set $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbf{R} \times \mathbf{R}$, possibly with some outliers.
- Theil-Sen estimator works as follows
  - Randomly sample N pairs $(x_i, y_i), (x_j, y_j)$ with $x_i \neq x_j$, and for each pair, compute the slope $\frac{y_i - y_j}{x_i - x_j}$ for the line passing through them.
  - Compute the median slope m of all the N slopes.
  - Compute the median bias b of all $y_1 - mx_1, \ldots, y_n - mx_n$.
  - The fitted line is $y = mx + b$.

A small problem



- Training set: 70 points with 3 outliers (y is 10 times larger due to wrong units)
- Test set: 30 points.



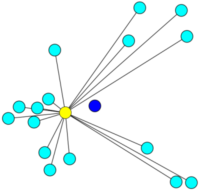
- OLS model is pulled away from the inliers by the outliers.
- Theil-Sen estimator appears unaffected by the outliers.

Multivariate problems

- Consider a training set $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbf{R}^d \times \mathbf{R}$, possibly with some outliers.
- Theil-Sen estimator works as follows
  - Randomly draw N subsamples of $d + 1$ *different* examples
  - For each subsample, find a linear least squares solution
  - Compute the geometric median of all the N linear least squares solutions as the fitted model.

Subproblems in Theil-Sen

- The geometric median of several points is the point with smallest total distance to them.



yellow: geometric median

blue: centroid

https://en.wikipedia.org/wiki/Geometric_median

- When a linear least squares problem doesn't have a unique solution, typically the one with minimum norm is chosen.
- There are algorithms for solving both problems above (not covered in this course).

Variants

- Instead of using random subsamples, we can use all subsamples of size $d+1$, provided that $\binom{n}{d+1}$ is not too large.
- Instead of using subsamples of size $d+1$, we can use subsamples of larger size.

RANSAC
(RANdom SAmple Consensus)

- Theil-Sen
  - fits models on many subsamples
  - makes no effort to ensure that these model are good
  - aggregate them to form a good model
- RANSAC takes into account that each model fit on a subsample is not necessarily good.
  - each such model is used as an outlier detector,
  - a candidate inlier model is trained using all detected inliers
  - the best candidate inlier model is chosen

RANSAC

for $i = 1$ to N **do**

randomly draw a subsample S of size n_0

fit a model M on S

compute the predictions of M on all n examples

classify examples with error less than a threshold t as inliers

fit a candidate inlier model M' using the inliers

compute M''s score s on the inliers

Choose the candidate inlier model with highest score

RANSAC

for $i = 1$ to N **do**
  randomly draw a subsample S of size n_0
  fit a model M on S
  compute the predictions of M on all n examples
  classify examples with error less than a threshold t as inliers
  fit a candidate inlier model M' using the inliers
  compute M''s score s on the inliers
Choose the candidate inlier model with highest score

Hyperparameter: subsample size n_0

- Subsample size n_0 is often chosen to be the minimum number of data points required for fitting a basis model, but can be a larger number.

RANSAC

for $i = 1$ to N **do**

randomly draw a subsample S of size n_0

fit a model M on S

compute the predictions of M on all n examples

classify examples with error less than a threshold t as inliers

fit a candidate inlier model M' using the inliers

compute M''s score s on the inliers

Choose the candidate inlier model with highest score

Hyperparameter: basis model

- RANSAC is generic and can be applied to any basis model, not just linear regression.

RANSAC

for $i = 1$ to N **do**

randomly draw a subsample S of size n_0

fit a model M on S

compute the predictions of M on all n examples

classify examples with error less than a threshold t as inliers

fit a candidate inlier model M' using the inliers

compute M''s score s on the inliers

Choose the candidate inlier model with highest score

Hyperparameter: error measurement

- An error function $L(y, \hat{y})$ is used to measure the prediction error (e.g. absolute error, quadratic error).

RANSAC

for $i = 1$ to N **do**

randomly draw a subsample S of size n_0

fit a model M on S

compute the predictions of M on all n examples

classify examples with error less than a threshold t as inliers

fit a candidate inlier model M' using the inliers

compute M''s score s on the inliers

Choose the candidate inlier model with highest score

Hyperparameter: error threshold

- The threshold t can be chosen as median of $L(y_1, y_{1/2}), \ldots, L(y_n, y_{1/2})$, where $y_{1/2}$ is the median of y_i's.

RANSAC

for $i = 1$ to N **do**

randomly draw a subsample S of size n_0

fit a model M on S

compute the predictions of M on all n examples

classify examples with error less than a threshold t as inliers

fit a candidate inlier model M' using the inliers

compute M''s score s on the inliers

Choose the candidate inlier model with highest score

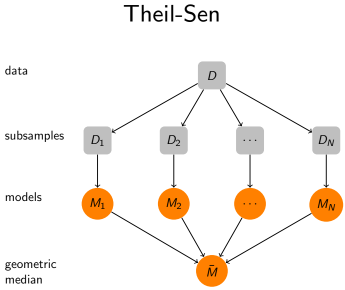
Hyperparameter: score

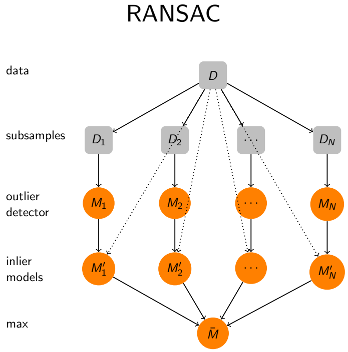
- R^2 is often used.

Additional details

- Typically, we also check the number of inliers detected by each M
  - if the number of inliers is not large enough, or if fewer inliers are detected than a previous outlier detector, we move on to the next subsample (without training a candidate inlier model)
- The number of trials N may be adjusted by estimating the number of trials needed to get at least one outlier-free subsample.
- Sometimes we're lucky and get a good sample early — we can terminate early once the number of inliers is sufficiently large.

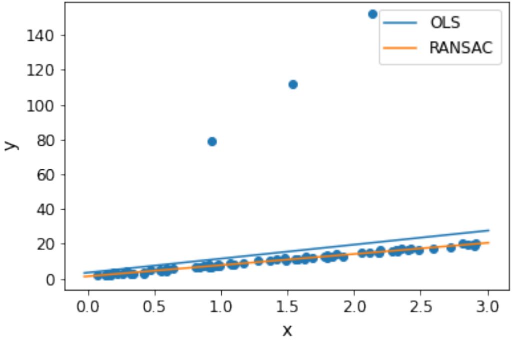
Theil-Sen vs RANSAC





Theil-Sen and RANSAC are implemented in `sklearn.linear_model` as `TheilSenRegressor` and `RANSACRegressor` respectively.

The small problem again



- Just like the Theil-Sen estimator, RANSAC appears unaffected by the outliers on this dataset.

M-estimators

- In regression, we often choose a model $f(\mathbf{x})$ to minimize its MSE

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2,$$

  where $\hat{y}_i = f(\mathbf{x}_i)$.
- If $(\mathbf{x}, y)$ is an outlier, then for an inlier model, the residual

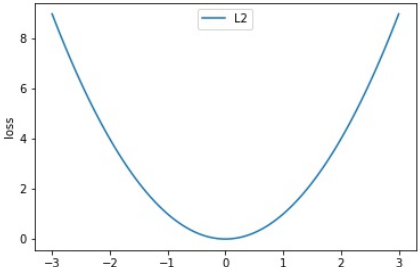
$$r = y - \hat{y}$$

  is likely large, and similarly for the quadratic loss

$$L_2(r) = L_2(y, \hat{y}) = (\hat{y} - y)^2.$$

Quadratic loss takes outliers (too) seriously

- While we want to ignore outliers, quadratic loss assigns much larger penalty to them than the inliers, because the penalty grows rapidly when the residual becomes larger.



M-estimators

- Instead of finding a model $f_{\mathbf{w}}(\mathbf{x})$ to minimize

$$\frac{1}{n}\sum_{i} L_2(y_i, f_{\mathbf{w}}(\mathbf{x}_i)),$$

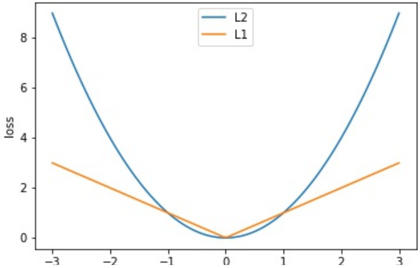
  M-estimators finds a model $f_{\mathbf{w}}(\mathbf{x})$ to minimize

$$\frac{1}{n}\sum_{i} L(y_i, f_{\mathbf{w}}(\mathbf{x}_i))$$

  for some other loss function L
- To build a model robust against outliers, L is chosen to apply less aggressive penalty to outliers than L_2.
- Many possible such L's, leading to many different robust models.

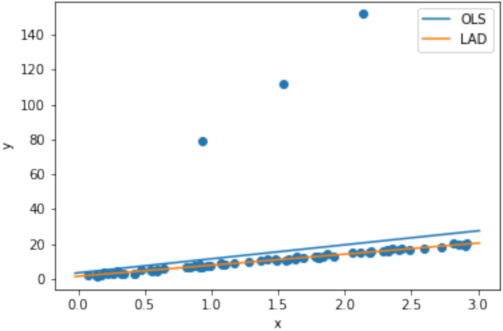
Least Absolute Deviations (LAD)

- LAD regression minimizes the ℓ_1 loss $L_1(r) = |r|$.



- As compared to the L_2 loss, L_1 applies slightly larger penalties to small errors, but much smaller penalties to large errors.

LAD regression on the small problem



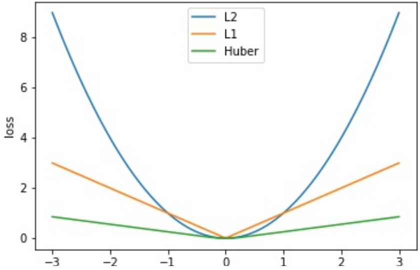
- Just like Theil-Sen and RANSAC, LAD regression appears unaffected by the outliers on this dataset.

Huber Regression

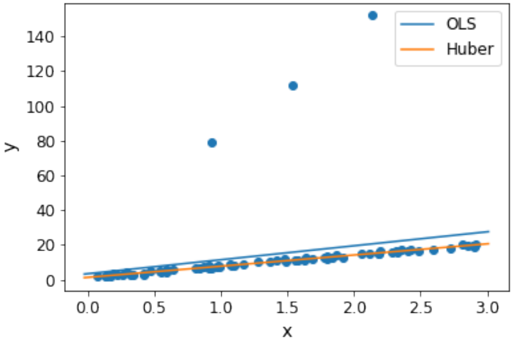
- Huber regression minimizes the Huber loss

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2, & |r| \leq \delta, \\ \delta\left(|r| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

  That is, it is quadratic for small r, then becomes linear.
- For $\delta \leq 1$, Huber loss is always smaller than L_1



Huber regression on the small problem



- Huber regression has no problem of passing the test of the small problem too.

Checking Your Understanding

Which of the following statement is correct? (Multiple choice)

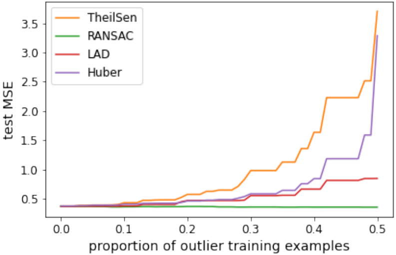
(a) One approach of learning an inlier model is to first filter out the outliers, then apply a regular learning algorithm to learn a model.

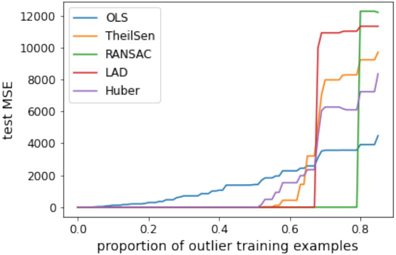
(b) LAD is a subsampling method for dealing with outliers.

(c) RANSAC is an M-estimator.

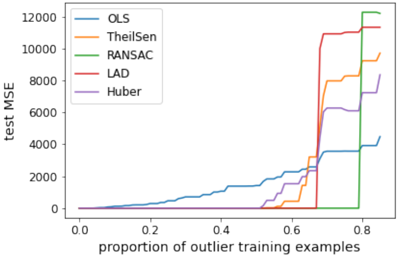
Comparing Robustness

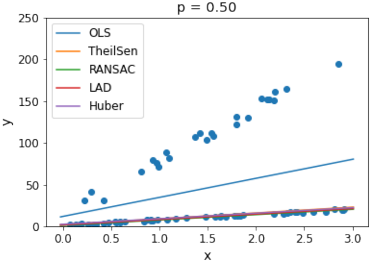
- While all the algorithms appear to learn the same model on the same problem (the small problem), there are actually minor differences.
- When we increase the proportion p of outliers, the differences become larger.



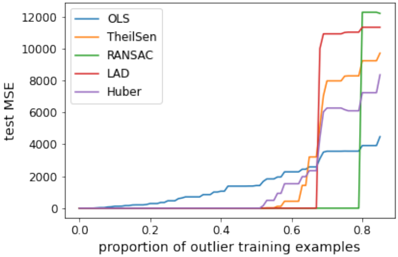


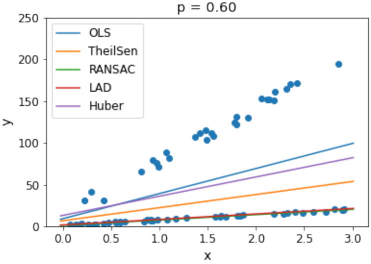
- When $p < 0.5$, all the robust methods are barely affected by the outliers
- Once we have more than 50% outliers, the outliers are not really outliers.
  - However, it takes the methods some time to figure this out (when p is much larger than 0.5).
  - These indicate that such methods can be seriously affected by outliers in some other datasets.



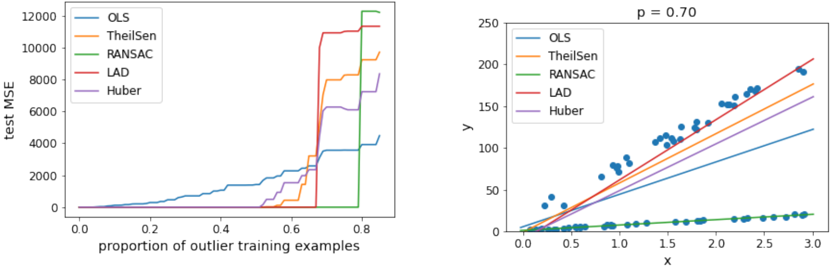


- When $p = 0.5$, OLS model is roughly at the middle of the inliers and outliers.

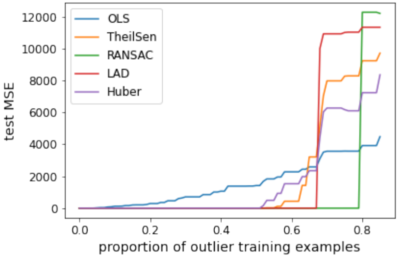


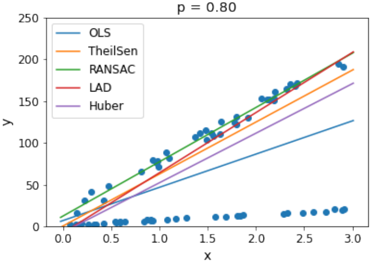


- When $p = 0.6$ Huber and Theil-Sen are moving up, while RANSAC and LAD are not making much move.



- When $p = 0.7$, LAD has found out that outliers are no longer outliers, Theil-Sen and Huber are close, but RANSAC is still not making a move.





- When $p = 0.8$, RANSAC has found out that outliers are no longer outliers, and it seems to do a better job than others.

- Data using larger units have smaller y values and can be considered to be simpler than data using smaller units.
- Thus all the robust methods seem to have a preference for model fitted on simpler data, and when most y values are recorded using the smaller unit, they fail to promptly switch to use the smaller unit as the norm.

Case Study: Boston House Prices

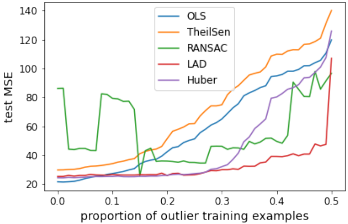
- 506 instances, random 354/152 train-test split
- Predict median house price in a town using 13 numeric features,
- Available in `sklearn.datasets`.
- Variables in the dataset

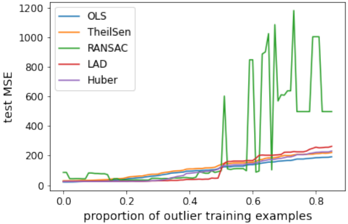
```
- CRIM     per capita crime rate by town
- ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS    proportion of non-retail business acres per town
- CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX      nitric oxides concentration (parts per 10 million)
- RM       average number of rooms per dwelling
- AGE      proportion of owner-occupied units built prior to 1940
- DIS      weighted distances to five Boston employment centres
- RAD      index of accessibility to radial highways
- TAX      full-value property-tax rate per $10,000
- PTRATIO  pupil-teacher ratio by town
- B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT    % lower status of the population
- MEDV     Median value of owner-occupied homes in $1000's
```

Outlier creation

- Wrong house price unit: outlier has housing price recorded in millions, instead of \$1,000
- Wrong nitric oxide unit: outlier has nitric oxides concentration recorded in parts per million instead of parts per 10 million
- Wrong house price and nitric oxide unit: outlier has wrong units for both house price and nitric oxides concentration

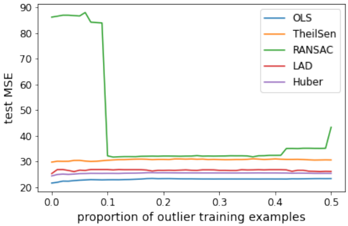
Wrong house price unit

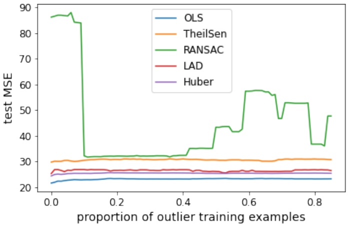




- Both Theil-Sen and RANSAC are not quite robust when some target value (price) are scaled down by 1000 times.
    - RANSAC appears to be highly unstable. This can be alleviated by using a larger N value.
- LAD and Huber are more robust than OLS.

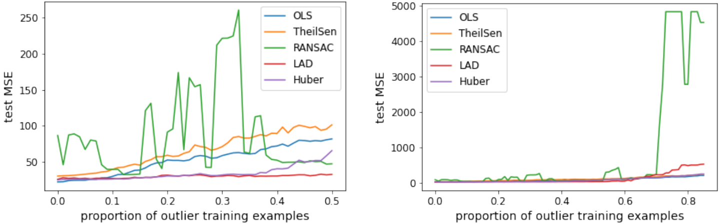
Wrong nitric oxide unit





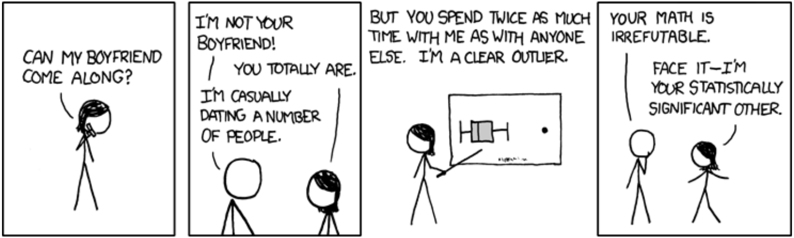
- All robust methods do not work well when some nitric oxides concentrations are scaled down by 10 times, but TheilSen, LAD and Huber are not much worse than OLS.

Wrong house price and nitric oxide units



- With both types of corruptions, LAD and Huber are much better.

Outliers $\neq$ Liars, out



https://xkcd.com/539/

Every swan is white, and then you see this in Australia...

- In many domains, outliers are important, and are what we are interested in.
- For example, in credit card transactions, we are interested in detecting frauds, but they are often outliers in some sense $\Rightarrow$ using an outlier detection algorithm to filter out the outliers removes what we are interested in.
- Algorithms designed to be robust against outliers shouldn't be used in such domains.

Checking Your Understanding

Which of the following statement is correct? (Multiple choice)

(a) RANSAC always produces a better model than OLS when there are outliers.

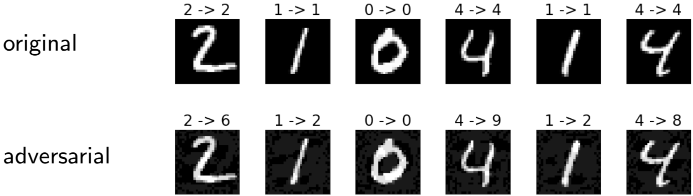
(b) Huber regression always produces a better model than OLS when there are outliers.

(c) When we increase the proportion of outliers, some robust methods fail earlier than others.

Adversarial Examples!

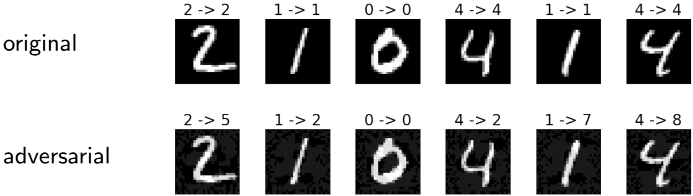
- Recall: adversarial examples appear indistinguishable to ‘easy’ examples, but they are incorrectly classified.
- Adversarial examples are not just something applicable to complex neural nets.
- Many machine learning models have difficulty with adversarial examples.

Adversarial examples for logistic regression



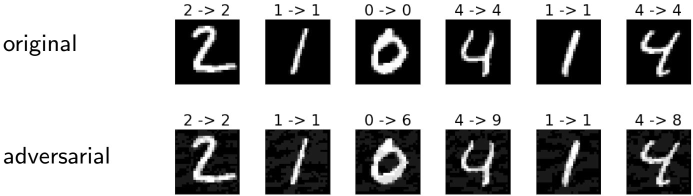
- Impercetible noise reduces accuracy from $6/6$ to $1/6$.
- $2 \to 6$ and $1 \to 2$ are quite unexpected.

Adversarial examples for SVM



- Impercetible noise reduces accuracy from 6/6 to 1/6.
- The adversarial images for SVM are different from those for logisic regression (hard/impossible to see the differences though).
- $2 \to 5$, $1 \to 2$, $4 \to 2$ are quite unexpected.

Adversarial examples for LeNet



- Impercetible noise reduces accuracy from 6/6 to 3/6.
- LeNet's errors seem somewhat more reasonable (the kind of errors that are more frequently made by humans).
- While both logistic regression and SVM have no problem with getting 0 correct with noise, LeNet misclassified the perturbed 0.

Adversarial Learning

- Defending adversarial examples is hard: many attempts, none always works.
- Improving robustness against adversarial examples
  - Data augmentation approach
    - Generate many adversarial examples, add them to the training set
    - Train your model on the new training set
  - Adversarially robust objective
    - Some attacks have a function A that produces an adversarial example $\mathbf{x}' = A(\mathbf{x}, y)$ for a given example $(\mathbf{x}, y)$.
    - We can define a robust objective by adding to the original training objective an extra penalty to wrong predictions on $(\mathbf{x}', y)$, for all training example $(\mathbf{x}, y)$.

What You Need to Know

- Robust machine learning methods try to produce models that work well with 'hard' data.
  - two types of hard data: outliers, adversarial examples
- Robust methods for outliers
  - Filtering before learning
  - Subsampling methods: Theil-Sen, RANSAC
  - M-estimators: LAD, Huber regression
- Robust methods for adversarial examples
  - Data augmentation approach
  - Adversarially robust learning objective