RANSAC

RANSAC is the abbreviation of Random Sample Consensus. It is a general algorithm that can be used with other parameter estimation methods in order to obtain robust models with a certain probability when the noise in the data doesn't obey the general noise assumption.

for a better world without outliers...

Contents

Introduction

General parameter estimation techniques base on an assumption that the noise in the training data is Gaussian noise with zero mean. So, it would be smoothed out after parameter optimization that averages over all of the data. However, this assumption doesn't hold when there exists gross errors that may occur with extreme values of noise or with wrong measurements during sampling. This kind of errors actually don't fit the model and should be rejected during training in order to obtain better fits for the model.

Overview

Motivation

RANSAC is aimed to determine function parameters when there exist gross-erroneous samples that can mislead the parameter estimation.

Assumptions

RANSAC assumes that the training data consists of *inliers* that can be explained with the model and *outliers* that are gross-erroneous samples which don't fit the model at all. So using outliers when training the model would increase our final prediction error, as they contain almost no information about the model. Furthermore, RANSAC trains the parametric model only with inliers while ignoring outliers. However, seperating the data as inliers and outliers would be a strong assumption, but also it is the main difference of RANSAC from the other methods. In addition, even if this assumption doesn't hold for a data set, RANSAC would make no harm to the parameter estimation, as in this condition it would consider the whole data set as inliers and train model with them.

In order to reject outliers during training, RANSAC uses a small set of samples to train a model rather than using all of the data and then enlarges set with other appropriate samples. By using a small set, it automatically assumes that the model can be estimated with a small number of inliers. However, it is a soft assumption and it holds for most of the cases; e.g. in order to capture a linear function, two data samples are sufficient.

Method

RANSAC uniformly at random selects a subset of data samples and uses it to estimate model parameters. Then it determines the samples that are within an error tolerance of the generated model. These samples are considered as agreed with the generated model and called as *consensus* set of the chosen data samples. Here, the data samples in the consensus as behaved as inliers and the rest as outliers by RANSAC. If the count of the samples in the consensus is high enough, it trains the final model of the consensus with using them. It repeats this process for a number of iterations and returns the model which has the smallest average error among the

generated models.

As a randomized algorithm, RANSAC doesn't guarantee to find the optimal parametric model with respect to the inliers. But, the probability of reaching the optimal solution can be kept over a lower bound with assigning suitable values to algorithm parameters.

Algorithm

Prerequisites

Actually, RANSAC is not a complete and a self-working algorithm. In fact, it is a complementary algorithm that uses a model and its distance function to estimate the model parameters robustly while there exist outliers. It simply applies an outlier seperation over the data set and trains the model only with inliers according to an optimization method and the distance function of the model. Hence, before using RANSAC, a model, a distance function and an optimization algorithm has to be determined. However, it also makes RANSAC completely modular and allows it to work with any estimation model, any distance function and any optimization method.

Flow

In order to understand better what RANSAC does, the flow of the algorithm is described generically as follows:

```
Given:
    data - a set of observed data points
    model - a model that can be fitted to data points
    n - the minimum number of data values required to fit the model
    k - the maximum number of iterations allowed in the algorithm
    t - a threshold value for determining when a data point fits a model
    d - the number of close data values required to assert that a model fits well to data
Return:
    bestfit - model parameters which best fit the data (or nil if no good model is found)
iterations = 0
bestfit = nil
besterr = something really large
while iterations < k
    maybeinliers = n randomly selected values from data
    maybemodel = model parameters fitted to maybeinliers
    alsoinliers = empty set
    for every point in data not in maybeinliers
        if point fits maybemodel with an error smaller than t
             add point to alsoinliers
    if the number of elements in alsoinliers is > d
        % this implies that we may have found a good model
        % now test how good it is
        bettermodel = model parameters fitted to all points in maybeinliers and alsoinlie
        thiserr = a measure of how well model fits these points
        if thiserr < besterr
            bestfit = bettermodel
            besterr = thiserr
    increment iterations
return bestfit
* The algorithm flow is referenced from Wikipedia (http://en.wikipedia.org/wiki/RANSAC)
```

Parameters

Like described in the beginning of the algorithm flow, RANSAC needs some predefined parameters to run - size of sample subset(n),error tolerance threshold(t),minimum consensus threshold(d) and number of iterations(k). In addition, it is important to estimate the proportion of inliers(w) in the data set, in order to calculate some of these parameters.

As RANSAC is a randomized algorithm, it is essential to truely estimate the parameters, in order to increase the probability of finding the optimal model, while still keeping the computational advantages of a randomized algorithm against an exhaustive deterministic algorithm. Here are some heuristics to calculate these parameters.

Proportion of Inliers

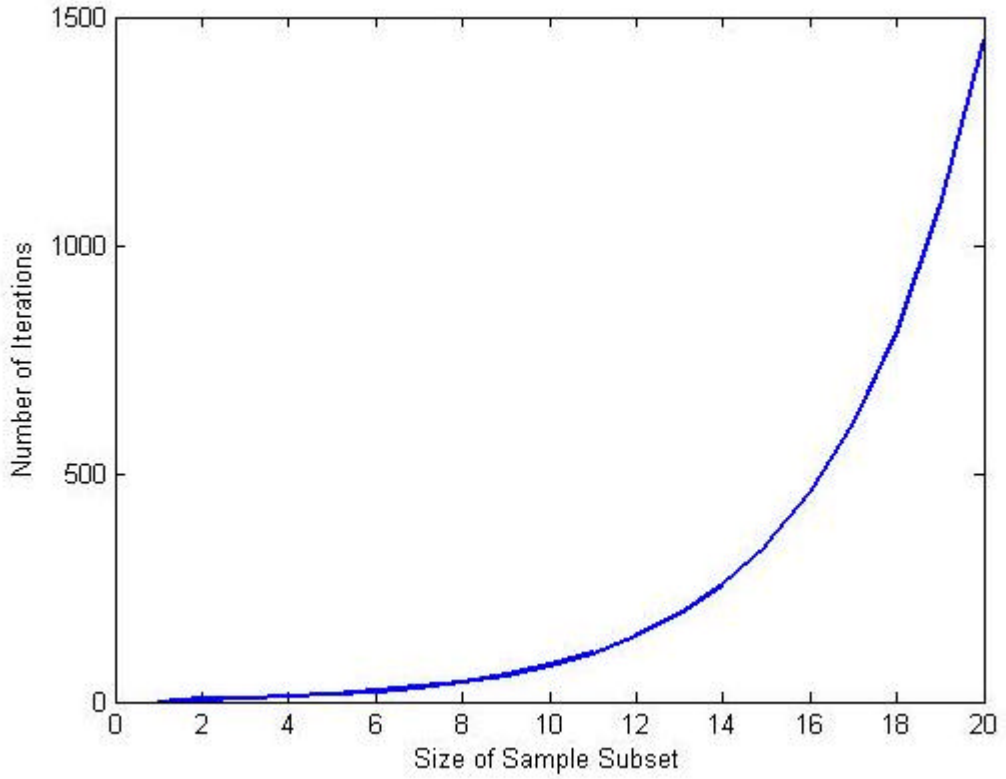
Although it is not a direct parameter in the algorithm, proportion of inliers are used in the calculations of algorithm parameters and even it can effect the algorithm complexity implicitly. Hence, it is beneficial to have some information about the outlier rate in the data set before running RANSAC.

Size of Sample Subset

Size of Sample Subset is the number of samples that are randomly chosen by RANSAC to inital model at each iteration. It is directly related with the model that is intended to fit the data set. RANSAC uses the minimum number of samples needed to define the model as the sample subset size. e.g. in order to fit a linear model it chooses 2 data samples or to fit a circle-shaped model it selects 3 data samples as 3 points would be sufficient to define a circle.

$$n \equiv \text{minimum number of samples to define model}$$

One could think that using more data samples then the minimal subset would be advantageous, as a better and a more accurate estimate of the model can be obtained initially. However, having more samples in the sample subset would increase the search space for the subset selection. So, in order to keep the probability of finding the optimal model at the same level, we would need to try more sample subset. Hence, an increase in the number of iterations is required, that mostly causes an increase in the computational complexity which outweighs the advantages of having a larger sample subset. Here is the relationship of sample subset size and number of iterations which directly affects the complexity.



Error Tolerance Threshold

Error Tolerance Threshold is used by RANSAC in order to determine if a data sample agrees with a model or not. The samples under this threshold would then form that consensus for that model, which would be the inliers of the data set if the correct model is found. Hence, it should be chosen according to the gaussian error in the inliers.

$$t \approx \text{gaussian error in inliers}$$

Minimum Consensus Threshold

Minimum Consensus Threshold is the minimum number of samples that would be accepted as a valid consensus to generate a final model for that iteration. As RANSAC tries to capture the inliers with constituting consensuses, the number of samples in a valid consensus is directly related with the number of inliers in the data set. Hence, RANSAC uses a threshold value that is equal to or a bit smaller - tolerated - than the number of inliers in order to accept consensus as valid.

If the total number of the samples in the data set is | Data Set | ,

$$d \approx w \cdot |\text{Data Set}|$$

Number of Iterations

Exhaustive deterministic algorithms would try every possible sample subset in order to find the best one, but actually it is not only computationally infeasible, but also unnecessary. Hence, instead of a deterministic way, RANSAC chooses sample subset randomly. But, it is also important to determine the number of these random choices in order to obtain a high probability that RANSAC would choose a sample subset which includes no outliers.

The expected number of iterations to have a successful run with a certain probability can be calculated as follows:

Probability of choosing an inlier:

$$P(\text{inlier}) \equiv w$$

Probability of choosing a sample subset with no outliers:

$$P(\text{subset with no outlier}) \equiv w^n$$

Probability of choosing a sample subset with outliers:

$$P(\text{subset with outlier(s)}) \equiv 1 - w^n$$

Probability of choosing a subset with outliers in all k iterations:

$$P(\text{k subsets with outlier(s)}) \equiv (1 - w^n)^k$$

Probability of an unsuccesful run:

$$P(\text{fail}) \equiv (1 - w^n)^k$$

Probability of a succesful run:

$$P(\text{success}) \equiv 1 - (1 - w^n)^k$$

Expected number of iterations:

$$\Rightarrow k = \frac{log(1 - P(\text{success}))}{log(1 - w^n)}$$

Hence, with determining a appropriate probability *P*(success) according to the desired realiability, the number of iterations can be estimated.

E.g. here are some calculated number of iterations for RANSAC on a line-fitting problem:

For $|\text{Data Set}| \equiv 12$, $n \equiv 2$ and $P(\text{success}) \equiv 0.99$

$w = 0.95 \Rightarrow k = 2$

$w = 0.75 \Rightarrow k = 6$

$w = 0.6 \Rightarrow k = 11$

$w = 0.5 \Rightarrow k = 17$

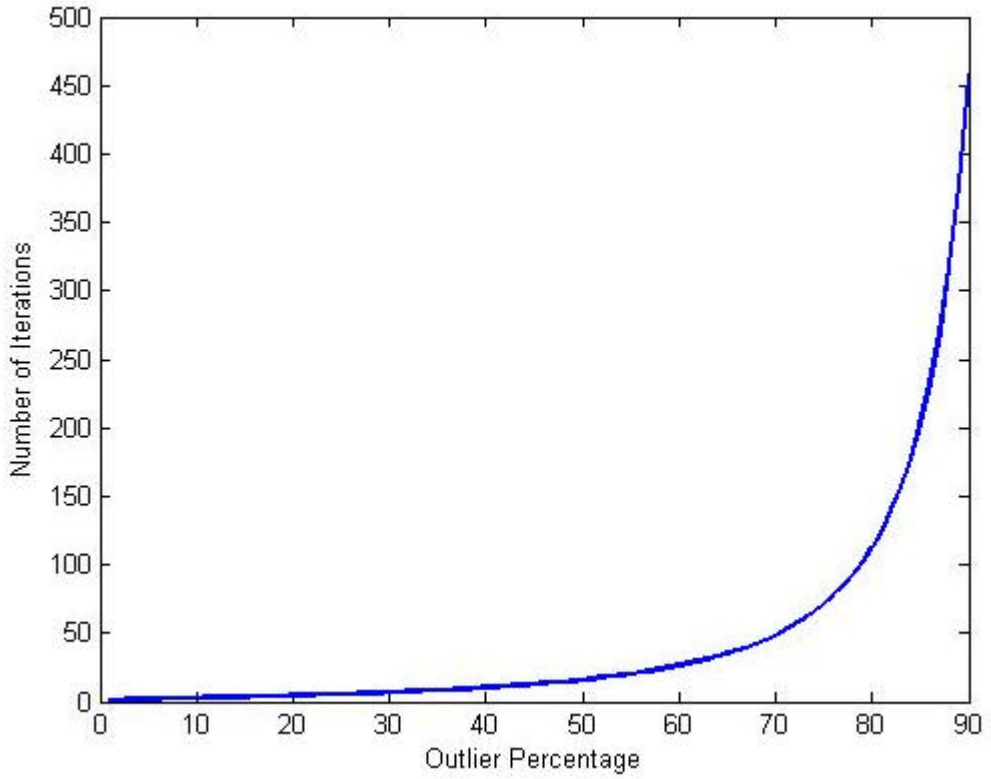
If we consider a deterministic algorithm that tries every possible sample pairs as a subset, for the same setting:

$$k_{det} = \binom{|DataSet|}{2} = \frac{|\text{Data Set}| \cdot (|\text{Data Set}| - 1)}{2} = 66$$

The computational complexity of RANSAC even in a highly outliered setting and a deterministic algorithm can be seen from the number of iterations, while RANSAC still keeps its reliability as having a failure chance of only $P(\text{fail}) = 0.01$.

Furthermore, number of iteration can be dramatically increased as the outlier rate in the data set increases.

For $n \equiv 2$ and $P(\text{success}) \equiv 0.99$



Extensions

Even RANSAC would run faster than a deterministic algorithm as it doesn't consider all possible sample subsets, its performance can still be improved with some heuristics.

Adaptive Parameter Calculation

The main idea of Adaptive Parameter Calculation is calculating the proportion of inliers(w) on run time and modifying the other parameters according to the new value of w. It can be used not only for complexity reduction but also when there is no information about w for the data set.

Adaptive Parameter Calculation starts with a worst case value of w and reassigns the better value when it founds a consensus set whose proportion of data samples is higher than current w. And then it updates number of iterations(k)and minimum consensus threshold(d) with the new value of w.

E.g. The initial value of w is chosen as 0, so k is calculated as ∞ and d is calculated as 0. Lets consider a situtation that a consensus which has the half of the data samples is found. As its proportion is better than the current estimation of w, w is updated as 0.5 and recalculate the values of k and d.With having a higher value for w, k would definitely decrease, so the algorithm will end earlier without losing its reliability. In addition, d is updated and would now

have a higher value, which allow less consensus sets to be accepted in order to construct a model. Hence, it reduces the complexity implicitly.

Here is a basic flow of Adaptive Parameter Calculation of RANSAC:

```
k = infinity , i = 0
while k > i
    RANSAC()
    v = proportion of inliers found
    if v > w
        w = v
        update k and d
    increment i by 1
```

Randomized RANSAC

At each step of iteration of RANSAC, a model is proposed and applied over all the data samples in order to determine if they agree or not. However, this process can be a high complexity load especially for huge data sets and it can be optimized with a randomized heuristic.

Randomized RANSAC tests all the generated models first with a small number of random data samples.If the model does not achieve an enough support for those random samples, it can be assumed that with high probability it is not a good estimate. Hence, test over all the data samples is skipped and the model is dropped. Otherwise, then the model is tested over the whole data set.

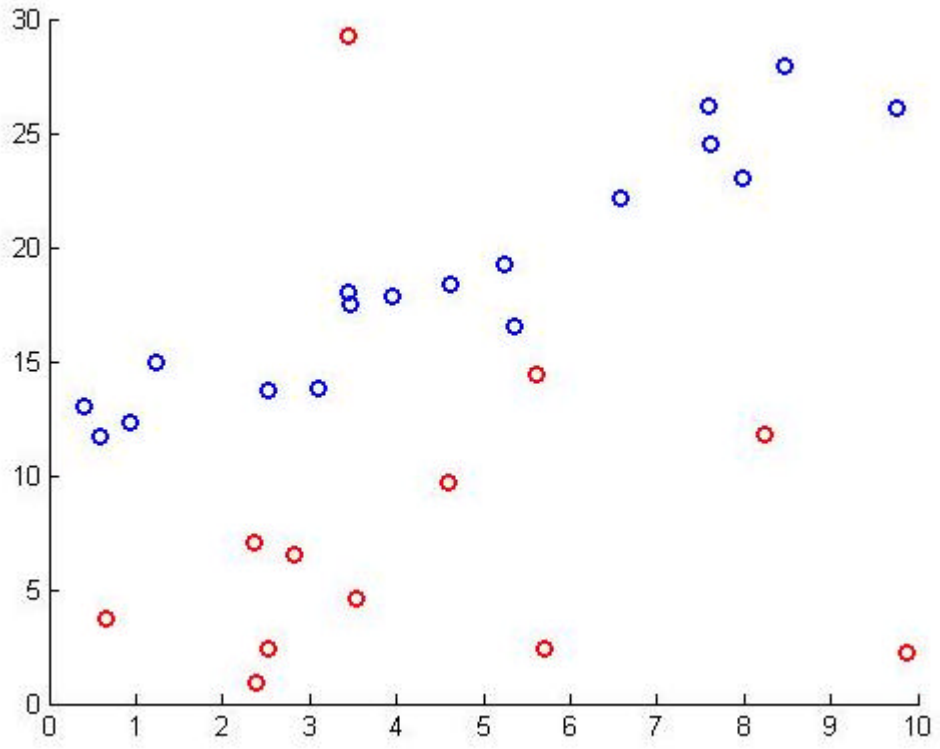
Spatial Sampling

As a general observation, outliers tend to lie seperately while inliers tend to locate closely. So, instead of uniform sampling, a different sampling can be used that takes into account the spatial relationship of samples. Hence, the first initial sample is selected uniformly and then the rest samples are selected with a probability that is inversely proportion to their distance from the initial one. This heuristic can effectively decrease the expected number of iterations required to find a good model.

Experiment : Linear Function Estimation

Linear Data with Outliers

Here is the data set that is used during the experiment:

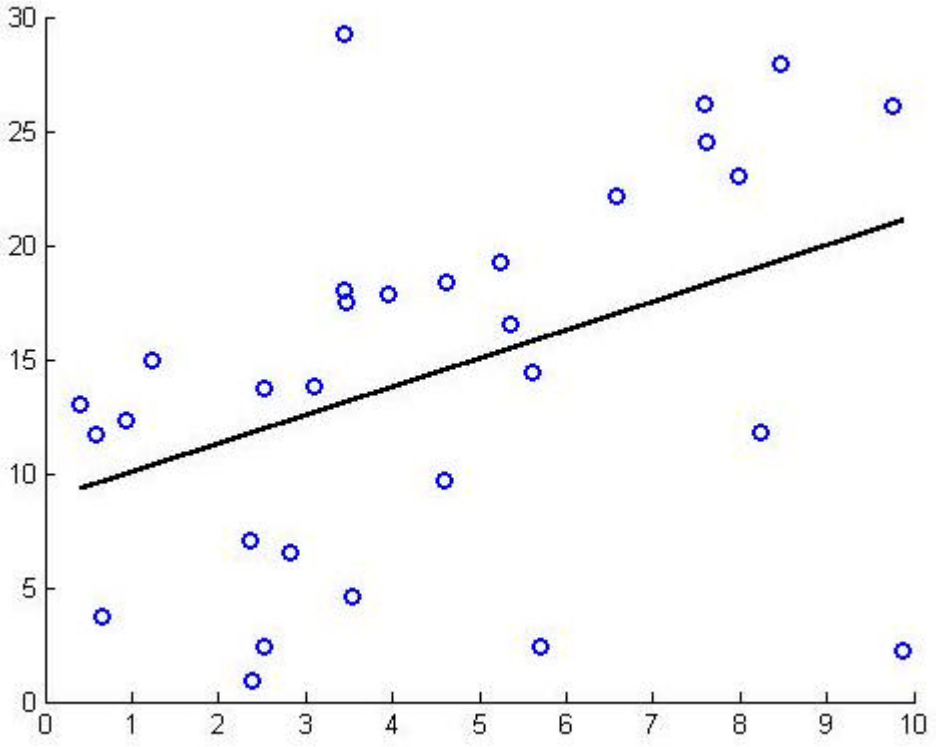


30 data samples are generated with y = 2x + 5 and w = 0.6 with gross errors to cause outliers and with white gaussian noise over the inliers. Outliers are colored with red while the inliers are blue.

This could be considered as a hard setting as the percentage of inliers in the data set is only 60%.

Linear Regression

Here is the result of the linear regression method without RANSAC. As it doesn't consider gross-erroneous samples, the generated fit is highly affected and misleaded by the outliers.

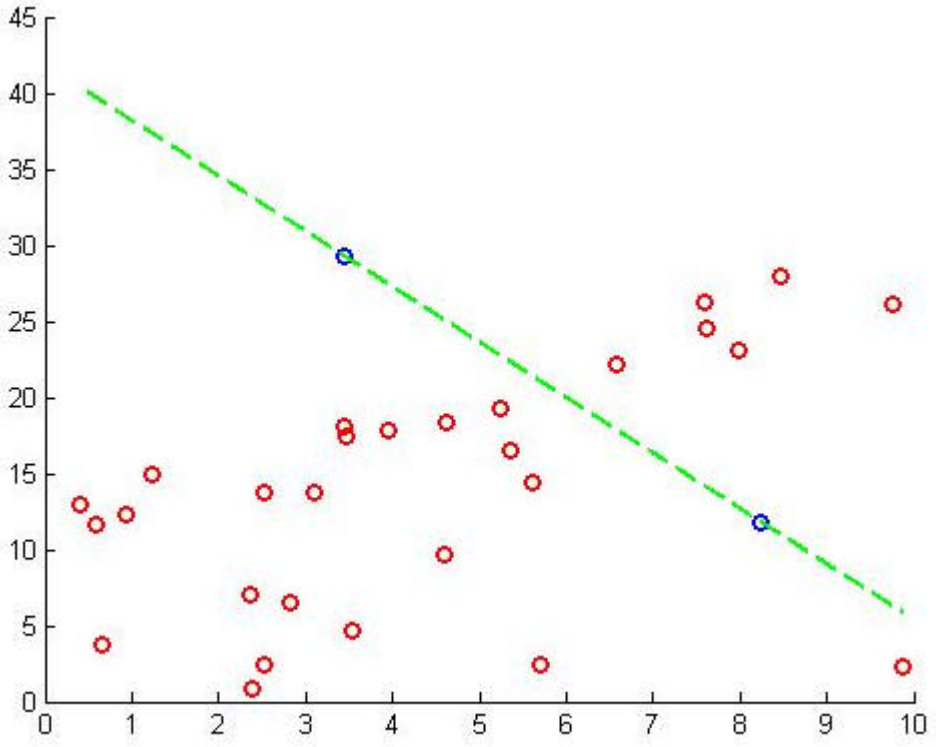


Sample RANSAC Iterations

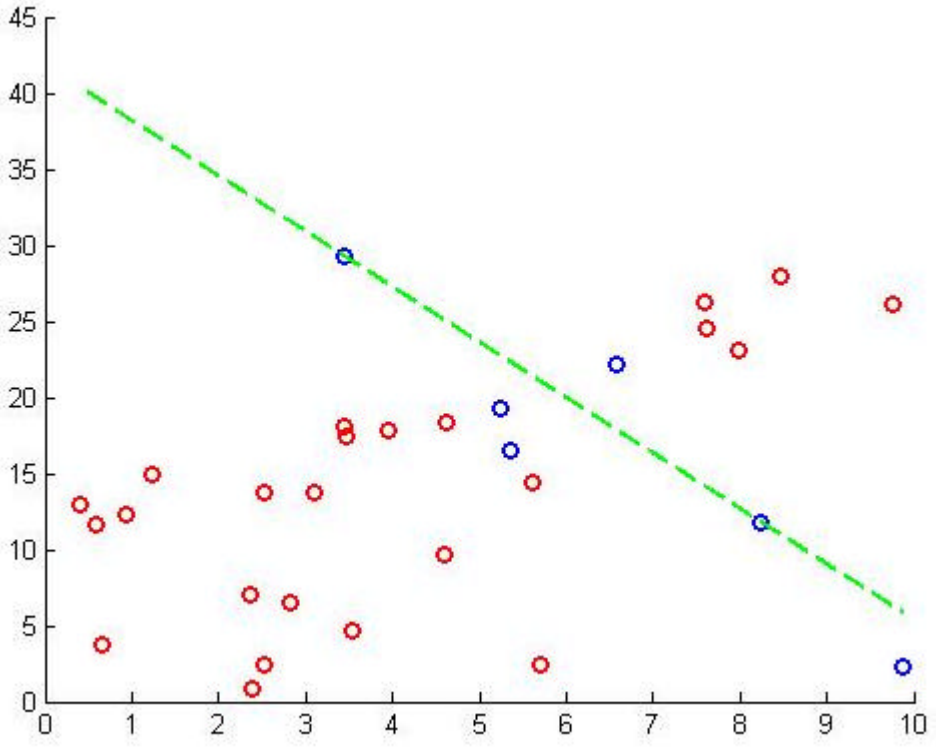
Here are some plots over the data set from the iterations of RANSAC.

Iteration 1

In this iteration, the random sample subset consists of two outliers.

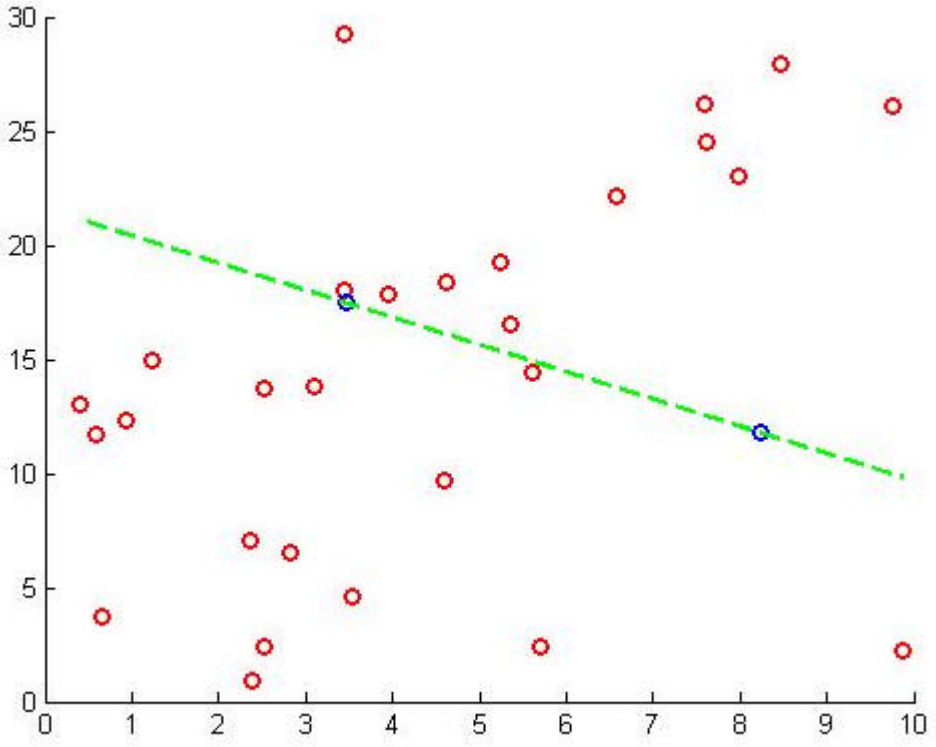


As the model is generated by outlier, it is definitely not a good estimate for the line. Hence it wouldn't pass the minimum consensus threshold as desired.

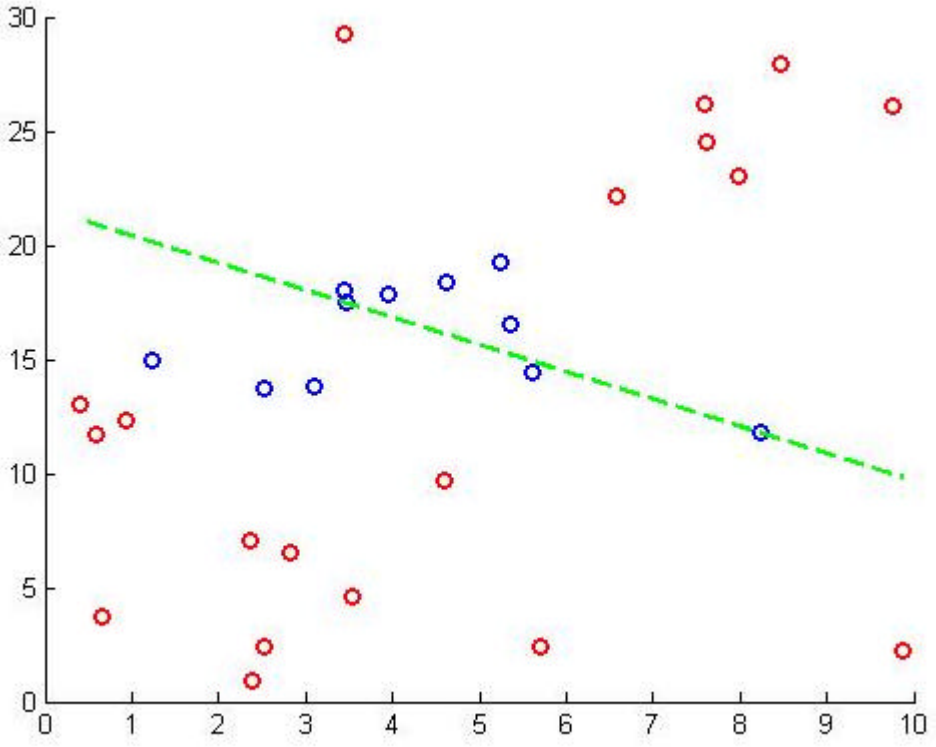


Iteration 2

Here, there is an inlier in the sample subset but still the second random choice is an outlier.

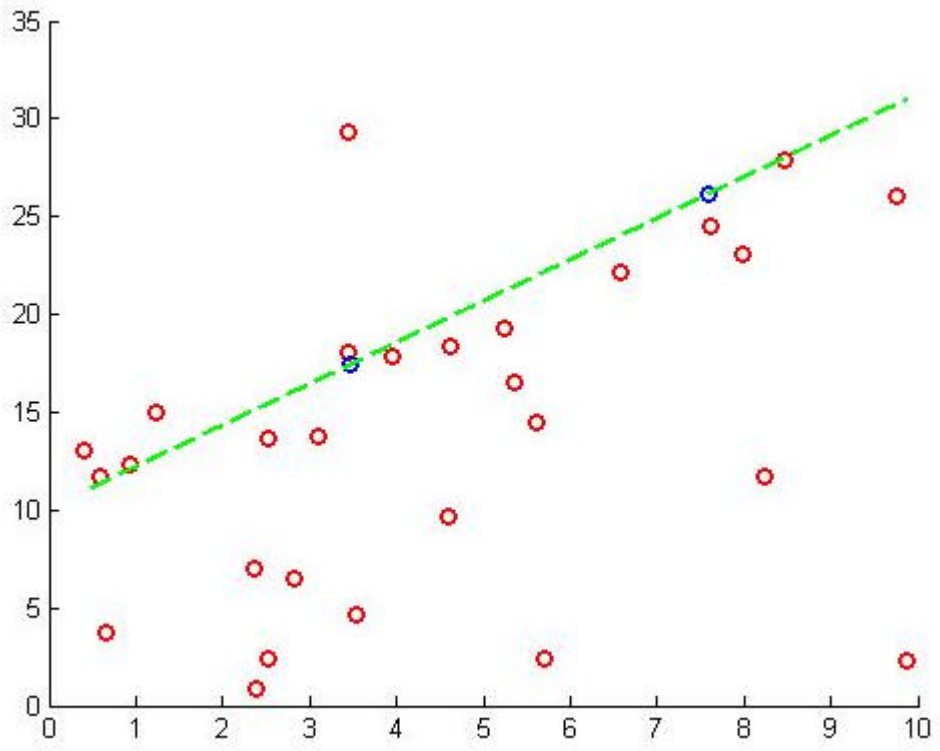


Due to the existence of an outlier in the sample subset, the generated model doesn't have enough support, so it wouldn't pass the minimum consensus threshold neither.

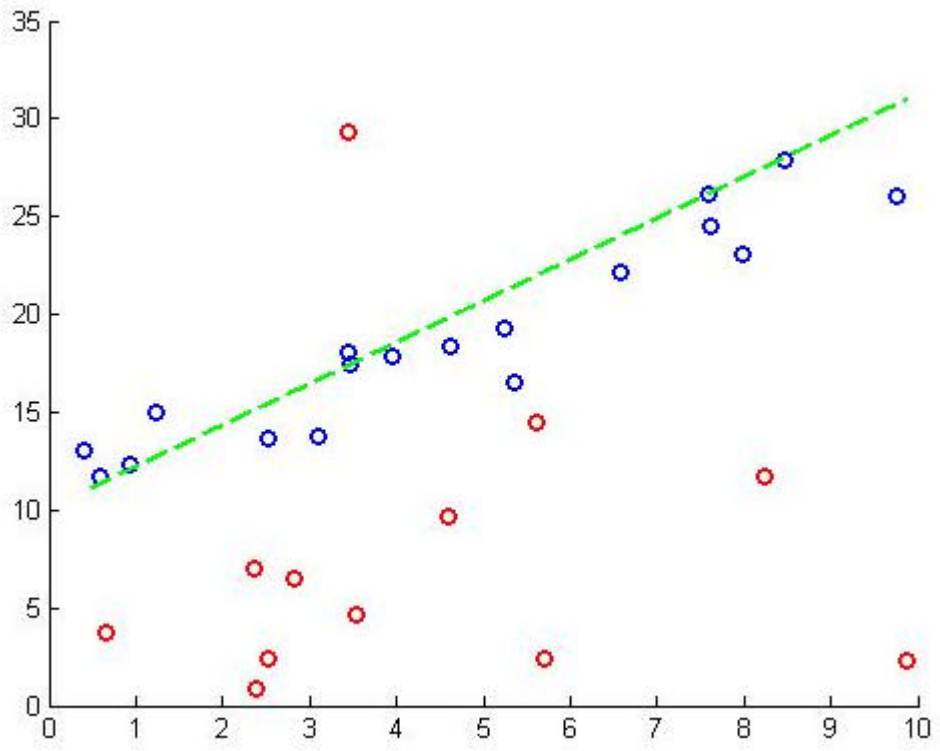


Iteration 3

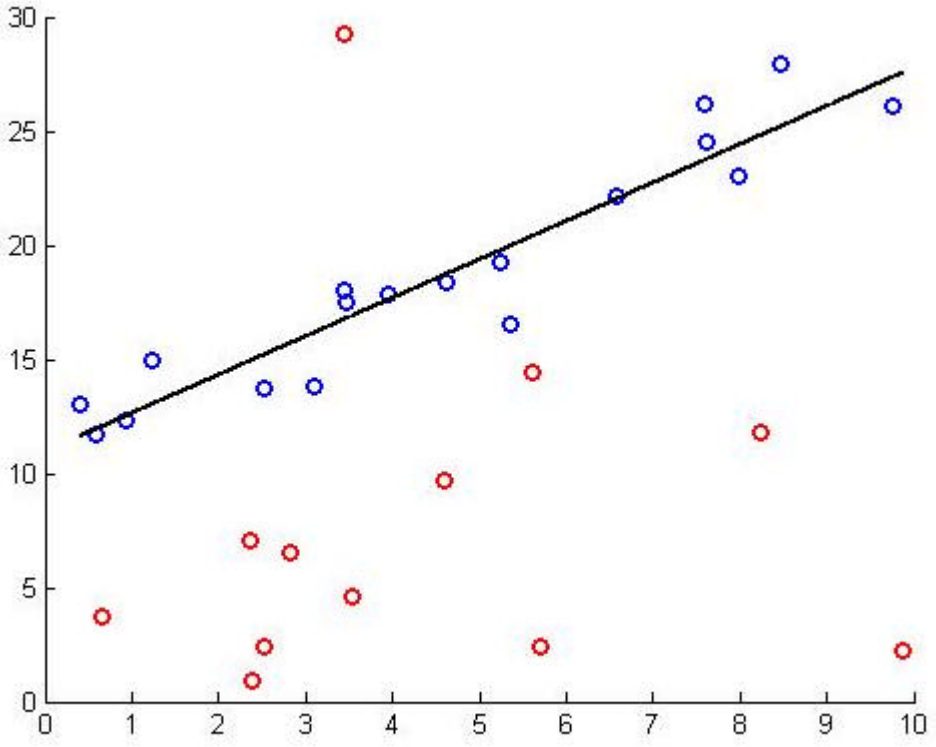
Here, RANSAC randomly chose two inliers into the sample subset.



As the sample subset includes only inliers, with a suitable error tolerance threshold, all the inliers could be captured as the consensus.



Having all the inliers in the consensus, the model is passes the minimum consensus threshold. Then, a final model is calculated with using all the data samples in the consensus with a least-square optimization.



Here, RANSAC managed to ignore all the outliers and trained the model with only using inliers.

Effects of Parameters

Number of Iterations

In the experiment, number of iterations for the RANSAC is calculated appropriate to the setting.

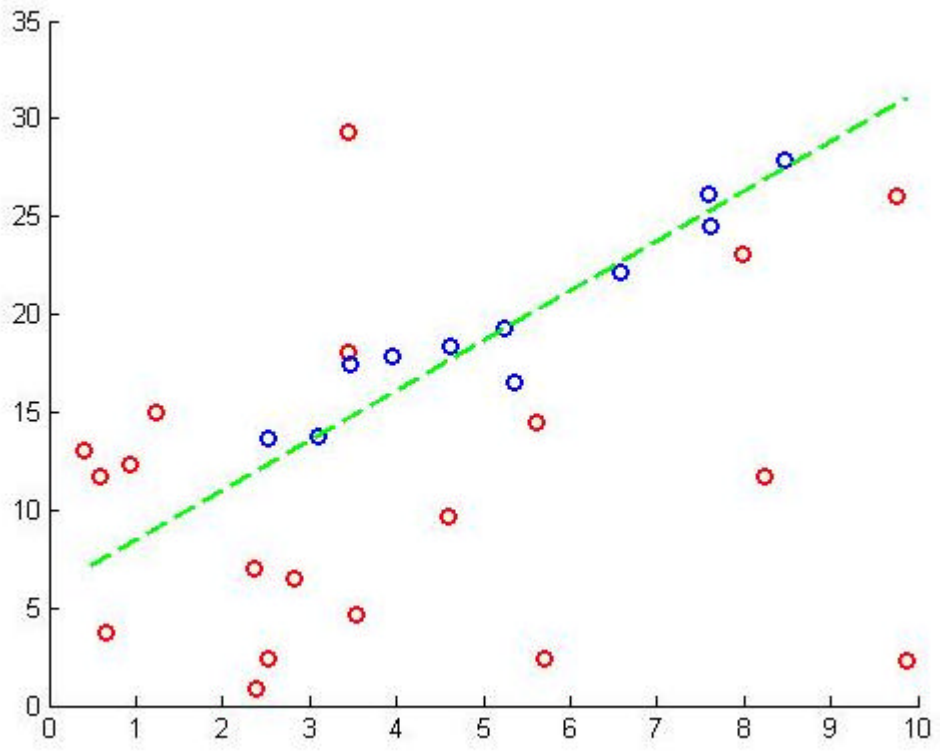
$$w \equiv 0.6 \ , \ P(\text{success}) \equiv 0.99 \ , \ n \equiv 2$$

$$k = \frac{log(1 - P(\text{success}))}{log(1 - w^n)} = \frac{log(1 - 0.99}{log(1 - (0.6)^2)} \approx 11$$

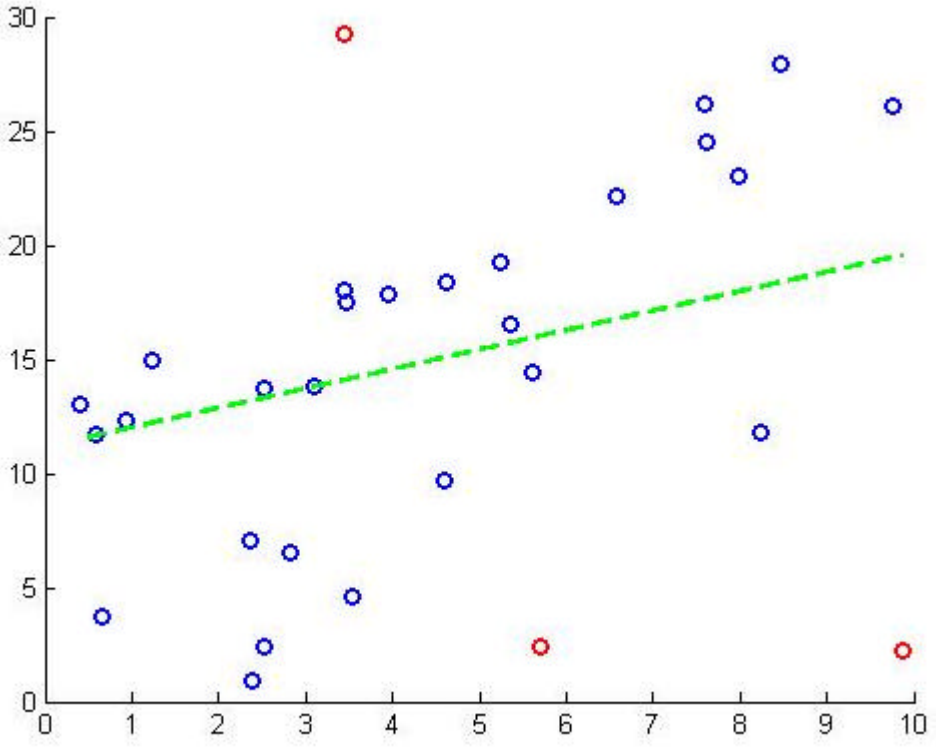
Error Tolerance Threshold

Error Tolerance Threshould should be chosen suitable to the setting. If not, RANSAC would fail to find the correct result.

If t is chosen smaller than it should, even RANSAC chose two inliers to the sample subset, it would fail to achieve a consensus that includes all of the inliers.



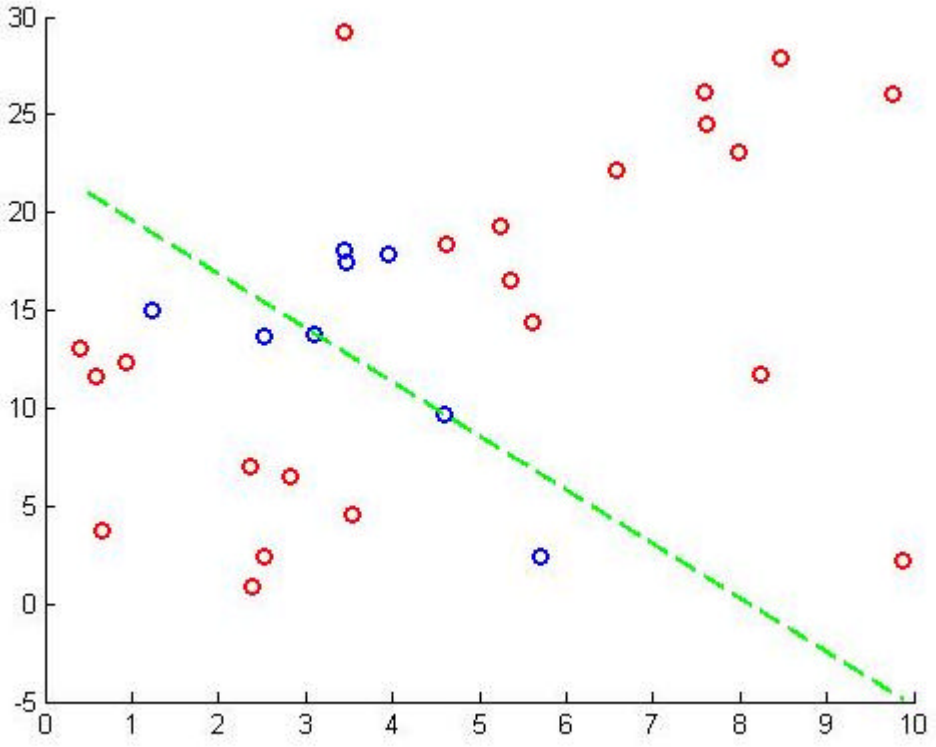
Or if t is chosen largely, the generated consensus would include some outliers even if the sample subset consists of only inliers.



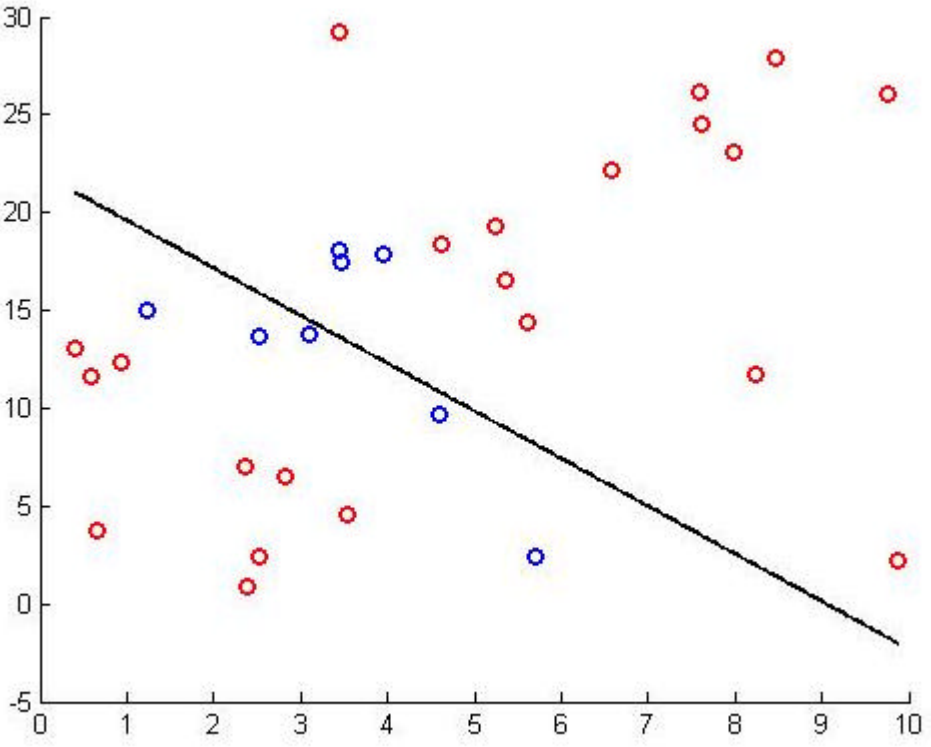
Minimum Consensus Threshold

Minimum Consensus Threshold has also an important role on the behaviour of RANSAC.

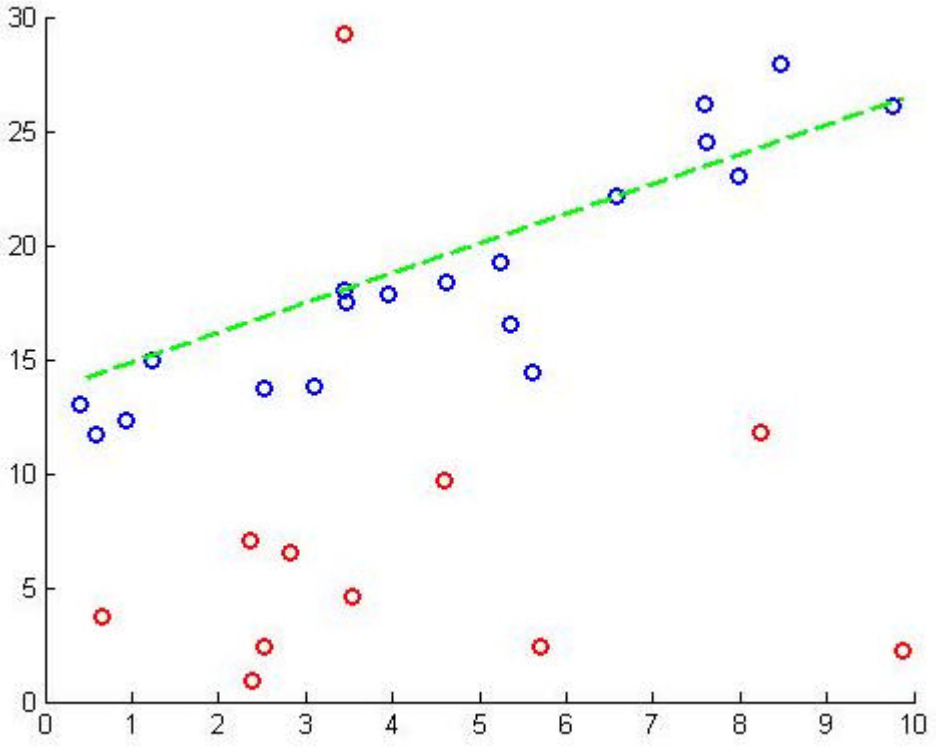
If d is chosen small, some random line that fits a portion of the samples would be considered as a valid consensus.



A final model would be generated with the parameters of this wrong estimation.



Either, if d is chosen too large, even a good estimate wouldn't pass the minimum consensus threshold and no final model results would be found by RANSAC.



Implementation

Applications

As RANSAC is a general algorithm that can be used in any parameter problem, it has a wide range of popularity. If a problem could have some wrong data samples due to measurements or hypothetical errors, RANSAC is one of the considered algorithms in order filter out gross-erroneous samples.

However, it is developed by Fischler and Bolles in order to solve Location Destination Problem (LDP) in Automated Cartography. Furthermore, it is widely used in Fundemental Matrix Estimation in order to find the similarities between the images. Hence, it could be said that RANSAC is mostly used in Image Processing and Computer Vision area.

Conclusions

Advantages

- robust against outliers
- a general method that can be applied most of the cases
- fast in huge data sets
- easy implementation
- modularity
- interpretability

Disadvantages

- has a certain probability of success
- requires prior knowledge about data
- number of iterations increases logarithmically with outlier percentage
- untrivial parameter assignments

References

- M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. (http://doi.acm.org/10.1145/358669.358692) Comm. of the ACM, Vol 24, pp 381-395, June 1981.
- Wikipedia - RANSAC (http://en.wikipedia.org/wiki/RANSAC)
- H. Cantzler. Random Sample Consensus(RANSAC) (http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/CANTZLER2/ransac.pdf) Institute of Perception, Action and Behaviour Division of Informatics, University of Edinburgh.
- R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2002
- O. Chum, J. Matas,Randomized RANSAC with $T_{d,d}$ test (http://www.bmva.ac.uk/bmvc/2002/papers/50/full_50.pdf)

Links

- Google (http://www.google.com/)
- Wikipedia (http://www.wikipedia.org/)