Fault Detection of Large Amounts of Photovoltaic Systems

Patrick Traxler

Software Competence Center Hagenberg, Austria
patrick.traxler@scch.at

Abstract. We study a model-based approach to detect sustainable faults of photovoltaic systems. We describe models and algorithms which allow us to analyze a large amount of photovoltaic systems. Our approach is based on median regression. We assume that we know the irradiance and produced energy of a photovoltaic system. The particular challenge of the data analysis problem we study here is the possible huge amount of photovoltaic systems and their continuous streams of data.

1 Introduction

Photovoltaic (PV) systems exist in large number. They are a common source of energy for resident homes. Many PV systems are equipped with sensors for measuring power, plane-of-array (POA) irradiance, and module temperature. Sensor data is often sent to a data center where it is stored and analyzed. One reason for measuring environmental and system quantities is to detect malfunctioning systems automatically. In this work we deal with the problem of fault detection. An emphasis is on analyzing many thousands to tens to possible hundreds of thousands of PV systems with a reasonable amount of computing resources.

An additional difficulty besides the large amount of PV systems is that we do not know of the faults in advance, i.e. data is not labeled. Labels of a particular measurement could be *tree-leaves* and *shading*, meaning that parts of the panels are covered by tree leaves and that shading occurred. We do not have this kind of information. We assume only that we receive power and irradiance measurements from a PV system with a reasonable quality; sometimes also the module temperature. We also assume that we get measurements for at least every 15 minutes. We do not make any other assumption. In particular, we do not need to know anything about the design or deployment of the PV system.

To overcome the lack of information we follow a model-based approach. We define the normal behavior of a PV system mathematically and say that it is malfunctioning if it does not show normal behavior. Our models are an adaption of models from [8]. We apply median regression on data of a whole day and use an indicator for the fitness of the model. If the fitness of the model is large enough the PV system passes the test for the particular day. Otherwise we say it is malfunctioning.

We describe and discuss our method in detail in Sec. 2. We remark already here that many of our design decisions relate to the large amount of data to be processed. We discuss computational efficiency in Sec. 3.

Results We distinguish between a *minor fault*, a fault occurring on a single day, and a *sustainable fault*, a fault occurring on many days. Our result is a method which performs in particular well for detecting sustainable faults.

We apply our method to real data of 21 PV systems and roughly 12500 days. Two PV systems showed a sustainable fault for a couple of months. For the remaining PV systems we get a small fraction of detected faults: up to 5.7% of all analyzed days per PV system. These detected faults are either minor faults or false detections. Both erroneous PV systems show faults during at least 30% of the days, i.e. considerable more than 5.7%. This allows us to identify PV systems with a sustainable fault.

Concerning computational efficiency, it requires less than 10 minutes to analyze the 12500 data sets on current commodity hardware. Our method works on many but small data sets. It does not depend on historical data. Moreover, it is possible to speed up the method by designing specialized heuristics. We provide an example of such a heuristic.

Motivation The main motivation of our work comes from the following use scenario. We want to monitor a large amount of deployed PV systems. As described above, the PV systems send measurements of power, irradiance, and possible module temperature to a data center. The problem is to detect faulty systems as good as possible. On the one side, we would like to know of a problem as soon as possible. On the other side, we want as little detections as possible. The reason for the latter is the large amount of PV systems. Even a small fraction of (unnecessary) detections per PV system would yield a large number of overall detections. Ten percent detections of 100000 PV systems are 10000 detections per day. We consider such a situation as not manageable: Monitoring and alerting might be the basis for deciding whether technical service is reasonable or even necessary. This consideration is the reason why we are interested in sustainable faults.

1.1 Related Work

As mentioned above our models are an adaption of models from [8]. Our models are time-dependent, unlike the models in [8]. Another difference is that our models do not require any knowledge of system parameters of the PV systems.

Other models are not appropriate if they involve information we do not have. Some of these models, e.g. [2], may be considered as simulations since they model the current flow of the PV system. Although these models are probable more accurate than the one used here they are not applicable. In the case of simulations, they require knowledge of the design.

Fault detection of PV systems is the topic of e.g. [2, 5, 3, 1]. Firth et al. [5] identify zero-efficiency (no energy generated) patterns as a fault. They also con-

sider *sustainable* faults. These faults last for more than one day. Although sporadic zero-efficiency faults reduce the amount of produced energy, the resulting energy loss is still moderate compared to the energy loss of a sustainable fault. Our approach may be considered as a generalization of detecting zero-efficiency faults. We check whether irradiance and generated energy are in a consistent relation throughout the day. In other words, we check if the conversion of sunlight to electricity works according to a model.

Another type of fault occurs if the panels are covered by snow, tree leaves, or something else. It is considered e.g. in [3]. The fraction of panel area covered is a crucial parameter.

Median regression is a special case of quantile regression. It is also called ℓ_1-regression. It is applied as an alternative to ordinary least squares estimation. An exposition on quantile regression is e.g. [6]. Usually we use linear programming [6] to solve ℓ_1-regression. A recent advance on algorithms for quantile regression is [10].

2 Method

We describe first the models we are going to use. Second, we present a measure for the fitness of a model estimated by median regression. It resembles the coefficient of determination or R^2-value in the context of ordinary least squares estimation. Finally, we present experimental results and discuss them.

2.1 Models: Power, Plane-of-Array Irradiance, and Module Temperature

A PV system converts sunlight into electricity. This process is influenced by a several factors: The solar irradiation, the number and size of panels, the orientation and tilt of the panels, the placement of the sensor(s) for measuring POA irradiance and module temperature, the strategy of tracking the maximum power point, and others. We do not consider the power inverter here, i.e. we only consider direct current.

We assume that we know the power, POA irradiance, and possible the module temperature. In what follows time is discrete. In a realistic scenario we have measurements every 5, 10, or 15 minutes. The 5-minutes sampling frequency has shown to be reasonable. In comparison to 1-second intervals the amount of generated data is considerable smaller and 1-hour intervals would yield too few data samples for estimation on some days.

We consider a single PV system $S_t = (P_t, E_t, T_t)$ at discrete time t. Here, P_t [W] is the power, E_t [W/m^2] the POA irradiance, and T_t [°C] the module temperature. The POA irradiance E_t is the system input, P_t the system output, and T_t a system state variable. For some integer $d \geq 1$ and i.i.d. random noise

ε_t we define

$$P_t = \sum_{j=t-d}^{i+d} a_j \cdot E_j + b \cdot E_t \cdot T_t + c \cdot T_t + \varepsilon_t. \tag{1}$$

and without module temperature

$$P_t = \sum_{j=t-d}^{i+d} a_j \cdot E_j + \varepsilon_t \tag{2}$$

Our models are motivated by a time-independent model found in [8], see Eq. (2) in [8]. The model in [8] fits our requirement that it only involves power, POA irradiance, and module temperature:

$$P = a \cdot E + b \cdot E \cdot T + c \cdot T. \tag{3}$$

One difference to Eq. 1 is the dependency on time. Another difference is that the parameters a, b, c are assumed to be known in [8]. We are going to motivate our models next.

Time Shift Analyzing real data has shown a time shift in the data for some PV systems. Note that unlike in an auto-regressive model the power P_t at time t depends not only on the previous values (e.g. $P_{t-1}, P_{t-2}, \ldots$) but also on future values. The variable E_t is an exogenous variable for which we observed slight shifts in time in comparison to power. In other words, energy was generated too early or too late.

Not every PV system showed this behavior. There are however some situations in which tolerance w.r.t. time shifts are preferable. For example, if the irradiance sensor and the solar panels differ in orientation/tilt or if the panels have different orientation/tilt. We do not consider these situations as a fault here.

Concrete values for d are set such that the power P_t depends on the POA irradiance values of the last and next hour. For example, we set $d := 12$ if we have measurements in 5-minutes intervals.

Random Noise A common way to model random noise is to define the ε_t's as independent Gaussians with zero mean and constant standard deviation. This is also what statistical tests for normality indicate for well-functioning PV systems. For PV systems and days with strong faults it is unlikely that the residuals are normally distributed. This is actually what we want since it allows us to analyze the residuals of the regression.

2.2 Method: Median Regression and Model Fitness

Median regression is a special case of quantile regression. See e.g. [6]. Important here is the estimator associated with median regression. Given data D as $(d-1)$-dimensional vectors of reals $x_1, \ldots, x_n$ and reals $y_1, \ldots, y_n$ the problem

is to find a $(d-1)$-dimensional vector of reals u such that $\sum_{i=1}^{n} |(x_i, u) - y_i|$ is minimized where $(\,\cdot\,,\,\cdot\,)$ denotes the inner product. For a solution u^* of this optimization problem we define $\hat{y}_i := (x_i, u^*)$. The estimator motivates the following definition.

Model Fitness

$$\mathsf{Fit}(D) := 1 - \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |y_i|}$$

Our definition is essentially the error of the estimation divided by a scaling factor. It resembles the coefficient of determination or R^2-value. Experimental results indicate that $0 \leq \mathsf{Fit}(D) \leq 1$. A value close to 1 indicates that the data D fits the model well. In particular, $\mathsf{Fit}(D) = 1$ iff $\hat{y}_i = y_i$ for all i. It approaches 0 if the variance in the data increases.

In our situation, the y_i's are the measured power values P_t and the $\hat{y}_i$'s are the estimated power values. As an example, for Eq. 2 the x_i's are the POA irradiance values: $x_i = (E_{t-12}, \ldots, E_t, \ldots, E_{t+12})$ if $y_i = P_t$.

Method Our method is to check whether $\mathsf{Fit}(D) < \theta_0$ where D is the data for one day and one PV system. We only take data points with POA irradiance values larger than E_0. If $\mathsf{Fit}(D) < \theta_0$ the outcome of the method is *fault*, otherwise *no-fault*. Our restrictions to POA irradiance values larger than E_0 comes from the fact that some PV systems only start to produce energy if the POA irradiance is large enough and possible higher accuracy of the measurements. We use in our experiments the values $\theta_0 = 0.9$ and $E_0 = 25$ W/m^2.

The parameter θ_0 is critical to our application. A too large variance in the data on days with no faults would render our approach useless. In this case, the fitness values are too small. Experimental results showed however that a value 0.9 is a reasonable choice to distinguish between fault and no-fault. It works actually for *all* PV systems we tested. We derived the value 0.9 by picking about 5 different PV systems and some days with no faults. We observed that $\mathsf{Fit}(D) > \theta_0 = 0.9$ for every selected daily data D.

Finally, to detect a sustainable fault, we check if for a period of $n \geq 14$ days at least $\frac{n}{3}$ of them were recognized as faulty.

Median Regression and Ordinary Least Squares OLS-estimation is defined as median regression but with the objective function $\sum_{i=1}^{n}((x_i, u) - y_i)^2$. Median regression is less sensitive to some minor faults. In particular, faults which last shortly yield a smaller error value in comparison to OLS-estimation. This is due to the usage of absolute differences in contrast to quadratic differences. This preference of faults with a longer duration was one of the motivations for median regression.

Comparing Daily Power Efficiency Our method checks the relation between irradiance and the generated energy of a single day. It may however happen that some panel-cover fault occurs overnight. During night both irradiance and power

are zero or close to zero, a situation we consider correct. However, if the panels are covered during night our method cannot detect it.

A simple and computational efficient approach to tackle this problem is the following. We set the POA irradiance E_t to 1000 W/m^2 and the module temperature T_t to 25 °C and then calculate the power P according to Eq. 1 or 2. We do the same for the previous day and get P'. We then check if P is more than 20% below or above of P'. In this case we say a fault happened, otherwise not. This approach does not require any complex computations. We only need to store the model parameters for one day. The values are motivated by the Standard Test Conditions for PV systems.

2.3 Experimental Results and Fault Types

Experimental Results We applied our method to real data. It comes from 21 PV systems. Two of them have a sustainable fault. We list results in Table 1. In the second column we list results for our primary model (Eq. 1), in the third for our model without module temperature (Eq. 2), and in the fourth for the time-independent model (Eq. 3). We list the average number of detected faults per year.

We recall that we do not know whether a detected fault is real. A manual analysis suggests that two PV systems have a sustainable fault. These two sustainable faults, PV system 13 and 17, were detected. The irradiance sensor for PV system 13 started to deliver too high values, in particular values larger than 100 W/m^2 during night. And PV system 17 stopped to produce energy at some days. During the first half of a particular day it did not produce energy, during the second half it did. At another day it produced a constant power for an hour although it should not. PV system 17 was also maintained during the observation period which influenced the outcome of our method.

We also list results for different values of θ_0, namely $0.85, 0.8, 0.75$, in Table 1. And in the last line of Table 1 we list the fraction of detected faults. The fraction relates to the number of days analyzed *without* PV system 13 and 17. We recall that our motivation is to analyze a large amount of PV systems. A small fraction is preferable. We draw the conclusion that Model (Eq. 1) is the best w.r.t. this number and at the same time allows us to distinguish between PV systems with a sustainable fault and PV systems with minor faults.

We also picked a single PV system and labeled every day manually, i.e. we decided whether a day has a fault or not. We identified 61 days with a minor fault from a total of 339 days. The results for different fitness values θ_0 are depicted in Table 2. The false positive rate is the number of falsely detected faults. We note that in comparison to the case of sustainable faults as discussed above the labeling is less accurate.

Fault Types There are different types of faults. Fig. 1 shows a day without fault (a) and a day with almost zero-efficiency fault (b). Both are real examples with fitness value greater than $\theta_0 = 0.9$. Graph (c) and (d) show POA irradiance. Examples (e) and (f) have both fitness values smaller than θ_0. We generated

Table 1. Numbers are rounded averages of faults per year. Results are for 19 PV systems with minor faults and two PV systems (13, 17) with sustainable faults. We distinguish between three different models and different fitness values.

	Median regression					
	Model 1	Model 2	Model 3	Model 1		
System	Fit 0.9	Fit 0.9	Fit 0.9	Fit 0.85	Fit 0.8	Fit 0.75
1	9	22	32	4	1	1
2	16	27	27	11	5	5
3	1	1	1	0	0	0
4	8	9	15	4	1	1
5	10	17	36	3	2	2
6	15	37	107	2	0	0
7	2	3	7	1	0	0
8	8	12	27	1	0	0
9	11	20	31	7	6	4
10	8	13	19	6	6	5
11	5	10	15	3	3	1
12	3	9	21	2	0	0
13	**111**	**153**	**200**	**82**	**49**	**27**
14	2	6	11	0	0	0
15	4	8	13	2	1	1
16	1	2	9	0	0	0
17	**209**	**289**	**314**	**127**	**59**	**39**
18	21	25	52	13	8	6
19	5	15	14	4	3	2
20	9	9	16	7	6	6
21	8	12	20	5	3	2
Total	*1.5%*	*2.7%*	*5.0%*	*0.7%*	*0.5%*	*0.4%*

Example (e) from (a) by decreasing the power by 40% for two hours. And we generated Example (f) from (b) by setting the power to roughly 15 W for two hours, i.e. we repeated the energy loss from Example (b) for two hours.

In what follows we discuss two fault types: covered panels (e.g. Example (e)) and a generalization of zero-efficiency faults (e.g. Example (f)). Another important type of fault, which we do not consider here, is the decrease of efficiency over years of a working PV system, called degradation, which is for example caused by soiling.

Table 2. The False Negative Rate and False Positive Rate for a single PV system. A total of 339 days were analyzed.

Fitness	False Negative Rate	False Positive Rate
0.9	89%	0%
0.95	80%	0%
0.96	74%	0%
0.97	66%	1%
0.98	54%	4%
0.985	46%	11%

Fault Type: Covered Panels One common type of faults results from covered panels. Let a_t be the fraction of covered area at discrete time t. It should be 0, i.e. the panels are not covered. It is 1 if the panels are covered completely. The fraction a_t can vary strongly in time, for example in case of melting snow or shading. In case of tree leaves it stays constant over time.

In the context of fault detection only the fraction a_t is of importance. Example (e) in Fig. 1 is motivated by the intuition that if 40% of the panels are covered (i.e. $a_t = 0.4$) then we observe roughly a 40% decrease in the power.

It is important here to note that our method and actually any method is only capable of detecting faults if a_t is large enough for some period of time. As an example, a single leave lasting for a minute is (presumable) not detectable.

Moreover, our method is not capable of detecting a total snow cover since zero-power and zero-irradiance result in a perfect model fit.

The situation is different with melting snow and shading. Shading in particular is a problem since it may influence the long-term energy efficiency of the PV system considerable. Our method can only detect shading if the fraction a_t caused by shading and its duration is large enough; e.g. Example (e) in Fig. 1.

Fault Type: Zero-Efficiency Faults and its Generalization A zero-efficiency fault happens if no energy was generated in the presence of large enough irradiance values. Zero-efficiency faults are identified in [5] as a cause of drastic energy loss. In [5] they are related to a string disconnect. They can easily be detected by checking if the irradiance is too large, i.e., whether the PV system should have generated energy. Our approach is motivated by [5] but is more general. To see this, consider Example (f) in Fig. 1. It resulted from Example (b) by setting the power to roughly 15 W for two hours. The faults (b) and (f) in Fig. 1 are not zero-efficiency faults but clearly an energy loss.

3 Computational Efficiency

An important property of our method is computational efficiency. The major benefit of our method is that we process a large amount of small data sets.

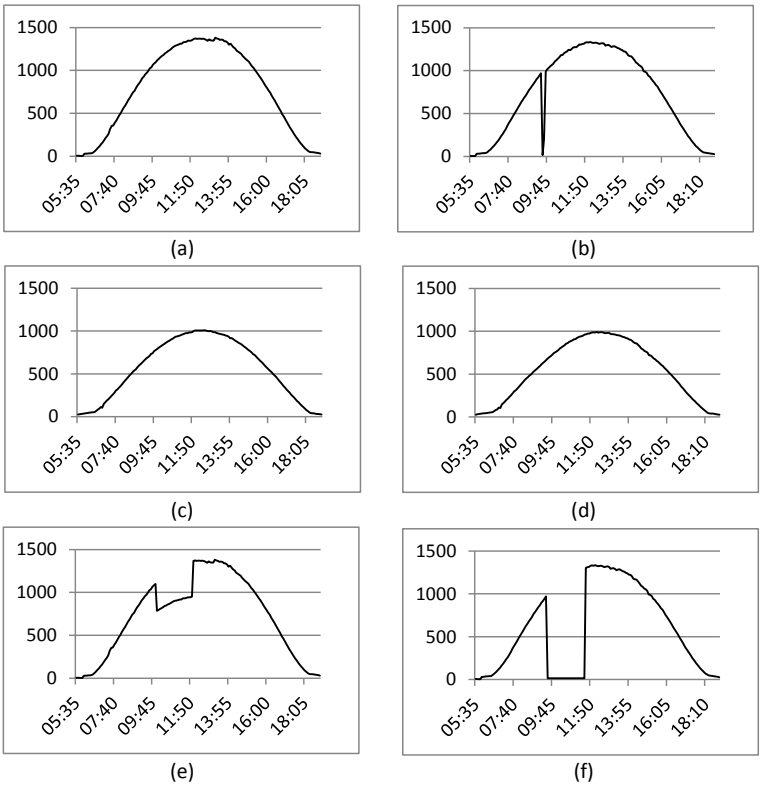

Fig. 1. The x-axis represents time, the y-axis measured power [W] for (a), (b), (e), (f), and measured POA irradiance [W/m^2] for (c), (d). Example (a) and (b) are real data: PV system Uni-solar 64W from [9] with plane-of-array irradiance (c) and (d) and for day 2013-06-03 and 2013-06-04 respectively. Example (e) and (f) are generated from (a) and (b) respectively. In Example (b) at around 9:45 the measured power drops to roughly 15 W.

For example, a data set D consists of at most $288 = 24 \cdot 12$ data points in case we have measurements every 5 minutes. The data set is even smaller if we omit data points with zero irradiance and zero power. Since every data set D is small and due to the existence of efficient algorithms for median regression we only need a small amount of computing resources. The other advantage is the possibility to distribute the data sets to analyze them in parallel on a single machine or a network since the data sets are analyzed independently of each other. In particular, we do not need to analyze historical data. This is in contrast to (e.g. prediction-based) approaches which may involve parameter estimation applied to weeks or even months of data. Multiplying with hundreds of thousands PV systems the running time and memory requirements may be considered infeasible.

A direct implementation shows that analyzing roughly 12500 data sets can be done in less than 10 minutes on a current computer of about 2.7 GHz clock speed using a single computing core. Our implementation was not optimized. We used the statistical programming language R and in particular the implementation of the simplex method for quantile regression in R [7]. It is thus reasonable to assume that an optimized implementation works considerable faster. Moreover, we conclude that we can analyze 125000 data sets on machine with 10 cores in less than 10 minutes.

To see the difference to other approaches let us assume we make an ordinary-least-squares (OLS) estimation of the last 30 days to learn a model. We use the model to predict the generated energy of the current day using irradiance measurements of the current day. This yields at the end of the day a predicted amount of produced energy E' and the measured amount E. If E' deviates too strongly from E we say we have a fault. This prediction-based approach is common in fault detection.

OLS involves the creation of a matrix of n columns and n rows where n is as above the number of data points. In our case $n^2 = (30 \cdot 288)^2 = 74649600$. Multiplying with the number of PV systems and the bytes allocated per data point yields the total memory requirements. It is in the terabyte range for hundreds of thousands of PV systems. The running time requirements are even worse.

Improvements To get even better computational performance we designed a heuristic with improved performance. It is able to handle data as it arrives. Data from a PV system arrives in chunks of data for one hour. Processing any of these 24 data chunks as soon as it arrives has the benefit to make use of computing resources evenly throughout the day.

We recall our method: Test whether $F := \mathsf{Fit}(D) < \theta_0$. Assume we have $\hat{F} = \hat{F}(D)$ such that $\hat{F} \leq F$ for all data sets D. If $\hat{F} > \theta_0$, then $F > \theta_0$. We describe such a heuristic with the additional property that it can process data streams incrementally. Altogether we have the following improvement of our initial method.

Input: Data set D for a single day and PV system (arriving in data chunks).
Output: Decision whether the PV system has a fault or not.
Parameter: θ_0.

1. Compute (using a fast heuristic) an approximation $\hat{F}$ for $\mathsf{Fit}(D)$.
2. If $\hat{F} > \theta_0$ then return "No fault".
3. Else compute $F = \mathsf{Fit}(D)$.
4. If $F > \theta_0$ then return "No fault", otherwise return "Fault".

Fig. 2. Improved method using a heuristic

Example of a Heuristic We present an example of a heuristic which can be used with our improved method, Fig. 3. The heuristic is specialized for the data we process. In particular, it is incremental. This means that it can process data as it arrives. For this purpose we use a data structure for searching. (See e.g. Chapter *Red-Black Trees* in [4].) We remark that our method, Fig. 3, can be combined for example with the algorithm in [10].

The heuristic gets an irradiance value and power pair (E_t, P_t) at discrete time t after it received $((E_1, P_1), \ldots, (E_{t-1}, P_{t-1}))$. It outputs an approximation $\hat{F}$ for $F = \mathsf{Fit}(D)$ such that $\hat{F} \leq F$.

The idea is to find an approximation $\hat{a}$ for $a^* := \min_{a \in \mathbf{R}} \sum_{t=1}^{n} |a \cdot E_t - P_t|$ such that $a^* \leq \hat{a}$. Second, $\hat{a}$ is our estimation for the parameter of variable E_t of Eq. 1 and 2 which we denoted by a_t. The remaining parameters $a_{t-d}, \ldots, a_{t+d}, b$ and c of Eq. 1 and 2 are set to 0. The resulting approximation is $\hat{F}$ and it holds that $\hat{F} \leq F$.

We assume that $((E_1, P_1), \ldots, (E_{t-1}, P_{t-1}))$ is sorted according to $\frac{P_i}{E_i}$, i.e., it is stored in a search data structure S. We omit pairs with $E_i = 0$. We also maintain the current solution $\hat{a} = \frac{P_j}{E_j}$ for some $j \in \{1, \ldots, t-1\}$. In words, the solution to our approximation problem is a value from $\frac{P_1}{E_1}, \ldots, \frac{P_{t-1}}{E_{t-1}}$.

We describe the insert operation for the new pair (P_t, E_t) next. First, we insert it into S w.r.t. $\frac{P_t}{E_t}$. Second, we update $\hat{a}$ according to the following rules. Let $a' := \frac{P_t}{E_t}$ and $e(\alpha, t) := \sum_{i=1}^{t} |\alpha \cdot E_i - P_i|$. Moreover, let $a'_0 := \frac{P_{j_0}}{E_{j_0}}$ where (P_{j_0}, E_{j_0}) is the element before (P_j, E_j) in S. Let $a'_1 := \frac{P_{j_1}}{E_{j_1}}$ where (P_{j_1}, E_{j_1}) is the element after (P_j, E_j) in S. The rules are: If $a' > \hat{a}$ and $e(a'_1, t) \geq e(\hat{a}, t)$ then set $\hat{a}$ to a'_1. If $a' < \hat{a}$ and $e(a'_0, t) \geq e(\hat{a}, t)$ then set $\hat{a}$ to a'_0.

Experimental results show an improved running time if we compare the running of our initial method implemented in R and our improved method, Fig. 3. Our improved method is at least 5 times faster. This is due to the fact that our heuristic is fast and that it computes reasonable approximations. To be more precise, our heuristic yielded a total fraction of detected faults of roughly 8.75%; compare Table 1. For this fraction we had to check a second time if the fitness is larger than θ_0. Since our heuristic is fast we observed a total running time of less than 2 minutes.

4 Conclusion and Outlook

We described a method for fault detection of a large amount of PV systems. It has the benefit that it works on many but small data sets. Our method should make it possible to analyze up to hundreds of thousands of PV systems on a daily basis. Our method is designed to detect sustainable faults and to be less sensitive to minor faults such as melting snow covers of tree leaves. It performs well on real data.

A topic that we did not deal with here is fault diagnosis. The problem is to classify the fault. E.g. as shading, tree leaves, zero-efficiency. We note that our

choice for ℓ_1-regression over ℓ_2-regression was partly motivated by fault diagnosis due to the well-known "robustness" property of the ℓ_1-estimator. The idea is to analyze the residuals of the ℓ_1-regression. Whether this yields an accurate method for fault diagnosis is the topic of further research.

References

1. H. Braun, S. T. Buddha, V. Krishnan, A. Spanias, C. Tepedelenlioglu, T. Yeider, and T. Takehara. Signal processing for fault detection in photovoltaic arrays. In *37th IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1681–1684, 2012.
2. K. H. Chao, S. H. Ho, and M. H. Wang. Modeling and fault diagnosis of a photovoltaic system. *Electric Power Systems Research*, 78(1):97–105, 2008.
3. A. Chouder and S. Silvestre. Fault detection and automatic supervision methodology for PV systems. *Energy conversion and Management*, 51:1929–1937, 2010.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3rd ed.)*. MIT Press and McGraw-Hill, 2009.
5. S.K. Firth, K.J. Lomas, and S.J. Rees. A simple model of PV system performance and its use in fault detection. *Solar Energy*, 84:624–635, 2010.
6. R. Koenker. *Quantile Regresssion*. Wiley Online Library, 2005.
7. R. Koenker. *quantreg: Quantile Regression*, 2012. R package version 4.94.
8. B. Marion. Comparison of predictive models for PV module performance. In *33rd IEEE Photovoltaic Specialist Conference*, pages 1–6, 2008.
9. University of Arizona, 2013. http://uapv.physics.arizona.edu/downloadData.php.
10. J. Yang, X. Meng, and Mahoney M. W. Quantile regression for large-scale applications. In *30th International Conference on Machine Learning*, (to appear) 2013.