CLUSTERING USING SUM-OF-NORMS REGULARIZATION
WITH APPLICATION TO PARTICLE FILTER OUTPUT COMPUTATION

Fredrik Lindsten, Henrik Ohlsson and Lennart Ljung

Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden.
{lindsten, ohlsson, ljung}@isy.liu.se

ABSTRACT

We present a novel clustering method, formulated as a convex optimization problem. The method is based on over-parameterization and uses a sum-of-norms (SON) regularization to control the trade-off between the model fit and the number of clusters. Hence, the number of clusters can be automatically adapted to best describe the data, and need not to be specified a priori. We apply SON clustering to cluster the particles in a particle filter, an application where the number of clusters is often unknown and time varying, making SON clustering an attractive alternative.

Index Terms— Clustering, particle filter, sum-of-norms

1. INTRODUCTION

Clustering is the problem of dividing a given set of data points into different groups, or clusters, based on some common properties of the points. Clustering is a fundamental cornerstone of machine learning, pattern recognition and statistics and an important tool in *e.g.,* image processing and biology. Clustering has a long history and, naturally, a huge variety of clustering techniques has been developed. We refer to [1] for an excellent survey of the field.

Many existing approaches, such as the well known *k-means* method [2, 3], are formulated in terms of non-convex optimization problems. The solution algorithms for such problems can thus be sensitive to initialization. It can easily be shown that k-means clustering, as an example, can yield considerably different results for two different initializations [4]. A number of clustering algorithms with convex objectives, which therefore are independent of initialization, have been proposed, see *e.g.,* [5, 6].

In this paper, we present a novel method for clustering, called sum-of-norms (SON) clustering. Two key features of this approach are that **i)** the problem is convex **ii)** the number of clusters need not be specified beforehand. In the complementary material [7], we analyse the method further and show that it can be seen as a convex relaxation of the k-means problem. A similar formulation to SON clustering was previously discussed by [8]. However, there is a main difference in how the regularization term is constructed. The presentation in [8] can be seen as inspired by *lasso* [9, 10] while our formulation can be seen as inspired by *group-lasso* [11]. Related contributions, using SON regularization in other contexts, are for instance [12, 13].

Besides providing a generally applicable clustering method, we shall in this paper focus on one specific application, namely particle clustering in a particle filter (PF). The PF [14, 15] is used for state inference by Monte Carlo integration in general (typically nonlinear/non-Gaussian) state-space models. The posterior distribution of the state is represented by a set of weighted point-masses, or particles. To be able to cluster the particles online, without prior knowledge about the number of clusters, opens up for a range of different algorithmic modification and extensions.

There are several PF based methods in the literature, in which particle clustering is an essential part. The distributed PFs by [16] rely on fitting a Gaussian mixture model (GMM) to the set of particles. Here, they use the EM algorithm [17]. However, this requires the number of clusters to be specified beforehand, and it can also be slow to converge [18]. The clustered PF by [19] uses a simple greedy method, but also state that "There are more robust clustering algorithms, based on the EM algorithm; however, these methods rely on knowing the number of clusters a priori". A similar approach is used in the mixture PF by [20], where k-means clustering is used together with splitting and merging of clusters in a heuristic manner. However, they also state that "The reclustering function can be implemented in any convenient way". In all of the above mentioned methods (and many more), SON clustering serves as an interesting alternative.

However, in this paper we focus on the clustering problem itself, to show that SON clustering can be used for particle clustering. We emphasise that, for the results presented in this paper, there is no feedback from the clustering to the PF. The clustering can thus (in this paper) be seen as an "add-on" to the filter. The idea is, as mentioned above, to illustrate the applicability of SON clustering for particle clustering, but the real purpose is of course to provide a clustering method which can be used as an intrinsic step in different particle based methods.

The remaining of this paper is organized as follows. In Section 2 we formulate the general clustering problem and present the novel SON clustering method. This is followed by a discussion on possible extensions of the method in Section 3. In Section 4 we turn to the problem of particle clustering in a PF and give experimental results using SON clustering. Finally, in Section 5 we draw conclusions.

2. SUM-OF-NORMS CLUSTERING

Let $\{x_j\}_{j=1}^N$ be a set of observations in $\mathbb{R}^d$. We wish to divide these observations into different clusters. Informally, points close to each other (in the Euclidian sense) should be assigned to the same cluster, and vice versa. Also, the number of clusters should not be unnecessary large, but we do not know beforehand what the appropriate number is.

It is natural to think of the clusters as subsets of $\mathbb{R}^d$, such that if any point x_j belongs to this subset it also belongs to the corresponding cluster. Consequently, we can say that each cluster has a centroid in $\mathbb{R}^d$. Now, since we do not want to specify how many clusters we are dealing with, we let μ_j be the centroid for the cluster containing x_j. Two x's are then said to belong to the same cluster if the corresponding μ's are the same. The sum-of-squares error, or the fit, is

then given by $\sum_{j=1}^{N} \|x_j - \mu_j\|^2$. Minimizing this expression with respect to the μ_j's would, due to the over-parameterization, not be of any use. The result would simply be to let $\mu_j = x_j$, $j = 1, \ldots, N$, *i.e.,* we would get N "clusters", each containing just one point. To circumvent this, we introduce a regularization term, penalizing the number of clusters. This leads to the *SON clustering* problem,

$$\min_{\mu_1 \cdots \mu_N} \sum_{j=1}^{N} \|x_j - \mu_j\|^2 + \lambda \sum_{j=2}^{N} \sum_{i<j} \|\mu_i - \mu_j\|_p, \qquad (1)$$

for some $p \geq 1$ (see Section 2.1 for a discussion on the choice of p). The name refers to the sum-of-norms (SON) used as a regularization. The reason for using SON is that it is a well known sparsity regularization, see *e.g.,* [11]. Hence, at the optimum, several of the terms $\|\mu_i - \mu_j\|_p$ will (typically) be exactly zero. Equivalently, several of the centroids $\{\mu_j\}_{j=1}^{N}$ will be identical, and associated x's can be seen as belonging to the same cluster, efficiently reducing the number of clusters.

Remark 1 (Sum-of-norms regularization)**.** *The SON regularization used in* (1) *is an ℓ_1-regularization of the p-norm of differences $\mu_i - \mu_j$, $j = 1, \ldots, N$, $i < j$. That is, the SON term is the ℓ_1-norm of the vector obtained by stacking $\|\mu_i - \mu_j\|_p$, for $j = 1, \ldots, N$, $i < j$. Hence, this stacked vector, and not the individual μ-vectors, will become sparse.*

The regularization parameter λ is a user choice that will control the tradeoff between model fit and the number of clusters. In many existing clustering algorithms (see *e.g.,* [1]), the user is asked to specify the number of clusters k beforehand. In SON clustering, this user choice is moved to the regularization parameter λ. Note that the number of clusters k does not appear in the criterion (1).

Another key property of SON clustering is that the criterion (1) is convex. That means that the global optimum can be found independently of initialization. Many existing clustering methods (there among k-means clustering) are dependent on a good initialization for a good result [4]. The same property also implies that convex constraints easily can be added to SON clustering.

2.1. Implementation aspects

Apart from the regularization parameter λ, we need to choose which norm to use in the regularization term, *i.e.,* to choose a value for p. We will in general use $p = 2$, but other choices are also possible. However, to get the properties discussed above, p should be chosen greater than one. With $p = 1$, we obtain a regularization variable having many of its components equal to zero, we obtain a sparse vector. When we use $p > 1$, the whole estimated regularization variable vector often becomes zero; but when it is nonzero, typically all its components are nonzero. Here, $p > 1$ is clearly to be preferred, since we desire the whole parameter vectors μ to be the same if they are not needed to be different. In a statistical linear regression framework, sum-of-norms regularization ($p > 1$) is called *group-lasso* [11], since it results in estimates in which many groups of variables are zero.

Also, when solving the problem (1), it is useful to apply an additional step. Having found the minimizing μ, say μ^*, of (1) we carry out a constrained least squares, where μ_i is set equal to μ_j if $\mu_i^* = \mu_j^*$. This is done to avoid a biased solution. In the following, we assume that the procedure of solving (1) is always followed by such a constrained least squares problem, whenever referring to SON clustering. However, note that this last step is relevant only if the actual centroid-values are of interest. To merely compute which x's that belong to which cluster, the last step can be skipped since we only need to know whether or not μ_i^* equals μ_j^*, not the actual values of the individual elements of μ^*.

2.2. Solution algorithms and software

Many standard methods of convex optimization can be used to solve the problem (1). Systems such as CVX [21, 22] or YALMIP [23] can readily handle the sum-of-norms regularization, by converting the problem to a cone problem and calling a standard interior-point method. For the special case $p = 1$, more efficient, special purpose algorithms and software can be used, such as `l1_ls` [24]. Recently, many authors have developed fast, first order methods for solving ℓ_1 regularized problems, and these methods can be extended to handle the sum-of-norms regularization used here; see, for example, [25, §2.2]. A code-package for solving (1) using CVX is available for download at `http://www.control.isy.liu.se/~ohlsson/code.html`.

3. EXTENSIONS

The problem (1) can be seen as the basic formulation of SON clustering. Here, we discuss two possible extensions to the method. First, since it is based on the Euclidian distance between data points, (1) can only handle linearly separable clusters. To address nonlinear clustering problems, the "kernel trick" can be used. We do not consider such problems further in this paper.

Second, it may be beneficial to add weights to the regularization term in (1). In [7], it is discussed that SON regularizations prefer unequally sized clusters. Since the sum in the regularization term ranges over all pairs of point, it will penalize distinct μ-values even if the corresponding data points are far apart. To circumvent this, we can localize the regularization penalty by adding data dependent weights. A modified optimization problem is then,

$$\min_{\mu_1 \cdots \mu_N} \sum_{j=1}^{N} \|x_j - \mu_j\|^2 + \lambda \sum_{j=2}^{N} \sum_{i<j} \kappa(x_i, x_j) \|\mu_i - \mu_j\|_p, \qquad (2)$$

where κ is a local kernel. Since κ depends only on the (fixed) data points $\{x_i\}_{i=1}^{N}$ and not on the optimization variables $\{\mu_i\}_{i=1}^{N}$, it does not change the convexity or the dimension of the problem.

Any local kernel (*e.g.,* Gaussian) can of course be used. However, from a computational point of view, it can be beneficial to use a kernel with bounded support, since this can significantly reduce the number of nonzero terms in the regularization sum. In this paper, we use a simple kNN-kernel (k nearest neighbours), *i.e.,*

$$\kappa(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \in k\text{NN}(x_j) \text{ or } x_j \in k\text{NN}(x_i), \\ 0 & \text{otherwise,} \end{cases} \qquad (3)$$

where kNN(x) is the set of x's k nearest neighbours.

4. PARTICLE FILTER OUTPUT COMPUTATION

In this section we shall study one application in which SON clustering is attractive, namely particle clustering in a PF. As mentioned in Section 1, there are several PF based methods in the literature, in which particle clustering is an essential part. It is also fairly easy to come up with new ideas on how existing methods could benefit from clustering. Take for instance the Gaussian sum PF [26], in which particles are used to fit a GMM to the filtering distribution, using

"predefined clusters". By augmenting this approach with a clustering step, the mixture becomes more adaptive, *e.g.,* allowing for a varying number of components. A similar idea could be used also with the PF presented in [27], designed for multi-rate sensor systems. However, due to space limitations, we will not pursue any of these algorithmic modifications, nor the ones discussed in Section 1, here. Instead, we illustrate the applicability of SON clustering on PF output computation, which is an interesting problem on its own. In the process, we will also propose a slight modification of the method, to make it more suitable for particle clustering.

In the PF, the posterior distribution of the state of a dynamical system is approximated by a point-mass distribution, defined by N weighted particles $\{x_t^j, w_t^j\}_{j=1}^N$, where x_t^j are the particle positions and w_t^j are the corresponding importance weights.

One of the strengths of the PF is its ability to approximate basically any distribution, and can for instance handle multimodality without problem. Multimodal posterior distributions might at first appear somewhat fictitious, but this is not the case as they arise in many realistic applications, *e.g.,* as a result of track confusion in target tracking (see also Section 4.2). Intuitively, multimodality is handled by the PF by splitting the particles into two or more groups, or clusters, each keeping track of a single mode of the distribution. Now assume that the PF is connected to some other system, or monitored by an end user. We then need to decide on some interface to the PF, *i.e.,* in what way the information available in the filter should be presented. A few common choices are,

1. Present all particles and weights. By doing so, we deliver all available information. However, if the number of particles N is high (which it typically is), this might be prohibitive due to limited communication capacities. Also, the result might be hard to interpret.
2. Compute the mean. This is probably the most common way to present an estimate derived from the PF. However, for multimodal distribution, the mean can be very misleading.
3. Compute the MAP estimate. Usually, this is done by simply extracting the one particle with the highest weight. This will focus the estimate on one of the modes, but neglects the remaining ones. Also, the particle with the highest weight can in fact be far from the true MAP [15], and can thus be a poor estimate of the state.

Here, we propose to find the modes of the distribution by particle clustering, and then deliver one estimate for each mode. More precisely, at each iteration we apply SON clustering to the particles (or rather the modified version of the method, presented below). For each cluster, we then compute a mean estimate and these estimates are delivered as output from the PF. We emphasize, once again, that the clustering step can be seen as an "add-on" to the PF, which is why we do not provide any details on the PF.

4.1. Including the weights

In the PF setting, we are concerned with a weighted particle system $\{x_t^j, w_t^j\}_{j=1}^N$. The weights w_t^j can be seen as measures of how "important" the corresponding samples, or particles, x_t^j are. It is natural to incorporate the weighting in the clustering method. This is done by replacing the sum-of-squares term in the criterion with a weighted sum-of-squares, *i.e.,* instead of (2) we use,

$$\min_{\mu_t^1 \cdots \mu_t^N} \sum_{j=1}^{N} w_t^j \|x_t^j - \mu_t^j\|^2 + \lambda \sum_{j=2}^{N} \sum_{i<j} \kappa(x_t^i, x_t^j) \|\mu_t^i - \mu_t^j\|_p. \tag{4}$$

In the example presented below, the above criterion will be used when referring to SON clustering. Similarly, since we compare SON clustering with k-means clustering, we use a (straightforwardly) modified version of k-means clustering which aims at minimizing the weighted within-cluster sum-of-squares.

4.2. Output computation in a target tracking example

To study the applicability of SON clustering for PF output computation we consider a target tracking example, in which we seek to track a vehicle moving in an urban environment. The position of the vehicle is measured using a range only sensor. The tracker is also supported by a road network, and tracking is done under the hypothesis that the vehicle always stays on the road (see *e.g.,* [28]). Hence, at each instant we know that the target is on one of the roads, but due to the rather uniformative measurement it is not always possible to determine which road. This ambiguity will cause multimodality in the posterior distribution. A bootstrap PF [14], using $N = 1000$ particles, is used to track the target.

Figure 1 shows three snapshots of the horizontal position of the vehicle, as seen from above. The particles, which is the trackers internal representation of the posterior distribution of the vehicle position, are shown as small dots. The top row of the figure shows the result from SON clustering applied to the particles, with $\lambda = 0.1$ and using a kNN-kernel with $k = 10$. The bottom row shows similar results, using k-means clustering with $k = 5$ clusters. The cluster centers, or mode estimates, are shown as circles and the total particle weight sum for each cluster is given next to the cluster. It should be emphasized that the individual particle weights are not visible in the figure, but the weights will indeed influence the clustering as discussed in Section 4.1. At $t = 30$, the rightmost cluster(s) have very low total weight. From a visualization point of view we could, of course, have chosen discard clusters with a total weight below some threshold. As can be seen from the plots, k-means uses a static number of clusters, even when the underlying distribution is unimodal. The mean and MAP estimates are not displayed in Figure 1, to avoid cluttering of the plots. However, it is not hard to see that the mean estimate, for instance, would fail at time $t = 15$ when the distribution is clearly bimodal.

5. CONCLUSION

We have proposed a novel clustering method, SON clustering, formulated as a convex optimization problem. The method does not require the number of clusters to be specified a priori. Instead, a regularization parameter λ is used to control the trade-off between model fit and the number of clusters. This feature gives the method the ability to dynamically adapt the number of clusters, *e.g.,* if it is applied to sequential data with a varying number of clusters.

This property was illustrated on a target tracking example, where the particles in a particle filter (PF) were clustered to find the modes of the particle distribution. SON clustering was compared with the common k-means clustering method. The latter produces a static number of clusters, even when the modality of the distribution changes. There are of course methods for estimating the number of clusters from the data. Such methods could be used together with k-mean to allow for a varying number of clusters. However, this would add an extra model order selection step, which needs to be carried out each time the clustering is to be performed. In this paper, we have illustrated that SON clustering has the *ability* to automatically adapt the number of clusters, for the same value of λ. However, how well this generalizes to other sequential clustering problems is a topic for further research, as it is not clear that the same value for λ gives the "best" result at each time step in the sequence.

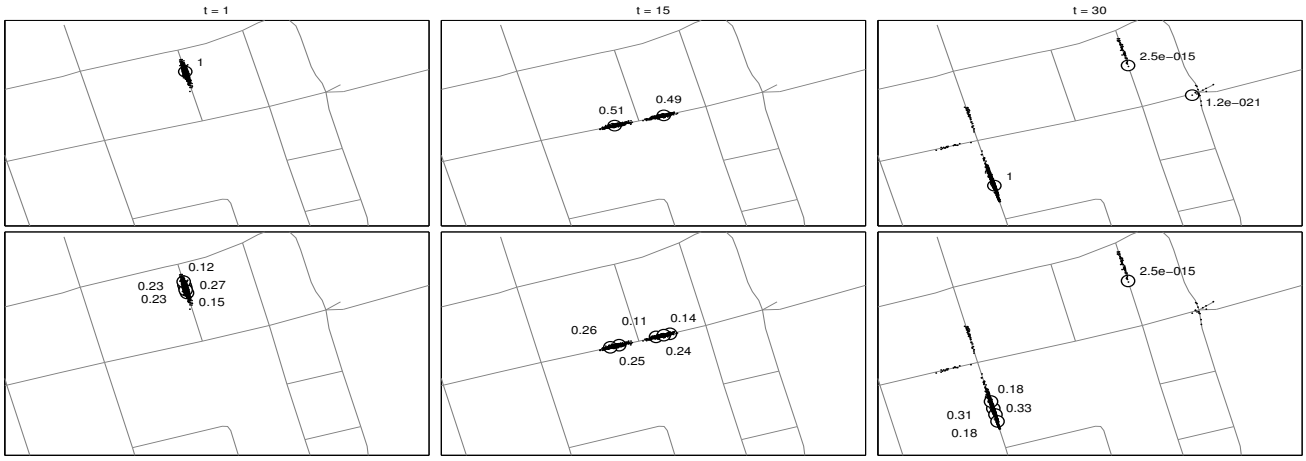

Fig. 1. Three snapshots of a two dimensional urban tracking scenario. The tracker is supported by a road network, shown in the background. The particles are shown as small black dots and the circles shows the cluster centroids. Next to each centroid is the total weight sum of the corresponding cluster. The top row gives the results from using SON clustering, and the bottom row using k-means clustering.

We have studied the applicability of SON clustering for particle clustering on the above mentioned output computation example. However, there are many PF based methods in the literature, in which clustering is an essential part. The intention of this paper has been to present SON clustering as a potential tool to be used together with these methods, as well as with future applications of clustering in a PF setting.

6. ACKNOWLEDGEMENT

The authors would like to thank Lic. Per Skoglar for supplying the data used in the target tracking example of Section 4.2. This work was supported by CADICS, a Linneaus Center funded by be Swedish Research Council.

7. REFERENCES

[1] R. Xu and II Wunsch, D., "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[2] H. Steinhaus, "Sur la division des corp materiels en parties," *Bull. Acad. Polon. Sci*, vol. 1, pp. 801–804, 1956.

[3] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[4] J. M. Peña, J. A. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern Recognition Letters*, vol. 20, pp. 1027–1040, Oct. 1999.

[5] D. Lashkari and P. Golland, "Convex clustering with exemplar-based models," in *Advances in Neural Information Processing Systems 20*, J.C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., pp. 825–832. MIT Press, Cambridge, MA, 2008.

[6] S. Nowozin and G. Bakir, "A decoupled approach to exemplar-based unsupervised learning," in *Proceedings of the 25th international conference on Machine learning*, New York, NY, USA, 2008, ICML '08, pp. 704–711, ACM.

[7] F. Lindsten, H. Ohlsson, and L. Ljung, "Just relax and come clustering! A convexification of k-means clustering," Tech. Rep. LiTH-ISY-R-2992, Department of Electrical Engineering, Linköping University, Linköping, Sweden, Feb. 2011.

[8] K. Pelckmans, J. De Brabanter, J.A.K. Suykens, and B. De Moor, "Convex clustering shrinkage," in *Statistics and Optimization of Clustering Workshop (PASCAL)*, London, U.K., July 2005, Lirias number: 181608.

[9] R. Tibsharani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[11] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society, Series B*, vol. 68, pp. 49–67, 2006.

[12] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[13] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of ARX-models using sum-of-norms regularization," *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.

[14] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107 –113, Apr. 1993.

[15] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[16] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005, pp. 181–188.

[17] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[18] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Whiley Series in Probability and Statistics. John Wiley & Sons, New York, USA, second edition, 2008.

[19] A. Milstein, J. N. Sánchez, and E. T. Williamson, "Robust global localization using clustered particle filtering," in *Proceedings of the 18th national conference on Artificial intelligence*, 2002.

[20] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multi-modality through mixture tracking," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1110–1116.

[21] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," `http://cvxr.com/cvx`, Aug. 2010.

[22] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. D. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited, 2008, `http://stanford.edu/~boyd/graph_dcp.html`.

[23] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[24] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[25] J. Roll, "Piecewise linear solution paths with application to direct weight optimization," *Automatica*, vol. 44, pp. 2745–2753, 2008.

[26] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.

[27] T. B. Schön, D. Törnqvist, and F. Gustafsson, "Fast particle filters for multi-rate sensors," in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007.

[28] Per Skoglar, Umut Orguner, David Törnqvist, and Fredrik Gustafsson, "Road target tracking with an approximative Rao-Blackwellized Particle filter," in *Proceedings of the 12th International Conference on Information Fusion*, 2009.