Applied Mathematics Letters 33 (2014) 29–34

Contents lists available at ScienceDirect

Applied Mathematics Letters

journal homepage: www.elsevier.com/locate/aml

Research announcement

Diffuse interface methods for multiclass segmentation of high-dimensional data

Ekaterina Merkurjev [a,*], Cristina Garcia-Cardona [b], Andrea L. Bertozzi [a], Arjuna Flenner [c], Allon G. Percus [b]

[a] Department of Mathematics, UCLA, Los Angeles, CA 90095, United States
[b] Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA 91711, United States
[c] Naval Air Warfare Center, China Lake, CA 93555, United States

ARTICLE INFO

Article history:
Received 20 February 2014
Accepted 21 February 2014
Available online 4 March 2014

Keywords:
Segmentation
Graphs
Ginzburg–Landau functional
MBO scheme
Convex splitting

ABSTRACT

We present two graph-based algorithms for multiclass segmentation of high-dimensional data, motivated by the binary diffuse interface model. One algorithm generalizes Ginzburg–Landau (GL) functional minimization on graphs to the Gibbs simplex. The other algorithm uses a reduction of GL minimization, based on the Merriman–Bence–Osher scheme for motion by mean curvature. These yield accurate and efficient algorithms for semi-supervised learning. Our algorithms outperform existing methods, including supervised learning approaches, on the benchmark datasets that we used. We refer to Garcia-Cardona (2014) for a more detailed illustration of the methods, as well as different experimental examples.

1. Introduction

Multiclass segmentation is a well studied problem in machine learning and computer vision. An energy formulated for the problem is often made up of a regularization term and a fidelity term. Recently, graph-based regularization terms have been used to take into account the similarities in the dataset. We present two graph-based segmentation procedures inspired by [1], which describes a binary segmentation method consisting of minimizing the Ginzburg–Landau functional on graphs using gradient descent. The first method extends this procedure to a multiclass case using the idea of the Gibbs simplex. The second method applies the simplex idea to the graph-based Merriman–Bence–Osher (MBO) scheme developed in [2]. Both algorithms result in efficient and accurate ways to perform multiclass segmentation. A more detailed explanation of the methods, intended for a computer science audience, is contained in [3], while here we follow an applied mathematics approach. We also present different results using other benchmark datasets than the ones in [3].

1.1. Ginzburg–Landau functional and diffuse interface model

The Ginzburg–Landau (GL) functional, originally proposed to describe physical phenomena, is formulated as:

$$GL(u) = \frac{\epsilon}{2}\int |\nabla u|^2 dx + \frac{1}{\epsilon}\int \Phi(u)dx, \tag{1}$$

* Corresponding author. Tel.: +1 3104902193.
E-mail addresses: kmerkurjev@gmail.com, kmerkurev@math.ucla.edu (E. Merkurjev), cristina.cgarcia@gmail.com (C. Garcia-Cardona), bertozzi@math.ucla.edu (A.L. Bertozzi), allon.percus@cgu.edu (A.G. Percus).

http://dx.doi.org/10.1016/j.aml.2014.02.008
0893-9659/

where u denotes the state of the phases, ∇ represents the spatial gradient operator, $\Phi(u)$ is a double-well potential, such as $\frac{1}{4}(u^2-1)^2$, and ϵ is a positive constant.

Kohn et al. show in [4] that the $\epsilon \to 0$ limit of (1), in the sense of Γ-convergence, is the total variation semi-norm:

$$GL(u) \to_{\Gamma} \|u\|_{TV}. \tag{2}$$

One might prefer to use the GL functional in data segmentation because its L_2 gradient flow results in a linear differential operator. By contrast, the TV semi-norm contains a nonlinear curvature term.

The diffuse interface description has been used successfully in image inpainting [5,6] and image segmentation [7]. The standard practice is to introduce an additional fidelity term F to allow for the specification of any known information $\hat{u}$:

$$E(u) = GL(u) + F(u, \hat{u}). \tag{3}$$

One way to minimize the energy is using gradient descent, producing the modified Allen–Cahn equation:

$$\frac{\partial u}{\partial t} = -\frac{\delta GL}{\delta u} - \mu\frac{\delta F}{\delta u} = \epsilon\Delta u - \frac{1}{\epsilon}\Phi'(u) - \mu\frac{\delta F}{\delta u}. \tag{4}$$

1.1.1. Graph framework for large datasets

In [1], Bertozzi and Flenner outline an approach for binary segmentation using the Ginzburg–Landau functional in a graph domain instead of the continuous domain of the original functional.

We consider a graph setting where $G = (V, E)$ is an undirected graph with vertices V and edges E. For each dataset, the vertices represent its building blocks; for example, in the case of an image, the vertices would consist of pixels. Each edge contains a value $w(i, j)$ assigned to it, measuring the similarity between the two vertices i and j it is connecting. In this work, we use the Gaussian function $w(i, j) = \exp(-d(i, j)^2/\sigma^2)$ as a similarity measure. Here $d(i, j)$ represents some measure of "distance" between vertices i and j.

If one defines $\mathbf{W}$ as the matrix $W_{ij} = w(i, j)$, the degree of a vertex $i \in V$ as $d_i = \sum_{j\in V} w(i, j)$, and $\mathbf{D}$ to be the diagonal matrix with elements d_i, then the graph Laplacian is formulated as $\boldsymbol{L = D - W}$.

1.1.2. Ginzburg–Landau functional on graphs

We use the theory of nonlocal calculus, described in [8], to generalize the continuous GL formulation to a graphical setting. The theory relates the spacial Laplace operator to the graph Laplacian matrix of the previous section. In fact, the eigenvectors of the discrete Laplacian converge to those of the Laplacian [1]. However, in the limit of large sample size, the matrix $\mathbf{L}$ must be scaled correctly to guarantee stability of convergence to the continuum differential operator [1]. A scaling we use is the normalized Laplacian

$$\mathbf{L_s} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \tag{5}$$

because the matrix is symmetric, which makes the linear algebra routines we use more efficient.

The GL functional on graphs is formulated as

$$GL(u) = \frac{\epsilon}{2}\langle u, \mathbf{L_s}u\rangle + \frac{1}{4\epsilon}\sum_{i\in V}\left(u_i^2 - 1\right)^2, \tag{6}$$

where u_i, the ith component of vector u, is the state of node i. Here, the gradient term in the original functional (1) is replaced by a more general operator on graphs. The second term is a double-well potential function having minima at ± 1. This representation is appropriate for binary classifications only.

Note that one benefit of using a graphical framework is having a way to deal with the case of nonlinearly separable classes. In addition, using graphs provides a straightforward way of processing high dimensional data.

1.1.3. Semi-supervised learning (SSL) on graphs

In semi-supervised learning (SSL), one uses the knowledge of the labels of a fraction of the nodes. As in (3), an additional fidelity term is introduced to specify the information:

$$E(u) = \frac{\epsilon}{2}\langle u, \mathbf{L_s}u\rangle + \frac{1}{4\epsilon}\sum_{i\in V}\left(u_i^2 - 1\right)^2 + \sum_{i\in V}\frac{\mu_i}{2}\left(u_i - \hat{u}_i\right)^2, \tag{7}$$

where u_i, the ith component of vector u, is the (real-valued) state of node i, μ_i is a parameter that is equal to a positive constant μ if i is a fidelity node and to zero otherwise, and $\hat{u}_i$ is the known value of fidelity node i.

Thus, given an initial state u_i of each vertex i, the goal is to minimize the GL functional with fidelity term. The classes are obtained by thresholding u_i. We extend this efficient binary data segmentation method to multiclass segmentation in the following sections.

2. Multiclass Ginzburg–Landau approach

2.1. Extension to multiclass segmentation

Let N_D be the dimension of the dataset and K the number of classes. To extend to the multiclass case, we now assign to node i a composition of states $\mathbf{u}_i \in \mathbb{R}^K$, where the k^{th} component of $\mathbf{u}_i$ is the strength that the node belongs to class k. Note that in the binary case, each node was assigned a single real value denoting in some way the class membership. Also, note that $\mathbf{u}$ is now a $N_D \times K$ matrix.

We require the vector $\mathbf{u}_i$ to be an element of the Gibbs simplex Σ^K, represented as:

$$\Sigma^K := \left\{ (x_1, \ldots, x_K) \in [0,1]^K \;\middle|\; \sum_{k=1}^{K} x_k = 1 \right\}. \tag{8}$$

Vertices of the simplex are elements in which all components vanish, except one which is equal to 1. These elements represent points that belong exclusively to a certain class. Note that the simplex formulation has a straightforward probabilistic interpretation, since each row of $\mathbf{u}$ is the probability distribution over the K classes.

The multiclass GL energy functional on graphs is expressed as:

$$E(\mathbf{u}) = \frac{\epsilon}{2}\langle \mathbf{u}, \mathbf{L_s u}\rangle + \frac{1}{2\epsilon}\sum_{i\in V}\left(\prod_{k=1}^{K}\frac{1}{4}\left\|\mathbf{u}_i - \mathbf{e}_k\right\|_{L_1}^2\right) + \sum_{i\in V}\frac{\mu_i}{2}\left\|\mathbf{u}_i - \hat{\mathbf{u}}_i\right\|^2,$$

where $\hat{\mathbf{u}}_i$ is a vector indicating prior class knowledge of node i, and $\langle \mathbf{u}, \mathbf{L_s u}\rangle = \text{trace}\,(\mathbf{u}^T\mathbf{L_s u})$.

The advantage of the simplex representation is that, unlike the case with some multiclass methods, the penalty assigned to neighbors that are differently labeled is independent of the labels. Note that in the second term of the energy, we use an L_1 norm as it prevents an undesirable minimum from occurring at the center of the simplex, as would be the case with an L_2 norm for large K. This potential aims to provide a clear way to calculate class memberships, as the phase composition is purer near the vertices of the simplex. The third (fidelity) term allows the user to input any *a priori* information.

2.2. Energy minimization

Similar to the procedure in [1], we use a convex splitting scheme for the minimization of the energy:

$$\begin{aligned}
&E(\mathbf{u}) = E_{\text{convex}}(\mathbf{u}) + E_{\text{concave}}(\mathbf{u})\\
&E_{\text{convex}}(\mathbf{u}) = \frac{\epsilon}{2}\langle \mathbf{u}, \mathbf{L_s u}\rangle + \frac{C}{2}\langle \mathbf{u}, \mathbf{u}\rangle\\
&E_{\text{concave}}(\mathbf{u}) = \frac{1}{2\epsilon}\sum_{i\in V}\prod_{k=1}^{K}\frac{1}{4}\left\|\mathbf{u}_i - \mathbf{e}_k\right\|_{L_1}^2 + \sum_{i\in V}\frac{\mu_i}{2}\left\|\mathbf{u}_i - \hat{\mathbf{u}}_i\right\|_{L_2}^2 - \frac{C}{2}\langle \mathbf{u}, \mathbf{u}\rangle.
\end{aligned}$$

Here $C \in \mathbb{R}$ is a constant big enough so the concavity/convexity of the above terms is valid. Under the right conditions, this approach results in an unconditionally stable scheme for gradient descent [7,9,10] of the form

$$\mathbf{u}^{n+1} + dt\,\frac{\delta E_{\text{convex}}}{\delta \mathbf{u}}(\mathbf{u}^{n+1}) = \mathbf{u}^n - dt\,\frac{\delta E_{\text{concave}}}{\delta \mathbf{u}}(\mathbf{u}^n). \tag{9}$$

We use an implicit scheme because of the stiffness of the differential equations. Otherwise, dt might be forced to be extremely small for any meaningful solution. We solve the scheme very efficiently using spectral methods for which only a small number of eigenfunctions are needed to obtain a good result. Note that after the update, the solution might no longer be an element of the Gibbs simplex, so we use the procedure in [11] to project the phase field back to the simplex. For initialization, we use a random point on the simplex for a non-fidelity node, and the corresponding vertex on the simplex for a fidelity node.

The stopping criterion we use for energy minimization is

$$\frac{\max_i \|\mathbf{u}_i^{n+1} - \mathbf{u}_i^n\|^2}{\max_i \|\mathbf{u}_i^{n+1}\|^2} < \eta. \tag{10}$$

Finally, we assign node i to class k if $\mathbf{u}_i$ is closest to vertex $\mathbf{e}_k$ on the simplex.

Note that other operator splitting methods have been studied for minimization problems (e.g. [12]). Ours however has the advantages: (i) it is direct and thus does not require one to solve further minimization problems, (ii) one can adjust the accuracy by changing the number of eigenfunctions, and (iii) its complexity is close to linear in $|V|$. Details about (iii) can be found in [3].

3. MBO reduction of the Ginzburg–Landau energy functional

In [13], Merriman, Bence and Osher describe a simple algorithm to approximate motion by mean curvature, or motion in which normal velocity equals mean curvature. Their algorithm consists of alternating between the following steps:

1. *Diffusion.* Let $u^{n+\frac{1}{2}} = S(\delta t)u^n$ where $S(\delta t)$ is the propagator (by time δt) of the standard heat equation: $\frac{\partial u}{\partial t} = \Delta u$.
2. *Thresholding.* Let

$$u^{n+1} = \begin{cases} 1 & \text{if } u^{n+\frac{1}{2}} \geq 0, \\ -1 & \text{if } u^{n+\frac{1}{2}} < 0. \end{cases}$$

Our interest in this algorithm comes from its relation to the basic (unmodified) Allen–Cahn equation:

$$\frac{\partial u}{\partial t} = \epsilon \Delta u - \frac{1}{\epsilon} \Phi'(u). \tag{11}$$

One can use a simple time-splitting scheme, in which one first propagates using the first term of the right side of (11) and then the second one, to evolve the Allen–Cahn equation. Note, however, that in the $\epsilon \to 0$ limit, the second step is simply thresholding [13]. Thus, as $\epsilon \to 0$, this time splitting scheme amounts to alternating between diffusion and thresholding steps, exactly the MBO scheme. Moreover, [14] shows that, as $\epsilon \to 0$, the rescaled solutions $u_\epsilon(z, t/\epsilon)$ of (11) yield motion by mean curvature of the interface between the two phases of the solutions.

Barles [15] and Evans [16] have proven rigorously that the MBO scheme approximates motion by mean curvature.

3.1. Graph formulation

The motion by mean curvature of the MBO scheme was generalized to graphs by Merkurjev et al. in [2], where a modified MBO scheme on graphs has been used to formulate an algorithm for binary classification and image processing. The authors apply a two-step time splitting scheme to (4) so that the second step is the same as the one in the original MBO scheme, and then replace the Δu term with a more general graph term $-\mathbf{L_s}u$. The result is a scheme that alternates between propagation using the modified Allen–Cahn Eq. (4) on graphs and thresholding.

3.2. Extension to multiclass segmentation

The standard Gibbs-simplex Σ^K defined in Eq. (8) allows the multiclass extension of the algorithm in [2] to be straightforward. Step 2 of the method above is modified so that the thresholding now takes the form of a displacement towards the vertex in the Gibbs simplex closest to the projection of the result of the diffusion step using the model in [11]. In summary, using the same notation as in Section 2, the new algorithm consists of alternating between:

1. Heat equation with forcing term:

$$\frac{\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}^n}{dt} = -\mathbf{L_s}\mathbf{u}^{n+\frac{1}{2}} - \boldsymbol{\mu}(\mathbf{u}^n - \hat{\mathbf{u}}). \tag{12}$$

2. Thresholding:

$$\mathbf{u}_i^{n+1} = \mathbf{e}_k, \tag{13}$$

where vertex $\mathbf{e}_k$ is the vertex in the simplex closest to the projection of $\mathbf{u}_i^{n+\frac{1}{2}}$ using [11].

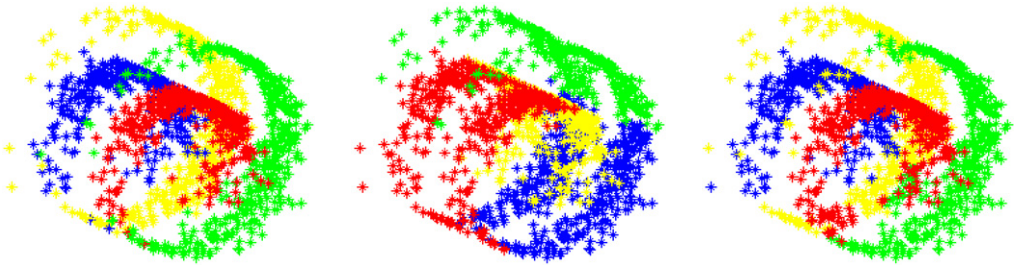
The scheme is solved using spectral methods. The initialization procedure and the stopping criterion are the same as in Section 2. In practice, the diffusion step can be repeated several times, denoted by N_S, before the thresholding step. Finally, node i is assigned to class k if the kth component of $\mathbf{u}_i$ is one.

In practice, the complexity of this algorithm is close to linear as well [3].

4. Numerical results

4.1. Previous results for MNIST, WebKB, COIL and three moons datasets

We first summarize the results in [3], which contains a more detailed description of our algorithms. The main datasets used in the paper were the WebKB [17], MNIST [18] and COIL [19] benchmark datasets. For WebKB, multiclass MBO and GL achieved an accuracy of 88.48% and 87.2%, respectively. This was compared with supervised learning methods reported in [20], such as centroid-normalized sum (82.66%), naive Bayes (83.52%) and SVM (linear kernel) (85.82%). For these methods, two thirds of the data was used for training, and one third for testing. Our methods obtain higher accuracy using only



(a) Swiss roll dataset. (b) Spectral clustering result (49%). (c) Our method's result (94.5%).

Fig. 1. Swiss roll dataset results.

20%—25% fidelity. For the MNIST dataset, the accuracy of the two methods was 96.91% and 96.8%, respectively, which was comparable to the results obtained by neutral/convolutional nets, boosted stumps and nonlinear classifiers, and higher than that of methods such as transductive classification and Cheeger cuts. For the COIL dataset, multiclass MBO and GL were able to achieve higher accuracy (91.46% and 91.2%, respectively) when compared to some of the best methods, like MP, SQ-Loss-I and sGT, mentioned in [21]. The paper also presents a co-segmentation example using an image, as well as results for the synthetic "three moons" dataset. For the latter, the results were 99.12% and 98.1% for multiclass MBO and GL methods, respectively.

4.2. Swiss roll

The Swiss roll dataset, pictured in Fig. 1a, contains 1600 3*D* points arranged in spirals. To calculate the weight matrix, we used the weight function in [22] with 10 nearest neighbors. To calculate the eigenvectors, we used the procedure in [23]. The parameters used were: 50 eigenvectors, $C = 51$, $\epsilon = 1$, $dt = 0.1$, $\mu = 50$ and $\eta = 10^{-7}$ with 80 fidelity points (5%).

Averaged over 100 runs, the average accuracies obtained were 94.1% and 91.8% for the multiclass GL and MBO methods, respectively. The visual result of the latter method is included in Fig. 1c. We compare this result to the one obtained using spectral clustering (Fig. 1b): average accuracy was only 49.75% over 100 runs, with the graph being the same as for multiclass GL and MBO cases. Calculations were done with 4 eigenvectors.

4.3. Landsat satellite dataset

This database contains multi-spectral values of pixels in neighborhoods in a satellite image. To calculate the weight matrix, we used the weight function in [22] with 30 nearest neighbors. To calculate the eigenvectors, we used the procedure in [23]. The parameters used were: 200 eigenvectors, $C = 51$, $\epsilon = 1$, $dt = 0.1$, $\mu = 50$ and $\eta = 10^{-7}$ with 350 fidelity points (5.44%).

Averaged over 30 runs, the average accuracies obtained were 87.05% and 87.25% for the multiclass GL and MBO methods, respectively. We compare these results to several supervised learning methods listed in [24]. These algorithms were performed using 80% training and 20% validation. The results were: 65.15% using SC-SVM, 75.43% using SH-SVM, 65.88% using S-LS, 86.65% using simplex boosting, and 90.15% using S-LS rbf. We outperform all but the last algorithm using only 5% fidelity, while these algorithms use 80% for training.

4.4. Human activity dataset

This dataset from the UCI Machine Learning Repository contains information about experiments carried out with 30 volunteers. Each person performed one of six actions: walking, walking upstairs, walking downstairs, sitting, standing and laying while wearing a smartphone on the waist. The smartphone recorded their linear acceleration and 3-axial angular velocity. The goal is to segment the people into six classes according to activity using the information obtained from the phone. We used the weight function in [22] with 59 nearest neighbors. The parameters used were: 50 eigenvectors, $C = 160$, $dt = 0.31$, $\mu = 159$ and $\eta = 10^{-7}$ with 5% of points being fidelity points.

The average accuracy was 89.7% for the multiclass MBO method, while being 88.7% for the GL method. We compare this to the results of [25], where the methods MC-SVM and MC-HF-SVM have accuracies of 89.3% and 89.0%, respectively. It is important to note that the results of the paper were obtained using supervised learning methods where 70% of the data was used for training and the rest for testing. We obtain higher accuracy (with multiclass MBO) using only 5% fidelity.

Acknowledgments

The authors would like to thank Chris Anderson for the code of the procedure of [23]. This work was supported by ONR grants N000141210838, N000141210040, N0001413WX20136, AFOSR MURI grant FA9550-10-1-0569, NSF grants DMS-1118971, DMS-0914856, and the Keck Foundation. Ekaterina Merkurjev is also supported by an NSF graduate fellowship.

References

[1] A. Bertozzi, A. Flenner, Diffuse interface models on graphs for classification of high dimensional data, Multiscale Model. Simul. 10 (2012) 1090–1118.
[2] E. Merkurjev, T. Kostic, A.L. Bertozzi, An MBO scheme on graphs for classification and image processing, SIAM J. Imaging Sci. 6 (2013) 1903–1930.
[3] C. Garcia-Cardona, E. Merkurjev, A. Bertozzi, A. Flenner, A. Percus, Fast multiclass segmentation using diffuse interface methods on graphs, IEEE Trans. Pattern Anal. Mach. Intell. (2014).
[4] R. Kohn, P. Sternberg, Local minimizers and singular perturbations, Proc. Roy. Soc. Edinburgh Sect. A 111 (1989) 69–84.
[5] A. Bertozzi, S. Esedoğlu, A. Gillette, Impainting of binary images using the Cahn–Hilliard equation, IEEE Transactions on Image Processing 16 (2007) 285–291.
[6] J.A. Dobrosotskaya, A.L. Bertozzi, A wavelet-Laplace variational technique for image deconvolution and inpainting, IEEE Trans. Image Process. 17 (2008) 657–663.
[7] S. Esedoğlu, Y. Tsai, Threshold dynamics for the piecewise constant Mumford–Shah functional, J. Comput. Phys. 211 (2006) 367–384.
[8] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, Multiscale Model. Simul. 7 (2008) 1005–1028.
[9] D.J. Eyre, An unconditionally stable one-step scheme for gradient systems, 1998. http://www.math.utah.edu/~eyre/research/methods/papers.html.
[10] A.L. Yuille, A. Rangarajan, The concave–convex procedure (CCCP), Neural Comput. 15 (2003) 915–936.
[11] Y. Chen, X. Ye, Projection onto a simplex, (2011) arXiv preprint arXiv:1101.6081.
[12] J. Lellmann, J.H. Kappes, J. Yuan, F. Becker, C. Schnörr, Convex Multi-Class Image Labeling by Simplex-Constrained Total Variation, Technical Report, IWR, University of Heidelberg, 2008.
[13] B. Merriman, J.K. Bence, S.J. Osher, Motion of multiple functions: a level set approach, J. Comput. Phys. 112 (1994) 334–363.
[14] J. Rubinstein, P. Sternberg, J. Keller, A simple proof of convergence for an approximation scheme for computing motions by mean curvature, SIAM J. Appl. Math. 49 (1989) 116–133.
[15] G. Barles, C. Georgelin, A simple proof of convergence for an approximation scheme for computing motions by mean curvature, SIAM J. Numer. Anal. 32 (1995) 484–500.
[16] L.C. Evans, Convergence of an algorithm for mean curvature motion, Indiana Univ. Math. J. 42 (1993) 533–557.
[17] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, Learning to extract symbolic knowledge from the world wide web, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), AAAI Press, pp. 509–516.
[18] Y. LeCun, C. Cortes, The MNIST database of handwritten digits, 1998. http://yann.lecun.com/exdb/mnist/.
[19] S. Nene, S. Nayar, H. Murase, Columbia Object Image Library (COIL-100), Technical Report CUCS-006-96, 1996.
[20] A. Cardoso, Datasets for single-label text categorization, 2007. http://web.ist.utl.pt/~acardoso/datasets/.
[21] A. Subramanya, J. Bilmes, Semi-supervised learning with measure propagation, J. Mach. Learn. Res. 12 (2011) 3311–3370.
[22] P. Perona, L. Zelnik-Manor, Self-tuning spectral clustering, Adv. Neural Inf. Process. Syst. 17 (2004) 1601–1608.
[23] C. Anderson, A Rayleigh–Chebyshev procedure for finding the smallest eigenvalues and associated eigenvectors of large sparse Hermitian matrices, J. Comput. Phys. 229 (2010) 7477–7487.
[24] Y. Mroueh, T. Poggio, L. Rosasco, J.-J.E. Slotine, Multi-class Learning: With Simplex Coding, 2011.
[25] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in: Ambient Assisted Living and Home Care, Springer, 2012, pp. 216–223.