Geometric PDEs on weighted graphs for semi-supervised classification
Matthieu Toutain, Abderrahim Elmoataz, Olivier Lézoray

To cite this version:
Matthieu Toutain, Abderrahim Elmoataz, Olivier Lézoray. Geometric PDEs on weighted graphs for semi-supervised classification. International Conference on Machine Learning and Applications, IEEE, 2012, Detroit, United States. hal-01108860

HAL Id: hal-01108860
https://hal.science/hal-01108860
Submitted on 23 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—In this paper, we consider the adaptation of two Partial Differential Equations (PDEs) on weighted graphs, $p$-Laplacian and eikonal equations, for semi-supervised classification tasks. These equations are a discrete analogue of well known geometric PDEs, which are widely used in image processing. While the $p$-Laplacian on graphs was intensively used in data classification, few works relate to the eikonal equation for data classification. The methods are illustrated through semi-supervised classification tasks on databases, where we compare the two algorithms. The results show that these methods perform well regarding the state-of-the-art and are applicable to the task of semi-supervised classification.

I. INTRODUCTION

Nowadays, more and more applications involve manipulating images or digital data defined on complex or high dimension manifolds, such as data collected as graphs, networks, or discrete data. Partial Differential Equations (PDEs) provide a major mathematical tool in image processing, and are involved in the resolution of many problems such as image segmentation, object detection, image restoration or denoising, and so on. There is actually much interest for adapting ideas and methods from the image processing literature on graphs to define new clustering and classification algorithms [1], [2], [3], [4], [5]. In this paper, we focus on the use of our recently proposed framework of discrete operators and geometric PDEs for front propagation and particularly the eikonal equation, and we show that it can be applied to perform semi-supervised classification tasks [6], [7]. In the sequel, we will call Partial difference Equations (PDEs) the adaptation on weighted graphs of PDEs. We compare it to our previously introduced graph based regularization framework, which has already been used for data classification [8], and to one of the state-of-the-art method [1]. This paper is organized as follows. Section 2 provides notations and basics on graphs, and our previously introduced framework of discrete operators on graphs. Section 3 presents the extension of the PDE based front propagation concept to graph of arbitrary topologies [9]. Section 4 recalls our previously introduced regularization framework in order to provide an algorithm of label diffusion [8]. In section 5, we show that these two previously described algorithms, originally designed for image processing tasks, can be applied to semi-supervised classification [10], experimenting them on standard databases and a database for diagnosis in pathology.

II. PRELIMINARIES ON GRAPH AND NOTATIONS

As the core structure of our approach, in this section we provide notations and basics on weighted graphs, recall our formulations of difference, morphological differences, and gradients on weighted graphs [11], [3]. This section is a required preliminary to fully understand the different operators defined on weighted graphs that will be introduced in the next sections.

Notations: A weighted graph $G = (V, E, w)$ consists in a finite set $V = \{v_1, \ldots, v_N\}$ of $N$ vertices and a finite set $E = \{e_1, \ldots, e_N\} \subset V \times V$ of $N'$ weighted edges. We assume $G$ to be undirected, with no self-loops and no multiple edges. Let $e_{ij} = (v_i, v_j)$ be the edge of $E$ that connects vertices $v_i$ and $v_j$ of $V$. Its weight, denoted by $w_{ij} = w(v_i, v_j)$, represents the similarity between its vertices. Similarities are usually computed by using a positive symmetric function $w: V \times V \to \mathbb{R}^+$ satisfying $w_{ij} = 0$ if $(v_i, v_j) \notin E$. The notation $v_i \sim v_j$ is also used to denote two adjacent vertices. We say that $G$ is connected whenever, for any pair of vertices $(v_k, v_l)$ there is a finite sequence $v_k = v_0, v_1, \ldots, v_n = v_l$ such that $v_{i-1}$ is a neighbor of $v_i$ for every $i \in \{1, \ldots, n\}$. Let $H(V)$ be the Hilbert space of real-valued functions defined on the vertices of a graph. A function $f: V \to \mathbb{R}$ of $H(V)$ assigns a real value $x_i = f(v_i)$ to each vertex $v_i \in V$. Clearly, a function $f \in H(V)$ can be represented by a column vector in $\mathbb{R}^{|V|}$: $[x_1, \ldots, x_N]^T = [f(v_1), \ldots, f(v_N)]^T$.

Operators on weighted graphs: Let $G = (V, E, w)$ be a weighted graph, and let $f: V \to \mathbb{R}$ be a function of $H(V)$. The difference operator, or directional derivative [11] of $f$, noted $d_w: H(V) \to H(E)$, is defined on an edge $e_{ij} \in E$ by:

$$(d_w f)(e_{ij}) = (d_w f)(v_i, v_j) = w_{ij}^{1/2}(f(v_j) - f(v_i))$$

(1)

The weighted gradient operator of a function $f \in H(V)$, at a vertex $v_i \in V$, is the vector operator defined by:

$$(\nabla_w f)(v_i) = [(d_w f)(v_i, v_j) : v_j \sim v_i]^T$$

(2)

The $L_p$ norm of this vector represents the local variation of the function $f$ at a vertex of the graph. It is defined by [11]:

$$\|\nabla_w f(v_i)\|_p = \left[ \sum_{v_j \sim v_i} w_{ij}^{p/2} |f(v_j) - f(v_i)|^p \right]^{1/p}$$

(3)
Based on the difference operator definition, two directional difference operators are defined. The weighted directional external and internal difference operators have been introduced in [3] as \( (d_{w}^{+}) : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{E}) \), by
\[
(d_{w}^{+}f)(v_i, v_j) = w_{ij}^{1/2} (f(v_j) - f(v_i))^{\pm}
\]
with the following notations: \((x)^{+} = \max(x, 0)\) and \((x)^{-} = -\min(x, 0)\). Similarly, the weighted morphological internal and external gradients at a vertex \(v_i\) are expressed as
\[
(\nabla_{w}^{\pm}f)(v_i) = [(d_{w}^{+}f)(v_i, v_j) : v_j \sim v_i]^T.
\]

III. PDE-based Geometric Diffusion

In this section we provide details and recalls on PdE based morphology to introduce our methods, which is based on the previous definitions provided on graphs [3]. We then establish a link with the static version of the eikonal equation (for further details, see [9]). This method permits to perform segmentation and data classification over graphs within the same framework.

**PDEs-based Morphology:** Discrete dilation and erosion on weighted graphs are defined by
\[
\partial_t f(v_i) = +|| (\nabla_{w}^{+}f)(v_i)||_p \quad \text{and} \quad \partial_t f(v_i) = -|| (\nabla_{w}^{-}f)(v_i)||_p,
\] respectively, with the notation \(\partial_t f = \frac{\partial f}{\partial t}\). These equations (7) constitute a PDEs based framework that extends algebraic and continuous morphological operators to graphs. For further details, see [3].

**PDE-based level-set and eikonal equation:** The level-set formulation has been introduced by Osher and Sethian [12] in order to describe the evolution of a parametrized curve evolving on a domain. Once formulated as a level-set problem, the curve evolution amounts to solving the following equation:

\[
\begin{cases}
\partial_t \phi(x, t) = \mathcal{F}(x, t) \nabla \phi, \\
\phi(x, 0) = \phi_0(x),
\end{cases}
\]
where \(\phi_0(x)\) is an implicit function that represents the initial front, and \(\phi(x, t)\) is an implicit function representing the front at time \(t\). \(\mathcal{F}\) is a controlling force representing the motion of the front. In [9], a transcription of (8) has been proposed to weighted graphs, that can be expressed as a morphological process with the following sum of dilation and erosion:
\[
\partial_t \phi(v_i) = \mathcal{F}^{+}(v_i)|| (\nabla_{w}^{+}\phi)(v_i)|| + \mathcal{F}^{-}(v_i)|| (\nabla_{w}^{-}\phi)(v_i)||,
\]
where \(\mathcal{F} \in \mathcal{H}(\mathcal{V})\) controls the front propagation. Such a formulation enables recovery of geometric diffusion models such as mean curvature motion, active contours, or a graph based transcription of the eikonal equation. The eikonal equation is also a very popular equation in computer graphics and computer vision and is involved in many applications. Numerous methods have been proposed to solve it on Cartesian grids and some particular non-Cartesian domains (see [9] and references therein). Recently, two adaptations of the eikonal equation have been proposed, first as a time dependent version [3], then as a static version [9] which is expressed as
\[
\begin{cases}
|| (\nabla_{w}^{+}f)(v_i)||_p = P(v_i), & \forall v_i \in \mathcal{V}, \\
f(v_i) = 0, & \forall v_i \in \mathcal{V}_0,
\end{cases}
\]
where \(\mathcal{V}_0 \subset \mathcal{V}\) corresponds to the initial set of seeds vertices, and \(P\) is a potential. Such adaptations are expressed using the PdEs based morphological erosion, and can be linked with the general geometric PdE equation (9).

**Label propagation:** In [9], the following label propagation algorithm has been proposed based on the resolution of (10), that enables the propagation of many labels on a graph:

1. **List of variables:**
2. \(S^0\) : the set of seed vertices.
3. \(A\) : the set of active vertices.
4. \(NB\) : the set of vertices in the narrow band.
5. \(FA\) : the set of vertices said as far away.
6. \(lab\) : the label indicator function.
7. **Initialization:**
8. \(lab(v_i) = \) Initial label of \(v_i\).
9. \(f(v_i) = 0 \forall v_i \in S^0\); \(f(v_i) = +\infty \forall v_i \in V \setminus S^0\).
10. \(s(v_i) = +\infty \forall v_i \in V \setminus S^0\).
11. \(A = S^0 \cup NB\).
12. \(FA = V \setminus A \cup NB\).
13. **Process:**
14. while \(FA \neq \emptyset\) do
15. \(v_i = \) first element of \(NB\).
16. remove \(v_i\) from \(NB\) and add \(v_i\) to \(A\).
17. for all \(v_j \in N(v_i) \cap A\) do
18. compute local solution \(t \leftarrow f(v_j)\).
19. if \(t < f(v_j)\) then
20. \(f(v_j) = t\).
21. if \(v \in FA\) then
22. remove \(v_j\) from \(FA\).
23. end if
24. end for
25. update position of \(v_j\) in \(NB\).
26. end if
27. if \(f(v_i)/w_{ij} < s(v_j)\) then
28. \(s(v_j) = f(v_i)/w_{ij}\),
29. \(lab(v_j) = lab(v_i)\).
30. end if
31. end if
32. end for
33. end while

The propagation is performed from a set of seeded vertices and until all vertices of the graph are marked with a label. The front propagation is therefore linear in the number of vertices. This algorithm enables many applications on graphs, such as geodesic distance computation on graphs, image segmentation and, in our case, semi-supervised data classification. The local solution \(t\) (line 19 of the previous algorithm) is computed using equations (6) and (10). For \(p \in \{1, 2\}\), we get:
\[
\left(\sum_{v_i \sim v_j} w_{ij}^{p/2} \max(0, (f(v_i) - f(v_j)))^p\right) = P(u).
\]
For the sake of clarity, we present here the solution for the L₁ norm formulation (see [9] for other cases): 
\[ P = \sum_{j=1}^{n} h_j v_j + C, \]
where \( x = f(v_j) \), \( n = |v_j| \sim |v_i|, a_i = f(v_j)|v_j| \sim |v_i| \) with \( j = 1, \ldots, n \), \( h_j = 1/\sqrt{|w_j|} \), and \( C = P(v_j) \). Further details on this graph based front propagation algorithm can be found in [9]. Even if it has already been used in image processing, this formulation has not been used yet for data classification. Classically, the potential functions used in the domain of image processing are: constant potential (\( P = 1 \)) and local norm of the gradient (\( P(v_j) = \| (\nabla f(v_j)) \|_2 \)). For the case of classification, we have adapted the potential function in order to obtain compact clusters, containing elements that are close to each others (small distances within clusters), and separating elements that are not close (large distances between clusters). To achieve both this requirements, we have used a potential that is the distance to the mean of an evolving cluster: \( P(v_j) = \| f(v_j) - C_m \|_2 \), with \( C_m \), the mean inside the front \( \Gamma_m \), corresponding to the m-th cluster. The mean of each cluster is updated each time a node is added to. In the experiments section, we have used these three potential functions to perform our tests, and we show a comparison.

IV. SEMI-SUPERVISED CLASSIFICATION USING p-LAPLACIAN OPERATOR ON GRAPH

In this section we provide details on our previously introduced graph-based regularization framework [11], in order to provide an algorithm for label diffusion over a graph using regularization [8].

Regularization: From the previous definitions, we introduce here the weighted p-laplace operator of a function \( f \in \mathcal{H}(\mathcal{V}) \), noted \( \Delta_{w,p} f : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{V}) \), at a vertex \( v_i \in \mathcal{V} \), is defined by:
\[
(\Delta_{w,p} f)(v_i) = \frac{1}{2} \sum_{v_j \sim v_i} \left( \gamma_{w,p}(v_i, v_j) (f(v_i) - f(v_j)) \right),
\]
with
\[
\gamma_{w,p}(v_i, v_j) = \frac{w_{ij}}{2 \sum_{v_k \sim v_j} \| \nabla w(f(v_k)) \|_2^2} + \frac{\| \nabla w(f(v_j)) \|_2^2}{2 \sum_{v_k \sim v_i} \| \nabla w(f(v_k)) \|_2^2}.
\]

In the case where \( p = 1 \) or 2, we have the definitions of the standard graph curvature \( \Delta_{0,1} f(v_i) = k f \) and graph Laplacian \( \Delta_{0,2} f(v_i) = \Delta f \) operators. More details on these definitions can be found in [11].

To regularize a function \( f^0 : \mathcal{V} \rightarrow \mathbb{R} \) using the weighted p-Laplace operator (12), we consider the following general variational problem on graphs:
\[
\min_{f : \mathcal{V} \rightarrow \mathbb{R}} \mathcal{E}_{w,p}(f) = R_{w,p}(f).
\]

The intuition behind regularization is to provide a smoother version of an initial function \( f^0 \), \( R_{w,p}(f) \), is the regularizer and is defined as the discrete Dirichlet form of the function \( f \in \mathcal{H}(\mathcal{V}) : R_{w,p}(f) = \frac{1}{2} \sum_{v_i \sim v_j} \| \nabla w(f(v_i)) \|_2^2 \). This term is a strictly convex function of \( f \) (see [13]). To approximate the solution of the minimization problem (14), we build a system of equations that we linearize and use the Gauss-Jacobi method to obtain the following iterative algorithm (for more details, see [11]):
\[
\begin{cases}
 f(t^0)(v_i) = f^0(v_i), \\
 f(t^{i+1})(v_i) = \frac{\sum_{v_j \sim v_i} \gamma(t(v_i), v_j)(f(t^i)(v_j))}{\sum_{v_j \sim v_i} \gamma(t(v_i), v_j)},
\end{cases}
\]

where \( \gamma(t(v_i), v_j) \) is the \( t \) function (in equation (13)) at the iteration step \( t \). At each iteration of the algorithm, the value of \( f \) at step \( t+1 \), for a vertex \( v_i \), only depends on two quantities: the original value \( f^0 \) and the sum of the weighted existing values \( f(t^i) \) in the neighborhood of \( v_i \). More efficient minimization techniques can be employed (see e.g., [14]) but we consider this formulation for the sake of simplicity. By using different formulations of \( w \) and different values of \( p \), a family of linear and nonlinear filters is obtained.

Label regularization: The previous discrete regularization can also be adapted to perform semi-supervised clustering by discrete label regularization. To accomodate it to label regularization, we must reformulate the problem. We rewrite the problem as follows [3]. Let \( \mathcal{V} = \{ v_1, \ldots, v_n \} \) be a finite set of data, where each data \( v_i \) is a vector of \( \mathbb{R}^m \), and let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, w) \) be a weighted graph such that data points are vertices and are connected by an edge of \( \mathcal{E} \). The semi-supervised clustering of \( \mathcal{V} \) consists in partitioning the set \( \mathcal{V} \) into \( k \) classes (known beforehand) given initial labels for some vertices of \( \mathcal{V} \). The aim is then to estimate the unlabeled data from the labeled ones. Let \( \mathcal{E}_l \) be a set of labeled vertices, these latter belonging to the \( l \)th class. Let \( \mathcal{V}_0 = \bigcup\mathcal{E}_l \) be the set of initial labeled vertices and let \( \mathcal{V}_0 = \mathcal{V} \setminus \mathcal{V}_0 \) be the initial unlabeled vertices. Each vertex of \( v_i \in \mathcal{V} \) is then described by a vector of labels \( f^0(v_i) = f(t^i)(v_i) \) with
\[
f(t^i)(v_i) = \begin{cases}
 1 & \text{if } v_i \in \mathcal{E}_l, \\
 -1 & \text{if } v_i \in \mathcal{E}_m \text{ for } m \neq l, \\
 0 & \forall v_i \in \mathcal{V} \setminus \mathcal{V}_0.
\end{cases}
\]

Then, the vertex labeling is performed by \( k \) independent regularization processes estimating membership functions \( f_l : \mathcal{V} \rightarrow \mathbb{R} \) for each class \( l \). Using the previously proposed discrete regularization process (15), we use the discrete regularization process (15) to compute each minimization. At the end of the label propagation processes, class membership probabilities have been estimated and the final classification can be obtained for a given vertex \( v_i \in \mathcal{V} \) by
\[
\arg \max_{l \in 1, \ldots, k} \left\{ \frac{f(t^i)(v_i)}{\sum_l f(t^i)(v_i)} \right\}.
\]

V. EXPERIMENTS

A. Literature databases

We have considered label diffusion using the p-Laplacian formulation and label propagation using the eikonal equation for the case of semi-supervised classification on three standard state-of-the-art databases: MNIST [15], OPTDIGITS [16], and PENDIGITS [17]. For these databases, we have merged both the training and the test set (as performed in [1]), resulting in datasets of 70000, 5620, and 10992 instances, for MNIST, OPTDIGITS, and PENDIGITS, respectively. For the OPTDIGITS and PENDIGITS databases, we have used a preprocessed version of the data, giving constant size vectors, and giving invariance to small distortions (see [16] and [17] for more details on the preprocessing routines). For MNIST, we did not preprocess the data. As the authors of [1], we have constructed a K-nearest neighbor graph on the merged datasets, with \( K = 10 \). For the MNIST dataset, we constructed the graph using the two-sided tangent distance [18], and for the two others we used the Euclidean distance between each data points. To compute weights between each vertex, we have used the well known Gaussian Kernel similarity: \( w_{ij} = \exp(-d(v_i, v_j)^2) \) with \( d(v_i, v_j) \) a distance function between two vertices. Since the parameter \( \sigma \) is a strong bottleneck of graph-based methods, we consider strategies for computing automatically its value. We tried two strategies: using a global scaling parameter \( \sigma \), and using a local parameter \( \sigma \) local to each vertex, as in [19], to have a local scaling weight function. In this particular case, the similarity function becomes: \( w_{ij} = \exp(-d(v_i, v_j)^2) \), with \( \sigma \) the local scaling parameter at vertex \( v_i \). We computed each \( \sigma \) as the distance to the \( \text{Mth} \) closest vertex to \( v_i \). To estimate a global \( \sigma \) parameter, we used the method described in
These authors proposed a robust method to estimate a global and a local \( \sigma \) parameter. In this work, we have used the global estimation, which is: \( \hat{\sigma} = 1.4826 \cdot \text{median}(|E_S|) - \text{median}(|E_S|) \), where \( E_S \) is the set of local residuals in the graph, computed as: \( E_i = (\sum_{j \sim v_i} f(v_i) - f(v_j))/\sqrt{|v_j \sim v_i|^2 + |v_j \sim v_i|} \), for a vertex \( v_i \). For further details and justifications, see [20].

\[ \text{Figure 1. Mean classification accuracy rates (y-axis) of the MNIST database, using different percentage of seeds (x-axis), and using global and local estimation of the } \sigma \text{ parameter in the weighting function of the graph. The } p \text{ parameter denotes the } L_p \text{ norm formulation of the label propagation algorithm using the static eikonal equation.} \]

\[ \text{Figure 2. Standard deviation of classification accuracy rates (y-axis) of the MNIST database, using different percentage of seeds (x-axis), and using global and local estimation of the } \sigma \text{ parameter in the weighting function of the graph. The } p \text{ parameter denotes the } L_p \text{ norm formulation of the label propagation algorithm using the static eikonal equation.} \]

\[ \text{Table I. PENDIGITS classification results. For the label propagation algorithm using the eikonal equation and the } p \text{-Laplacian formulation, the best average classification rates were obtained using a potential that is the distance to the mean of the data inside of the growing front and using a globally estimated } \sigma. \]

\[ \text{Table II. OPTDIGITS classification results. For the label propagation algorithm using the eikonal equation and the } p \text{-Laplacian formulation, best average results were achieved the same way as OPTDIGITS classification results.} \]
Once nuclei are extracted, to take a diagnosis, the goal is to classify them in order to know if abnormal cells are present in the slide. To do this, we may use machine learning approaches, that rely on a learning dataset. To create a learning database, we have to handle an images’ database of extracted cells: this kind of databases can contain millions of nuclei to classify. Building a learning database of such a quantity of objects is obviously manually untractable by pathologists. In this context, semi supervised classification techniques can be a great tool to help the labeling of databases of cells. We propose here to build a learning database with both our methods of semi supervised classification on graphs, and evaluating it on a fully labeled reference database. This approach has three main advantages. First, there are few objects to labelize. Second, there is no classifier training, modifying the reference dataset does not imply to retrain a classifier, but to add the new data as a vertex in the graph. Third, graphs intrinsically provide a representation of the organization of the different classes and the position of each nucleus in each class. Such an information can be of crucial importance for cytopathologists on ambiguous cells. We evaluated our methods on the database of cells used in [22]. It is composed of 3956 objects, divided in 18 classes, that can be aggregated into 4 classes: Polynuclear (1117 instances), lymphocytes (1111 instances), macrophages / mesothelials (1231 instances), and abnormal cells (497 instances). We can consider the 3 firsts classes as normal cells. Each nucleus is represented by features (surface, shape, color, texture, ...), leading to 45 characteristics. The tests protocol used for evaluation is the same as in the last subsection. We also used the same graph-weighting methods and potential functions for the eikonal equation. 

Table IV shows the mean results of classification when using the 4 classes described above. We achieve comparable results to the state-of-the-art [22], while we used a semi-supervised classifier and not a supervised one. It is also important to mention here that when crossing pathologists labeling on this database, the recovering rate is only about 60%, which shows the difficulty of the labeling problem. The way of labeling cells we propose can also be a manner of consensus between pathologists. This rate is mostly due to non accordance when labeling normal cells. As one can see in table V, classifying into normal and abnormal cells gives much better results. This information is of great importance when taking

and fifth columns give results for the label diffusion algorithm using the $p$-Laplacian formulation. For each method with given parameters and fixed percentages of seeds, we provide the rate of the best result as well as the mean of the ten trials. Best rates are bolded for each percentage of seeds. To compare our method, we compare ourselves with the most recent and efficient method called Multiclass Total Variation clustering [1]. As it can be seen from the results, for MNIST and PENDIGITS datasets there is always one of our methods that outperforms the state-of-the-art while for OPTDIGITS our configurations compare well with. Depending on the dataset, either the $p$-Laplacian regularization or the eikonal diffusion provides the best results. We can mention here that the algorithm that uses the eikonal equation is much faster than the discrete regularization of the $p$-laplacian, due to the fact that no iteration is necessary for the algorithm, leading to a linear computational complexity. We plan to sequentially combine these two methods to further enhance results, i.e., initializing a partition with the eikonal diffusion, and refining it with the $p$-Laplacian formulation.

B. Digital pathology

Traditionally, the diagnostic decision taken by cytopathologists are based on the study of morphological and texture features of cellular components, seen through a microscope. This is a long process, as a slide can contain thousands or millions of cells in which abnormal cells are very rare or fortunately absent. During the last decade, the advent of fast and efficient high-resolution slide scanners along with the development of computer vision has opened the way to using digital pathology as a diagnosis tool. Fig. 5 gives examples of cytological images. Automatic segmentation can be performed using eikonal equation, as we have done in [21].
We also experimented both our algorithms on a digital pathology framework, initially designed for image processing tasks, can be a diagnostic decision. Table VI shows the execution time of the eikonal equation solving with different potential, and p-Laplacian, using the $L_1$ and $L_2$ norm. First thing to note: solving the eikonal equation is, from far, faster than the p-Laplacian diffusion. This is mainly due to the employed algorithms. For the eikonal equation, we used a fast marching like algorithm, which is linear in terms of nodes and neighborhood in the graph, and does not require to iterate. For the p-Laplacian, we used the Gauss-Jacobi algorithm to solve it iteratively till convergence.

Table VI

<table>
<thead>
<tr>
<th>Alg</th>
<th>Eikonal, $P = 1$</th>
<th>Eikonal, $P = P =$ gradient</th>
<th>Eikonal, $P =$ distance to mean</th>
<th>$p$-Laplacian, $p = 2$</th>
<th>$p$-Laplacian, $p = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>31.4</td>
<td>215.7</td>
<td>45.5</td>
<td>1877.4</td>
<td>781.3</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

We have presented our recently proposed graph-based functional regularization framework and geometric diffusion with the static version of the eikonal equation. We have shown that these two frameworks, initially designed for image processing tasks, can be successfully applied to semi-supervised classification tasks, through three experiments on well known databases of the literature, comparing ourselves with one of the most recent and efficient method. We also experimented both our algorithms on a digital pathology database, showing that they can be used to help pathologists to create a reference database, as well as helping them to take a diagnostic decision, with very short execution time.

ACKNOWLEDGMENT

This work was funded under a doctoral grant supported by the Coeur et Cancer association and the regional council of Lower-Normandy.

REFERENCES


