Geometric PDEs on weighted graphs for semi-supervised classification

Matthieu Toutain, Abderrahim Elmoataz, Olivier Lézoray

Geometric PDEs on weighted graphs for semi-supervised classification

Matthieu Toutain, Abderrahim Elmoataz, Olivier Lézoray
Normandie Univ. UNICAEN, ENSICAEN
GREYC UMR CNRS 6072, Caen, France.

***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, originaly 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ consists in a finite set $\mathcal{V} = \{v_1, \ldots, v_N\}$ of N vertices and a finite set $\mathcal{E} = \{e_1, \ldots, e_{N'}\} \subset \mathcal{V} \times \mathcal{V}$ of N' weighted edges. We assume $\mathcal{G}$ to be undirected, with no self-loops and no multiple edges. Let $e_{ij} = (v_i, v_j)$ be the edge of $\mathcal{E}$ that connects vertices v_i and v_j of $\mathcal{V}$. 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 $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ satisfying $w_{ij} = 0$ if $(v_i, v_j) \notin \mathcal{E}$. The notation $v_i \sim v_j$ is also used to denote two adjacent vertices. We say that $\mathcal{G}$ 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 $\mathcal{H}(\mathcal{V})$ be the Hilbert space of real-valued functions defined on the vertices of a graph. A function $f : \mathcal{V} \to \mathbb{R}$ of $\mathcal{H}(\mathcal{V})$ assigns a real value $x_i = f(v_i)$ to each vertex $v_i \in \mathcal{V}$. Clearly, a function $f \in \mathcal{H}(\mathcal{V})$ can be represented by a column vector in $\mathbb{R}^{|\mathcal{V}|}$: $[x_1, \ldots, x_N]^T = [f(v_1), \ldots, f(v_N)]^T$.

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

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

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

$$(\nabla_w f)(v_i) = [(d_w f)(v_i, v_j) \ : \ v_j \sim v_i]^T. \quad (2)$$

The $\mathcal{L}_p$ norm of this vector represents the *local variation* of the function f at a vertex of the graph. It is defined by [11]:

$$\|(\nabla_w f)(v_i)\|_p = \Big[\sum_{v_j \sim v_i} w_{ij}^{p/2} \big| f(v_j) - f(v_i) \big|^p \Big]^{1/p} \ . \quad (3)$$

Based on the difference operator definition, two weighted directional difference operators are defined. The weighted directional external and internal difference operators have been introduced in [3] as $(d_w^{\pm}) : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{E})$, by

$$(d_w^{\pm} f)(v_i, v_j) = w_{ij}^{1/2}(f(v_j) - f(v_i))^{\pm} \tag{4}$$

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

$$(\nabla_w^{\pm} f)(v_i) = [(d_w^{\pm} f)(v_i, v_j) \ : \ v_j \sim v_i]^T. \tag{5}$$

with the following $\mathcal{L}_p (p \in \{1, 2\})$ norms

$$\|(\nabla_w^{\pm} f)(v_i)\|_p = \Big[\sum_{v_j \sim v_i} w_{ij}^{p/2} \big|(f(v_j) - f(v_i))^{\pm}\big|^p \Big]^{1/p} . \tag{6}$$

III. PDE-BASED GEOMETRIC DIFFUSION

In this section we provide details and recalls on PdEs 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

$$\partial_t f(v_i) = +||(\nabla_w^+ f)(v_i)||_p \text{ and } \partial_t f(v_i) = -||(\nabla_w^- f)(v_i)||_p, \tag{7}$$

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:

$$\begin{cases} \partial_t \phi(x, t) = \mathcal{F}(x, t)|\nabla \phi|, \\ \phi(x, 0) = \phi_0(x), \end{cases} \tag{8}$$

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 controling 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:

$$\partial_t \phi(v_i) = \mathcal{F}^+(v_i)||(\nabla_w^+ \phi)(v_i)|| + \mathcal{F}^-(v_i)||(\nabla_w^- \phi)(v_i)||, \tag{9}$$

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

$$\begin{cases} ||(\nabla_w^- f)(v_i)||_p = P(v_i), & \forall v_i \in \mathcal{V}, \\ f(v_i) = 0, & \forall v_i \in \mathcal{V}_0, \end{cases} \tag{10}$$

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: $f(v_i) = 0 \ \forall \ v_i \in S^0; f(v_i) = +\infty \ \forall \ v_i \in V \backslash S^0$
10: $s(v_i) = +\infty \ \forall \ v_i \in V \backslash S^0$
11: $A = S^0; NB = \{v_i \mid \exists \ v_j \in A \text{ and } v_j \in N(v_i)\}$
12: $FA = V \backslash A \cup NB$
13: **Process :**
14: **while**
15: $FA \neq \emptyset$ **do**
16: $v_i \leftarrow$ first element ofNB
17: remove v_i from NB and add v_i to A
18: **for all** $v_j \in N(v_i) \cap \bar{A}$ **do**
19: compute local solution $t \leftarrow f(v_j)$
20: **if** $t < f(v_j)$ **then**
21: $f(v_j) = t$
22: **if** $v \in FA$ **then**
23: remove v_j from FA and add v_j in NB.
24: **else**
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:

$$\Big(\sum_{v_j \sim v_i} w_{ij}^{p/2} \max(0, (f(v_i) - f(v_j)))^p \Big) = P(u). \tag{11}$$

For the sake of clarity, we present here the solution for the $\mathcal{L}_1$ norm formulation (see [9] for other cases): $\overline{x} = \frac{\sum_{i=1}^{n} h_i a_i + C}{\sum_{i=1}^{n} h_i}$. where $x = f(v_i)$, $n = |v_j \sim v_i|$, $a_i = \{f(v_j)|v_j \sim v_i$ with $j = 1, ..., n\}$, $h_j = 1/\sqrt{w_{ij}}$, and $C = P(v_i)$. 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 ($P(v_i) = ||(\nabla f)(v_i)||_2$). 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: $P(v_i) = ||f(v_i) - \mathcal{C}_m||$, with $\mathcal{C}_m$ the mean inside the front Γ_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-Laplace 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 $\Delta_{w,p} : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{V})$, at a vertex $v_i \in \mathcal{V}$, is defined by:

$$(\Delta_{w,p} f)(v_i) = \frac{1}{2} \sum_{v_j \sim v_i} (\gamma^i_{w,p} f)(v_i, v_j)(f(v_i) - f(v_j)), \quad (12)$$

with

$$(\gamma_{w,p} f)(v_i, v_j) = w_{ij}(\|(\nabla_w f)(v_j)\|_2^{p-2} + \|(\nabla_w f)(v_i)\|_2^{p-2}). \quad (13)$$

In the case where $p = 1$ or 2, we have the definitions of the standard graph curvature $\Delta_{w,1} f(u) = \kappa f$ and graph Laplacian $\Delta_{w,2} f(u) = \Delta f$ 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:

$$\min_{f:\mathcal{V}\rightarrow\mathbb{R}} \{\mathcal{E}_{w,p}(f) = R_{w,p}(f)\}. \quad (14)$$

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 $f \in \mathcal{H}(\mathcal{V}) : R_{w,p}(f) = \frac{1}{2} \sum_{v_i \in \mathcal{V}} ||\nabla_w f(v_i)||_2^p$. This term is a strictly convex function of f (see [13]). To approximate the solution of the minization (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]):

$$\begin{cases} f^{(0)}(v_i) = f^0(v_i), \\ f^{(t+1)}(v_i) = \frac{\sum_{v_j \sim v_i} \gamma^{(t)}(v_i, v_j) f^{(t)}(v_j)}{\sum_{v_j \sim v_i} \gamma^{(t)}(v_i, v_j)}. \end{cases} \quad (15)$$

where $\gamma^{(t)}(v_i, v_j)$ is the γ 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 $f^{(t)}$ 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 inital labels for some vertices of $\mathcal{V}$. The aim is then to estimate the unlabeled data from the labeled ones. Let $\mathcal{C}_l$ be a set of labeled vertices, these latter belonging to the l^{th} class. Let $\mathcal{V}_0 = \bigcup\{\mathcal{C}_l\}_{l=1,\ldots,k}$ be the set of initial *labeled* vertices and let $\mathcal{V}\backslash\mathcal{V}_0$ be the initial *unlabeled* vertices. Each vertex of $v_i \in \mathcal{V}$ is then described by a vector of labels $f^0(v_i) = (f_l^0(v_i))_{l=1,\ldots,k}^T$ with

$$f_l^0(v_i) = \begin{cases} +1 & \text{if } v_i \in \mathcal{C}_l \\ -1 & \text{if } v_i \in \mathcal{C}_m | m \neq l \\ 0 & \forall v_i \in \mathcal{V}\backslash\mathcal{V}_0 \end{cases} \quad (16)$$

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 framework, this is formulated as $\min_{f_l:\mathcal{V}\rightarrow\mathbb{R}} \{R_{w,p}(f_l)\}$. 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 $\arg\max_{l\in 1,\ldots,k} \left\{ \frac{f_l^{(t)}(v_i)}{\sum_l f_l^{(t)}(v_i)} \right\}$.

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 preprocessed 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: $w_{ij} = \exp \frac{-d(v_i,v_j)^2}{\sigma^2}$, with $d(v_i, v_j)$ a distance function between two vertices. Since the parameter σ 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 σ, and using a σ_i local to each vertex, as in [19], to have a local scaling weight function. In this particular case, the similarity function becomes: $w_{ij} = \exp \frac{-d(v_i,v_j)^2}{\sigma_i \sigma_j}$, with σ_i the local scaling parameter at vertex v_i. We computed each σ_i as the distance to the Mth closest vertex to v_i. To estimate a global σ parameter, we used the method described in

[20]. These authors proposed a robust method to estimate a global and a local σ parameter. In this work, we have used the global estimation, which is: $\hat{\sigma} = 1.4826 median(||\underline{\mathcal{E}_{\mathcal{S}}}| - median|\underline{\mathcal{E}_{\mathcal{S}}}||)$, where $\underline{\mathcal{E}_{\mathcal{S}}}$ is the set of local residuals in the graph, computed as: $\mathcal{E}_i = (\sum_{v_j \sim v_i} f(v_i) - f(v_j))/\sqrt{|v_j \sim v_i|^2 + |v_j \sim v_i|}$, for a vertex v_i. For further details and justifications, see [20]. As a

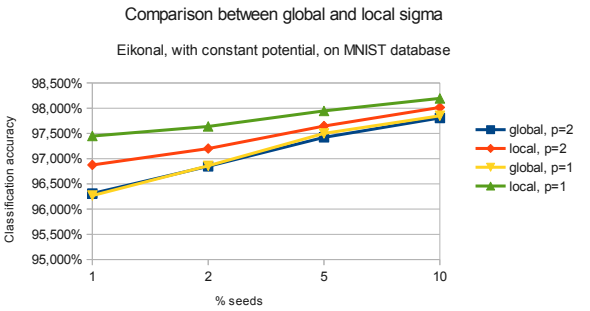


Figure 1. Mean classification accuracy rates (y-axis) of the MNIST database, using different percentage of seeds (x-axis), and using global and local estimation of the σ parameter in the weighting function of the graph. The p parameter denotes the $\mathcal{L}_p$ norm formulation of the label propagation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

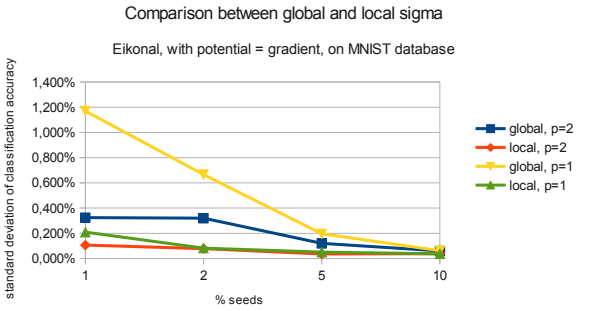
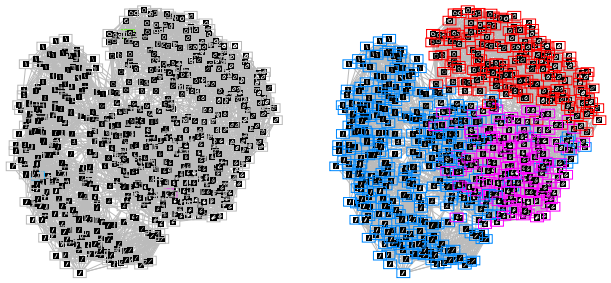


Figure 2. Standard deviation of classification accuracy rates (y-axis) of the MNIST database, using different percentage of seeds (x-axis), and using global and local estimation of the σ parameter in the weighting function of the graph. The p parameter denotes the $\mathcal{L}_p$ norm formulation of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

test protocol, we make ten runs for each algorithm, and we use a percentage of already labelized vertices, settled randomly each time. A typical labeling result is shown in Fig. 3. To get an intuition on how to choose the right weighting method, we evaluated the algorithms using both the global and local σ estimation. Fig. 1 and 2 show mean classification accuracy results and standard deviation of classification accuracy over the ten runs, respectively, with global and locally tuned σ, on the MNIST database, using the label propagation algorithm described in section III. As one can see, in this case results are better when choosing a locally tuned σ. Fig. 2 shows that standard deviation is much better when using a locally tuned σ, we can state here that it permits to be less sensistive to seeds initialization. This is not always the case, depending on the data. For example, best results for OPTDIGITS and PENDIGITS were achieved using a globally estimated σ. However, in our tests, even if it did not always give the best classification results, local



(a) (b)

Figure 3. Illustration of a graph built on a small dataset of digits from MNIST. Subfigure (a) shows one seed per class, and subfigure (b) shows the corresponding label propagation.

Table I
OPTDIGITS CLASSIFICATION RESULTS. FOR THE LABEL PROPAGATION ALGORITHM USING THE EIKONAL EQUATION AND THE p-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 σ.

seeds	Eikonal		pLpl		MTV
	$p=2$	$p=1$	$p=2$	$p=1$	
1%	96.96%	97.24%	97.31%	**98.06%**	-
mean	95.22%	95.11%	93.74%	95%	**98.29%**
2%	98.19%	98.19%	97.79%	**98.54%**	-
mean	97.41%	97.19%	96.31%	97.53%	**98.35%**
5%	98.54%	98.51%	98.24%	**98.63%**	-
mean	98.09%	98.06%	97.51%	98.12%	**98.38%**
10%	**98.67%**	98.65%	98.36%	98.47%	-
mean	98.41%	98.38%	97.82%	98.05%	**98.45%**

σ estimation was almost always the most stable, achieving the best performance in standard deviation among the ten runs. Fig. 4 presents a comparison of the 3 potential functions we used in the eikonal equation on the OPTDIGITS database. As one can see, the potential we proposed works better in general, mostly when there is a small amount of seeded vertices. Classification results are shown in Tables I, II, and III. Due to the lack of space, we only present here results that have provided the best average classification rates. In the Tables, the first column tells the amount of initial labelized data (seeds). Second and third columns give rates for the label propagation algorithm using the eikonal equation. Fourth

Table II
PENDIGITS CLASSIFICATION RESULTS. FOR THE LABEL PROPAGATION ALGORITHM USING EIKONAL EQUATION, AND THE p-LAPLACIAN FORMULATION, BEST AVERAGE RESULTS WERE ACHIEVED THE SAME WAY AS OPTDIGITS CLASSIFICATION RESULTS.

seeds	Eikonal		pLpl		MTV
	$p=2$	$p=1$	$p=2$	$p=1$	
1%	97.25%	**97.3%**	97.25%	96.33%	-
mean	**95.87%**	95.75%	94.45%	94.97%	93.73%
2%	**98.47%**	98.44%	97.54%	98.04%	-
mean	97.31%	**97.38%**	96.45%	96.81%	95.83%
5%	**98.91%**	98.86%	98.51%	98.64%	-
mean	98.23%	**98.25%**	97.86%	97.95%	97.98%
10%	99.16%	**99.18%**	98.74%	98.75%	-
mean	**98.94%**	98.91%	98.61%	98.61%	98.22%

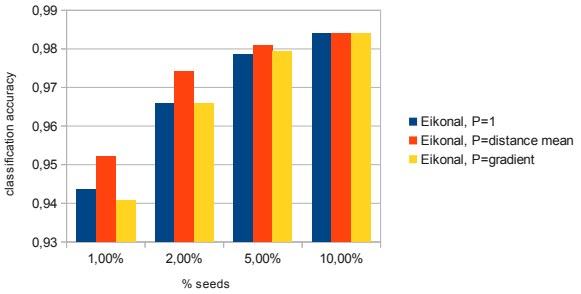


Figure 4. Comparison of the 3 potential functions we used to test the eikonal equation for the task of semi-supervised classification.

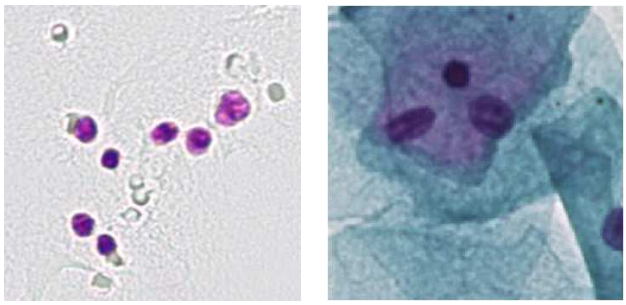
Table III
MNIST CLASSIFICATION RESULTS. FOR THE LABEL PROPAGATION ALGORITHM USING EIKONAL EQUATION, BEST AVERAGE RESULTS WERE ACHIEVED USING A CONSTANT POTENTIAL FOR $p = 1$, AND WITH THE GRADIENT NORM POTENTIAL FOR $p = 2$. BOTH WERE ACHIEVED USING A SELF TUNED σ PARAMETER FOR THE SIMILARITY FUNCTION (WITH THE METHOD OF ZELNIK-MANOR AND PERONA [19]). FOR THE p-LAPLACIAN FORMULATION, BEST AVERAGE RESULTS WERE ACHIEVED USING A GLOBALLY ESTIMATED σ PARAMETER.

seeds	Eikonal		pLpl		MTV
	$p = 2$	$p = 1$	$p = 2$	$p = 1$	
1%	97.21%	97.52%	97.55%	**97.96%**	-
mean	97.08%	97.45%	97.15%	**97.84%**	97.59%
2%	97.48%	97.73%	97.53%	**98.04%**	-
mean	97.37%	97.64%	97.29%	**97.88%**	97.72%
5%	97.83%	98.01%	97.54%	**98.07%**	-
mean	97.79%	97.95%	97.42%	**97.99%**	97.79%
10%	98.19%	**98.25%**	97.64%	98.06%	-
mean	98.13%	**98.19%**	97.53%	98.02%	98.05%

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].



(a) (b)

Figure 5. Examples of cytological images. (a) Feulgen stained image, nuclei are represented in pink. (b) Papanicolaou stained image (gynecology), nuclei are depicted in dark blue/purple, while cytoplasm is stained in light blue.

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 show 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

Table IV
FOUR CLASSES CLASSIFICATION RESULTS

seeds	Eikonal		pLpl	
	$p=2$	$p=1$	$p=2$	$p=1$
1%	65.66%	**66.18%**	60.82%	57.56%
2%	69.83%	**70.49%**	62.73%	65.79%
5%	72.71%	**74.38%**	65.24%	67.84%
10%	75.71%	**77.27%**	68.77%	70.1%

Table V
NORMAL VS ABNORMAL CELLS

seeds	Eikonal		pLpl	
	$p=2$	$p=1$	$p=2$	$p=1$
1%	96.72%	**97.24%**	92.70%	93.23%
2%	97.21%	97.64%	94.86%	**97.73%**
5%	97.89%	98.1%	98.14%	**98.2%**
10%	98.27%	**98.47%**	98.3%	98.28%

a diagnostic decision. Table VI shows the execution time of the eikonal equation solving with different potential, and p-Laplacian, using the $\mathcal{L}_1$ and $\mathcal{L}_2$ 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
AVERAGE EXECUTION TIME COMPARISON, IN MILLISECONDS.

Alg	time
Eikonal, $P=1$	31.4
Eikonal, $P=$ gradient	215.7
Eikonal, $P=$ distance to mean	45.5
p-Laplacian, $p=2$	1877.4
p-Laplacian, $p=1$	781.3

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 dataset, 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

[1] X. Bresson et al., "Multiclass total variation clustering." in *NIPS*, 2013.

[2] E. Merkurjev et al., "An mbo scheme on graphs for classification and image processing." *SIAM J. Imaging Sciences*, vol. 6, no. 4, 2013.

[3] V. T. Ta et al., "Nonlocal pdes-based morphology on weighted graphs for image and data processing," *IEEE Transactions on Image Processing*, vol. 20, no. 6, June 2011.

[4] C. Garcia-Cardona et al., "Fast multiclass segmentation using diffuse interface methods on graphs," *arXiv preprint arXiv:1302.3913*, 2013.

[5] A. Elmoataz et al., "Non-local morphological pdes and p-laplacian equation on graphs with applications in image processing and machine learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 7, pp. 764–779, 2012.

[6] D. Zhou et al., "Semi–supervised learning on directed graphs," in *Advances in Neural Information Processing Systems*, 2005.

[7] M. Hein and M. Maier, "Manifold denoising as preprocessing for finding natural representations of data." in *AAAI*. AAAI Press, 2007.

[8] V.-T. Ta et al., "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognition*, vol. 42, no. 6, 2009.

[9] X. Desquesnes et al., "Eikonal equation adaptation on weighted graphs: fast geometric diffusion process for local and non-local image and data processing," *Journal of Mathematical Imaging and Vision*, vol. 46, 2013.

[10] O. Chapelle et al., *Semi-Supervised Learning*. MIT Press, 2006.

[11] A. Elmoataz et al., "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, 2008.

[12] J. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge university press, 1999, vol. 3.

[13] T. Chan et al., "The digital TV filter and nonlinear denoising." *IEEE Transactions on Image Processing*, vol. 10, 2001.

[14] M. Hidane et al., "Nonlinear multilayered representation of graph-signals." *Journal of Mathematical Imaging and Vision*, vol. 45, no. 2, 2013.

[15] Y. LeCun and C. Cortes. (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

[16] E. Alpaydin and C. Kaynak, "UCI optical recognition of handwritten digits data set," 1998.

[17] E. Alpaydin and F. Alimoglu, "UCI pen-based recognition of handwritten digits data set," 1998.

[18] D. Keysers et al., "Experiments with an extended tangent distance." in *ICPR*, 2000.

[19] L. Zelnik-manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2004.

[20] C. Kervrann, "An adaptive window approach for image smoothing and structures preserving." in *ECCV (3)*, 2004.

[21] M. Toutain et al., "A unified geometric model for virtual slide images processing," in *IFAC*, Caen, France, 2013.

[22] O. Lezoray et al., "A color object recognition scheme: application to cellular sorting," *Machine Vision and Applications*, vol. 14, 2003.