Optimal Neighborhood Preserving Visualization by Maximum Satisfiability[*]

Kerstin Bunte
HIIT, Aalto University
Finland

Matti Järvisalo
HIIT, University of Helsinki
Finland

Jeremias Berg
HIIT, University of Helsinki
Finland

Petri Myllymäki
HIIT, University of Helsinki
Finland

Jaakko Peltonen
University of Tampere, Finland
HIIT, Aalto University, Finland

Samuel Kaski
HIIT, Aalto University,
University of Helsinki, Finland

Abstract

We present a novel approach to low-dimensional neighbor embedding for visualization, based on formulating an information retrieval based neighborhood preservation cost function as Maximum satisfiability on a discretized output display. The method has a rigorous interpretation as optimal visualization based on the cost function. Unlike previous low-dimensional neighbor embedding methods, our formulation is guaranteed to yield globally optimal visualizations, and does so reasonably fast. Unlike previous manifold learning methods yielding global optima of their cost functions, our cost function and method are designed for low-dimensional visualization where evaluation and minimization of visualization errors are crucial. Our method performs well in experiments, yielding clean embeddings of datasets where a state-of-the-art comparison method yields poor arrangements. In a real-world case study for semi-supervised WLAN signal mapping in buildings we outperform state-of-the-art methods.

Introduction

Low-dimensional visualization of high-dimensional datasets is an important and challenging application of nonlinear dimensionality reduction. Many methods are devised to find a lower-dimensional manifold from the data space and thus not designed to reduce dimensionality below the effective dimensionality of the manifold. In visualization the typical target dimensionality is two or three; many methods are not designed to minimize errors that necessarily occur due to the low dimensionality, and perform poorly in visualization (Venna et al. 2010). Recent successful approaches use neighbor embedding (Hinton and Roweis 2002; van der Maaten and Hinton 2008; Venna et al. 2010), fitting neighborhoods in the original space to neighborhoods on the display. The best performing methods in recent comparisons (Venna et al. 2010) use iterative gradient search (IGS). While computationally somewhat demanding, IGS finds local optima only, which reduces quality and reliability of produced visualizations, as results may strongly depend on initialization. Scatterplot visualization was recently formalized as an information retrieval problem (Venna et al. 2010) of retrieving real neighbors of points based on the display; this gives visualization a well-defined objective, but to solve it the authors had to smooth the cost function and apply gradient-based methods which suffer from local optima. We introduce a novel approach to low-dimensional visualization. We solve the information retrieval task directly as a constrained optimization problem on a discrete output display grid. Visualization on a grid is convenient when display size and resolution are constrained as in mobile use, or to ensure on-screen items are non-overlapping for visual clarity and ease of interaction; grid displays have been used in image search and browsing interfaces (Quadrianto et al. 2010). Our approach yields globally optimal visualizations on grids. Optimality is crucial when only a few visualizations can be shown, for example in printed media.

Our approach is based on setting soft constraints on neighbors remaining close-by and non-neighbors remaining far off. Constraints are weighted based on original closeness of the neighbors/non-neighbors. Advantages of our approach are: (1) The solution globally optimally satisfies the neighborhood constraints and needs no initialization or optimization parameters. (2) The method has a well-defined information retrieval interpretation as minimizing total cost of retrieval errors. (3) The method lets end-users restrict the search to visualizations satisfying desired (e.g. structural) properties, by setting additional constraints; for instance we can let the user iteratively narrow the search space by constraints formed from previous solutions. In experiments, we show that globally optimal solutions are found in reasonable time for typical datasets from neighbor embedding literature. Our method gives clean embeddings with fewer artifacts than a state-of-the-art comparison, and outperforms the state-of-the-art in a real-life WLAN signal mapping task.

Neighbor Embedding as Information Retrieval

Low-dimensional visualization is often approached by using nonlinear dimensionality reduction (NLDR). Many NLDR methods are not designed to reduce dimensionality beyond the effective dimensionality of an underlying data manifold; in contrast, on low-dimensional (2d) displays all original data relationships cannot be preserved. Minimizing

[*]Work supported by Academy of Finland (grants #251170, #252845, #256233) and Finnish Funding Agency for Technology and Innovation (project D2I). The authors thank Jessica Davies for providing the MaxHS solver, Teemu Pulkkinen for help with the WLAN data, and the Aalto Science-IT project for computational resources.

the inevitable visualization errors is crucial. Two kinds of errors may occur in neighborhood relationships: *misses* are pairs of data that are close-by (neighbors) in the original space but not on the display, and *false neighbors* are pairs of data close-by on the display but not in the original space. Good visualizations minimize these errors. In fact, this minimization corresponds to an information retrieval task (Venna et al. 2010): minimizing misses maximizes *recall* of retrieving the true neighbors of a point from the display, and minimizing false neighbors maximizes *precision* of retrieving the true neighbors. Recent NLDR methods based on neighbor embedding have an information retrieval interpretation, maximizing recall (Hinton and Roweis 2002; van der Maaten and Hinton 2008), or tradeoffs between recall and precision (Venna et al. 2010). We introduce a new well-performing neighbor embedding approach where, for any definition of neighbors, the information retrieval task is solved exactly by encoding it declaratively as Maximum satisfiability (Li and Manyà 2009). Our approach is directly based on given pairwise neighborhood assignments, and thus very general: it does not require a general similarity measure for the data domain, so we can treat data and its domain knowledge in whichever form is easiest to provide. Pairwise (dis)similarities, direct neighbor constraints, and also missing information can be handled naturally.

Maximum Satisfiability

For a Boolean variable x, there are two literals, x and $\neg x$. A clause is a disjunction ($\vee$, logical OR) of literals and a (truth) assignment is a function τ from Boolean variables to $\{0,1\}$. A clause C is satisfied by τ ($\tau(C)=1$) if $\tau(x)=1$ for a variable x in C, or $\tau(x)=0$ for a literal $\neg x$ in C. A set F of clauses is satisfiable if there is an assignment τ satisfying all clauses in F ($\tau(F)=1$) and unsatisfiable ($\tau(F)=0$ for any τ) otherwise. An instance $F=(F_h,F_s,c)$ of the *weighted partial MaxSAT* problem consists of two sets of clauses, a set F_h of *hard* clauses and a set F_s of *soft* clauses, and a function $c: F_s \rightarrow \mathbb{R}^+$ that associates a non-negative cost (weight) with each soft clause.[1] We refer to weighted partial MaxSAT instances simply as MaxSAT instances. An assignment τ that satisfies F_h is a *solution* to F. The *cost* of a solution τ to F is

$$\textsc{cost}(F,\tau)=\sum_{C\in F_s\colon \tau(C)=0} c(C),$$

i.e., the sum of the costs of the soft clauses not satisfied by τ. A solution τ is (globally) *optimal* for F if $\textsc{cost}(F,\tau)\leq \textsc{cost}(F,\tau')$ for any solution τ' to F. The cost of the optimal solutions of F is denoted by $\textsc{opt}(F)$. The MaxSAT problem asks to find an optimal solution to a given instance F.

In general, the MaxSAT-based approach has two steps; (1) the problem is encoded as a MaxSAT instance F so that any optimal solution to F can be mapped to an optimal solution of the original problem; (2) an off-the-shelf MaxSAT solver is used to find an optimal MaxSAT solution. MaxSAT is today a viable approach to find globally optimal solutions to challenging optimization problems (Chen et al. 2010; Zhu, Weissenbacher, and Malik 2011; Jose and Majumdar 2011; Guerra and Lynce 2012; Berg and Järvisalo 2013; Berg, Järvisalo, and Malone 2014).

Problem Statement: NLDR onto a Grid

Assume we have a finite set $P=\{1,\ldots,n\}$ of (high-dimensional) datapoints to be placed onto a 2d display. Traditional NLDR starts from a distance matrix between points in P. Our setting is more general, as explained next.

Input: Weight matrix. Suppose some pairs of points in P are known to be similar in the sense that *they should be kept close-by on the display*, and others are known to be dissimilar in the sense *that they should not be put close-by*. Such knowledge can arise, e.g., from transformation of a known (dis)similarity matrix of the high-dimensional data, from a known adjacency matrix of graph data, or from domain knowledge of analysts. This knowledge can be encoded as a (possibly asymmetric) *weight matrix* $\mathrm{W}\in\bar{\mathbb{R}}^{n\times n}$ over P, where $\bar{\mathbb{R}}$ stands for $\mathbb{R}\cup\{-\infty,+\infty\}$. For each pair of points x and y, the corresponding matrix element $\mathrm{W}(x,y)$ takes values as follows. $\mathrm{W}(x,y)>0$ denotes that we consider point y a neighbor of point x which is important to keep close-by to x. The greater the value of $\mathrm{W}(x,y)$, the more importance we attach to this neighborhood relationship, where $\mathrm{W}(x,y)=+\infty$ is the maximal importance. $\mathrm{W}(x,y)<0$ denotes that we consider y a non-neighbor of x which is important to keep away from x. The more negative the value of $\mathrm{W}(x,y)$, the more importance we attach to this non-neighbor relationship, where $\mathrm{W}(x,y)=-\infty$ denotes the maximal importance. The value $\mathrm{W}(x,y)=0$ denotes that we have no particular interest where y is placed relative to x. We will use W to define explicit pairwise constraints.

Output onto a grid. We consider the task of placing the points in P onto a two-dimensional display discretized as a $N\times M$ grid, containing a set G of $N\times M$ grid-positions. The locations of the points will be optimized based on the weight matrix W. On the grid, we consider a basic relationship between the grid-positions: *which positions are neighbors*. The notion of a grid-neighborhood is defined via a symmetric grid-neighborhood function $\mathbb{N}: G\rightarrow 2^G$, where $\eta\in\mathbb{N}(\xi)$ means that η and ξ are neighboring grid positions and $\eta\notin\mathbb{N}(\xi)$ means they are not.

Objective. The weight matrix W tells which points we want to place close-by and which ones not. The objective is then to map each of the points in P onto a grid position (i.e., to find a function $\mathbb{G}: P\rightarrow G$) in a way that the grid-neighborhood of the points on the grid resembles the knowledge in W “as closely as possible”. To formalize the notion of resemblance, we use two types of constraints.

Recall: If $\mathrm{W}(x,y)>0$ we set a constraint to map x and y into neighboring grid positions ($\mathbb{G}(y)\in\mathbb{N}(\mathbb{G}(x))$), and set the weight of the constraint to correspond to the importance $\mathrm{W}(x,y)$. Satisfying such constraints from x to the known similar points y means they are placed close to x. If we retrieve all points close to x on the grid, satisfying these constraints minimizes missed similar points and maximizes recall, hence we call them *recall constraints*.

[1] This is more general than the standard definition $c: F_s\rightarrow\mathbb{N}^+$; we do not restrict the costs to be integral. We employ a recent MaxSAT solver which allows real-valued costs on clauses.

Precision: If $\mathrm{W}(x, y) < 0$ we set a constraint to map x and y into non-neighboring grid positions ($\mathbb{G}(y) \notin \mathbb{N}(\mathbb{G}(x))$) with weight $|\mathrm{W}(x, y)|$. Satisfying the set of such constraints from x to other points y minimizes how many known dissimilar points y are retrieved from positions close to x, hence we call them *precision constraints*.

Formally, the objective is to minimize the sum of the weights of the violated recall and precision constraints:

$$\min \sum_{\mathrm{W}(x,y)>0} \frac{1}{2}\mathrm{W}(x, y) \cdot I[\mathbb{G}(y) \notin \mathbb{N}(\mathbb{G}(x))] + \sum_{\mathrm{W}(x,y)<0} \frac{1}{2}|\mathrm{W}(x, y)| \cdot I[\mathbb{G}(y) \in \mathbb{N}(\mathbb{G}(x))] \quad (1)$$

where the indicator function $I[c]$ is 1 (0) iff the condition c holds (does not hold). The objective function has a natural interpretation: it is the importance-weighted sum of misses and false neighbors when, for each point x, we retrieve close-by points y from the grid-neighborhood and compare them to the known neighbors and non-neighbors in W.

If $\mathrm{W}(x, y) \neq \mathrm{W}(y, x)$ (i.e., the weights wrt x, y are asymmetric), the objective function averages the weights of the recall and precision constraints over (x, y). Infinite importance yield hard constraints: If $\mathrm{W}(x, y) = +\infty$ for some x and y, in order to obtain a bounded objective function value, we require $\mathbb{G}(y) \in \mathbb{N}(\mathbb{G}(x))$. Similarly, if $\mathrm{W}(x, y) = -\infty$, we require $\mathbb{G}(y) \notin \mathbb{N}(\mathbb{G}(x))$.

Discussion. Our problem definition is applicable in all typical NLDR settings. It covers arbitrary neighborhood functions for constructing a neighborhood graph, e.g. k-nearest neighborhoods by letting $\mathrm{W}(x, y) = 1$ if y is one of the k nearest neighbors of x, and $\mathrm{W}(x, y) = -1$ otherwise. Known similarity or dissimilarity scores can be mapped into importance weights as we show in experiments. For vectorial data, weights can be constructed based on any distance computation such as geodesic distances (Tenenbaum, de Silva, and Langford 2000) or following a probabilistic approach (Hinton and Roweis 2002; van der Maaten and Hinton 2008). In an interactive setting, additional constraints indicated by users could easily be incorporated to adjust an initial visualization simply by changing the respective entries of W and recomputing.

Our formulation bears some resemblance to graph embedding approaches. Structure preserving embedding (SPE) preserves global topological properties of input graphs, defined by a neighborhood function, in the low-dimensional space (Shaw and Jebara 2009). However, our approach is not restricted to linear constraints, can be weighted, does not need a full adjacency matrix, and the matrix need not be symmetric. Unknown or unsure relationships can be represented as zero-valued entries in W which do not induce any neighborhood constraints. This flexibility arises from the grid-based discrete output; the choice between discrete and continuous depends on the application.

A Compact MaxSAT Formulation

A naive Maxsat formulation of the NLDR objective would use Boolean variables $x^{\boldsymbol{x}}_{\boldsymbol{\xi}}$ which are true iff one data point $\boldsymbol{x}$ is mapped to grid position $\boldsymbol{\xi}$. However, this would yield a quadratic number of variables, and would require a cubic number of constraints. We now detail a more compact log-encoding type of a MaxSAT formulation.

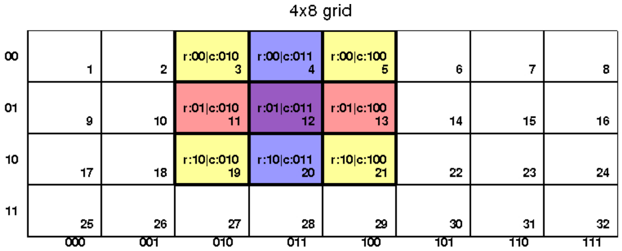
A Bit-based MaxSAT Encoding. Our encoding assumes that the target is a two-dimensional grid containing N rows and M columns such that $N = 2^R$ and $M = 2^C$ for some integers R and C. With these assumptions we can enumerate each row (column) as a binary number using $R = \log_2 N$ and C = $\log_2 M$ bits. An example of a 4×8 grid is shown in Fig. 1. Now the mapping of each point x onto the grid (i.e. the value $\mathbb{G}(x)$) can be represented by the assignment of R (C) row (column) bit variables enumerated from right (i.e., from the least significant bit) to left (i.e., to the most significant bit): $r^x_R, \ldots, r^x_2, r^x_1$ ($c^x_C, \ldots, c^x_2, c^x_1$).

We now encode the NLDR task, starting by defining several intermediate clauses which then allow compact encoding of the whole task. As the grid-neighboorhood $\mathbb{N}$, we define that any two points x and y are mapped to neighboring positions on the grid whenever they are mapped to adjacent rows or columns (or both). More precisely: let r^x (r^y) be the row and c^x (c^y) the column to which the point x (y) is mapped to. Then x and y are neighbors on the grid iff $|r^x - r^y| \leq 1$ and $|c^x - c^y| \leq 1$. The hard clauses of the encoding define this concept by introducing auxiliary variables. The auxiliary variables are then used to formulate the soft precision and recall constraints. For increased clarity we present the encoding in terms of propositional logic.

Hard Clauses: For a fixed pair of points x and y the hard clauses are used in order to define four variables: $\mathrm{SC}^{xy}, \mathrm{SR}^{xy}, \mathrm{AC}^{xy}, \mathrm{AR}^{xy}$ denoting whether or not x and y are mapped to the same column, same row, adjacent columns or adjacent rows respectively. We next describe the constraints defining SC^{xy}, AC^{xy}, SR^{xy}, and AR^{xy}.

As just discussed, points x and y are mapped to the same (adjacent) columns on the grid whenever the values of $c^x_C, \ldots, c^x_2, c^x_1$ and $c^y_C, \ldots, c^y_2, c^y_1$ as binary numbers are equal (differ by at most 1). In order to state this precisely we need to first define the concept of individual bits being equal. We introduce auxiliary variables $\{EQ^{xy}_j\}^C_{j=1}$ defined to be true iff the j-th column bit of both points is the same:

$$EQ^{xy}_j \leftrightarrow (c^x_j \leftrightarrow c^y_j).$$



4x8 grid

00	1	2	r:00|c:010 3	r:00|c:011 4	r:00|c:100 5	6	7	8
01	9	10	r:01|c:010 11	r:01|c:011 12	r:01|c:100 13	14	15	16
10	17	18	r:10|c:010 19	r:10|c:011 20	r:10|c:100 21	22	23	24
11	25	26	27	28	29	30	31	32
	000	001	010	011	100	101	110	111

Figure 1: Illustration of bit-based encoding for 32 grid positions. Grid-neighbors of position 12 (Row: 01, Column: 011) all have row values between: 00 - 10 and column values between 010 - 100. Column, row, and diagonal neighbors of 12 are 4 and 20; 11 and 13; 3, 5, 19, and 21, respectively.

Using these variables the definition of SC^{xy} is straightforward. Points x and y are mapped to the same column iff each column bit in both x and y is the same:

$$\mathrm{SC}^{xy} \leftrightarrow \bigwedge_{j=1}^{C} EQ_j^{xy}.$$

The definition of AC^{xy} is slightly more intricate. We note that if the values represented by the column bits of x and y differ at most by one, the following *differing condition* holds for all $i = 1..C$: "If $c_i^x \neq c_i^y$ and $c_k^x = c_k^y$ for all $k = i + 1..C$, then $c_{k'}^x \neq c_i^x$ and $c_{k'}^y \neq c_i^y$ for all $k' = 1..i - 1$". To be able to encode this compactly we first introduce auxiliary variables F_i^{xy} and F_i^{yx} with the following interpretation: F_i^{xy} (F_i^{yx}) is true iff $c_j^x = 0$ ($c_j^y = 0$) and $c_j^y = 1$ ($c_j^x = 1$) for all $1 \leq j < i$. As constraints:

$$F_i^{xy} \leftrightarrow \bigwedge_{1 \leq j < i} (\neg c_j^x \wedge c_j^y) \text{ and } F_i^{yx} \leftrightarrow \bigwedge_{1 \leq j < i} (c_j^x \wedge \neg c_j^y).$$

Using this we can introduce variables A_i^{xy} and B_i^{xy} which are true iff the differing condition holds at bit i:

$$A_i^{xy} \leftrightarrow \left[\left[\neg EQ_i^{xy} \wedge \bigwedge_{j=i+1}^{C} EQ_j^{xy}\right] \rightarrow (c_i^x \rightarrow F_i^{xy})\right], \text{ and}$$

$$B_i^{xy} \leftrightarrow \left[\left[\neg EQ_i^{xy} \wedge \bigwedge_{j=i+1}^{C} EQ_j^{xy}\right] \rightarrow (\neg c_i^x \rightarrow F_i^{yx})\right].$$

Now points x and y are mapped to adjacent columns iff the differing conditions holds at all column bits:

$$\mathrm{AC}^{xy} \leftrightarrow \bigwedge_{i=1}^{C} (A_i^{xy} \wedge B_i^{xy}).$$

The constraints defining SR^{xy} and AR^{xy} are the same as for SC^{xy} and AC^{xy} except that they are stated over row bits instead of column bits.

Using the four variables SC^{xy}, SR^{xy}, AC^{xy}, and AR^{xy}, we finally define the concept of two points being neighbors in the grid. In a two dimensional grid there are three ways in which x and y can be mapped to neighboring positions. We say that they are column neighbors if they are mapped to the same row and adjacent columns, row neighbors if they are mapped to the same column and adjacent rows, and diagonal neighbors if they are mapped to both adjacent rows and adjacent columns. We introduce three variables CN^{xy}, RN^{xy} and DN^{xy} that are true iff the points x and y are row, column or diagonal neighbors respectively:

$$\begin{aligned}\mathrm{CN}^{xy} &\leftrightarrow (\mathrm{SR}^{xy} \wedge \mathrm{AC}^{xy}),\\ \mathrm{RN}^{xy} &\leftrightarrow (\mathrm{SC}^{xy} \wedge \mathrm{AR}^{xy}),\\ \mathrm{DN}^{xy} &\leftrightarrow (\mathrm{AR}^{xy} \wedge \mathrm{AC}^{xy}).\end{aligned}$$

Soft Clauses: Soft clauses of the encoding encode the objective function via recall and precision constraints using the variables defined by the hard clauses.

Recall: If $\mathrm{W}(x, y) > 0$ in the high dimensional space, we want x and y to be row, column, or diagonal neighbors on the grid: for each point x and for all y such that $\mathrm{W}(x, y) > 0$, we introduce the soft clause

$$(\mathrm{RN}^{xy} \vee \mathrm{CN}^{xy} \vee \mathrm{DN}^{xy})$$

with weight $\mathrm{W}(x, y)/2$. Each such clause exactly corresponds to one term in the recall part of the objective function, that is, one term in the first sum in (1). Note that when $\mathrm{W}(x, y) = +\infty$, the resulting clause becomes hard.

Precision: If $\mathrm{W}(x, y) < 0$, we want x and y not to be row, column, or diagonal neighbors on the grid. We encode this by introducing a new variable PR^{xy}, the soft clause (PR^{xy}) with weight $|\mathrm{W}(x, y)|/2$, and the hard constraint

$$\mathrm{PR}^{xy} \rightarrow (\neg\mathrm{RN}^{xy} \wedge \neg\mathrm{CN}^{xy} \wedge \neg\mathrm{DN}^{xy}).$$

In words, points x, y cannot be mapped to neighboring grid positions whenever the soft clause (PR^{xy}) is satisfied. Furthermore, whenever the x and y are not mapped to neighboring grid positions, the soft clause (PR^{xy}) can be satisfied by simply assigning PR^{xy} to 1. Each such clause corresponds to one term in the precision part of the objective function, that is, one term in the second sum in Eq. (1). Again, when $\mathrm{W}(x, y) = -\infty$, the clause (PR^{xy}) becomes hard.

The resulting bit-based MaxSAT encoding consists of all hard and soft constraints defined above.

Theorem 1 *The bit-based MaxSAT encoding is correct in that, given as input any weight matrix* W *over a set* P *of datapoints, and an* $N \times M$ *grid* G *where* $N = 2^R$ *and* $M = 2^C$ *for some* R *and* C*, there is a one-to-one correspondence between the optimal mappings (wrt the objective function Eq. (1)) of* P *into* G*, and the optimal solutions to the weighted partial MaxSAT instance produced by the bit-based MaxSAT encoding on input* W.

The encoding is much more compact than a naive one: w.r.t. the number of variables and number of clauses, the encoding is quadratic in the number of datapoints, with a logarithmic factor for maximum of the number of rows and columns. The encoding allows several points to be mapped to the same grid position. If desired, this can be ruled out by simply adding the clause $(\neg\mathrm{SC}^{xy} \vee \neg\mathrm{SR}^{xy})$ for each pair of points x, y, forbidding assigning x and y to the same grid-position.

Experiments

We apply the bit-based MaxSAT encoding to the visualization of five different types of synthetic and real-world datasets. In particular, after a comparison with the popular t-SNE method on benchmark data, we showcase our method in a real-world application scenario on WLAN signal mapping, where it outperforms a current state-of-the-art technique (Pulkkinen, Roos, and Myllymäki 2011).

In the experiments we construct the neighborhoods using the *perplexity* measure (Hinton and Roweis 2002) widely applied in the field. Weights for the constraints are derived by the probability p_{ij} for each object $\boldsymbol{x}_i$ to have $\boldsymbol{x}_j$ as neighbor:

$$p_{ij} = \frac{\exp(-d_{ij})}{\sum_{k \neq i} \exp(-d_{ik})}, \text{ where } d_{ij} = \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_i} \ . \quad (2)$$

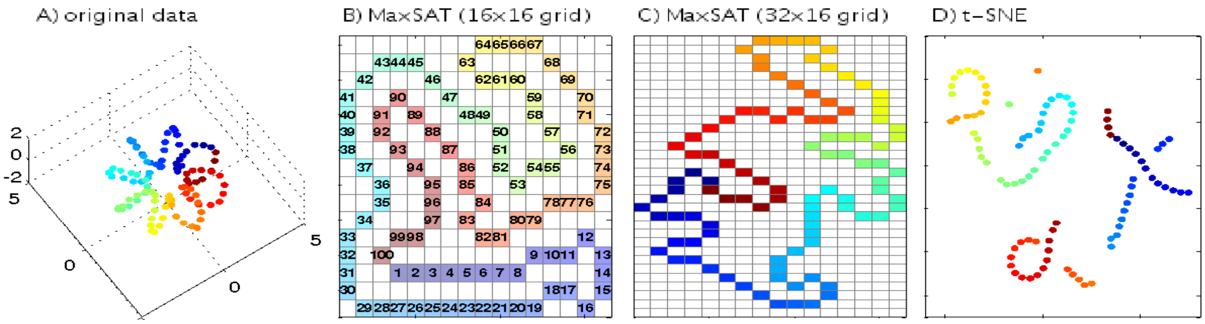


Figure 2: Helix data (A). On two different grids our method unfolds the helix (B-C), whereas t-SNE (D) breaks it apart.

The σ_i is chosen to set the entropy of the distribution over neighbors equal to $\log k$, where the perplexity k denotes the effective number of local neighbors. The weight matrix W is finally summarized by thresholding the probability p_{ij}:

$$\mathrm{W}(i,j) = \begin{cases} p_{ij} & \text{if } p_{ij} \geq \epsilon \;\; \text{(recall constraint)} \\ -p_{ij} & \text{if } p_{ij} < \delta \;\; \text{(precision constraint)} \\ 0 & \text{otherwise} \;\; \text{(no constraint)} \end{cases} \quad (3)$$

where $\delta \in [0, \epsilon]$. For each i, the row $\mathrm{W}(i, \cdot)$ is normalized so that positive entries sum to $+1$ and negative ones to -1, so that recall and precision constraints have equal impact on the embedding. For a 9-cell grid-neighborhood, the threshold ϵ is chosen so that at maximum 5 recall neighborhood constraints are built. If $\delta = \epsilon$ for every point outside the recall neighborhood a precision constraint is constructed. For $\delta < \epsilon$, a region near the recall neighborhood stays unconstrained, alleviating problems related to dense, over-constrained regions in data. The instance constructing source code, the detailed encoding and additional experimental results can be found at *http://research.ics.aalto.fi/mi/software/satnerv/*.

Comparisons on Benchmark Data. For the benchmark data we simply use the constructed W as the ground truth for performance evaluation and compare the neighborhoods on the display to it. For the t-SNE comparison, we set perplexity to yield the same p_{ij} as in W, and use the thresholds when retrieving neighbors from the t-SNE output. We used the following datasets and threshold values for computing the weight matrices in our experiments:

Helix: 100 datapoints from a three-dimensional coiled ring (synthetic), see Fig. 2(A). We computed the neighborhood weights following Eq. (3) using perplexity 5 and threshold $\epsilon = \delta = 0.17$ for precision and recall, resulting in an effective neighborhood of 2.

Coil: A standard dimension reduction and visualization benchmark dataset (Nene, Nayar, and Murase 1996) used in the original t-SNE paper (van der Maaten and Hinton 2008), with images of rotated objects of the first 5 classes. Weights were computed using perplexity 5, $\epsilon = 0.2$ and $\delta = 0.01$, resulting in an effective neighborhood of 4.

Olivetti: The Olivetti (Samaria and Harter 1994) database contains 400 grayscale facial images, with size 64×64, of several persons. Weights computed using perplexity 5 and $\epsilon = \delta = 0.15$.

ISMB: Gene expression microarray experiments (Caldas et al. 2009) from the ArrayExpress database (Parkinson et al. 2009). Following (Caldas et al. 2009), we used 105 experiments with latent variables describing gene set activities as features, and 13 topics and a color scheme. Weights computed using perplexity 5, $\epsilon = 0.2$, $\delta = 0.001$.

We used MaxHS (Davies and Bacchus 2013) as the MaxSAT solver. We compared our approach to the current perhaps most widely used NE method, t-SNE (van der Maaten and Hinton 2008), and SPE (Shaw and Jebara 2009).

As the true high-dimensional neighborhood is known, one can directly count violations of neighborhood (i.e., precision and recall) constraints to compare the methods. An overview of the results is given in Table 1. The SPE implementation by the SPE authors ran out of memory (64 GB) on all datasets except Helix. On all datasets, MaxSAT outperforms t-SNE in terms of violated neighborhood constraints. Fig. 2 shows the result in more detail for the Helix data set: as seen from the figure, t-SNE (D) has difficulties preserving the helical structure, breaking it up into several pieces, and violating 55 neighborhood constraints. However, our method (B–C) succeeds and finds an optimal solution without any neighborhood violations.

Showcase: WLAN signal map. A WLAN positioning system constructs a *radio map* based on the variation of the signal strength measurements according to the geographical location. It is used indoors, where GPS coverage is unavailable and the collection of location tagged training data is tedious and time consuming. The dataset contains 540 *fingerprint vectors* which are each composed of 34 received signal

Table 1: Overview of benchmark results. For each method, neighborhood violations are listed as: (number of recall constraint violations, number of precision constraint violations). The solution quality of MaxSAT is best on all data sets.

Dataset	Neighborhood violations			time	
	MaxSAT	t-SNE	SPE	MaxSAT	t-SNE
Helix	**(0,0)**	(28,401)	**(0,0)**	14s	6s
Coil	**(0,0)**	(30,41158)	memout	14h	8s
Olivetti	**(133,364)**	(137,34699)	memout	10h	7s
ISMB	**(48,30)**	(66,632)	memout	14s	1s

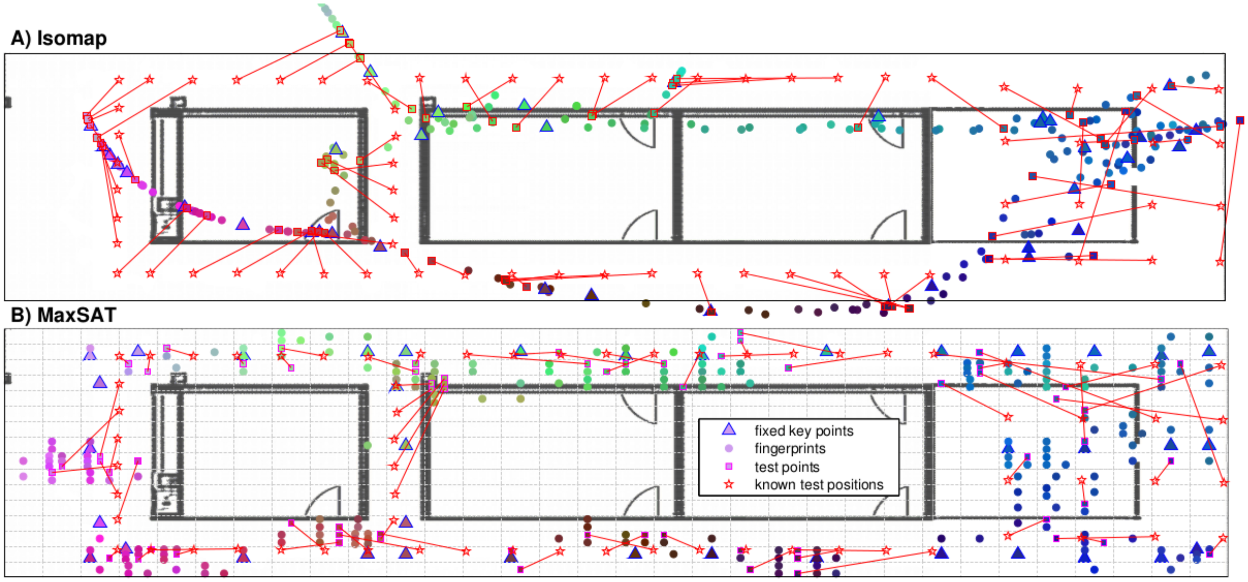


Figure 3: Visualizations of WLAN by the 2-stage Isomap (A) and MaxSAT on a 16 × 64 grid (B). Dots represent the 200 fingerprint vectors, triangles the 38 key points, squares the 66 mapped test points. Similar RSSI vectors are colored similarly. Stars are the recorded geographical positions of the test points; lines connect the mapped and recorded positions.

strength indicator (RSSI) values collected in a real-world office building space of size 24 m × 7 m (Pulkkinen, Roos, and Myllymäki 2011). The positioning task is to place the fingerprints on the floor plan. For 104 fingerprints the geographical coordinates in the area is known, out of them 38 are used for training and are denoted *key points*, and the remaining 66 are used as test points for evaluation purposes. The original paper proposes a two-stage semi-supervised approach: (1) fingerprint vectors are mapped to 2d with the non-linear manifold learning technique Isomap (Tenenbaum, de Silva, and Langford 2000); (2) the key points are used to fix the mapping positions to geographical coordinates using a regression procedure. In contrast, our method uses the position of the key points in the grid directly by simply adding hard constraints over the respective bit variables. We computed the weights with perplexity 15, $\epsilon = 0.1$ and $\delta = 10^{-7}$, resulting in maximal 5 recall neighbors for the MaxSAT encoding. The Isomap approach is replicated using comparable settings with $k = 5$ as input.

We report on a set of experiments, based on varying the number of fingerprints and key points used for constructing the radio map. We evaluate the produced radio maps numerically using the average Euclidean distance from the mapped test points to their recorded geographical position. The Isomap runtimes were $\approx$ 5 seconds. Table 2 provides a comparison of Isomap and our approach. MaxSAT clearly outperforms Isomap: we generally (with only one exception) produce better radio maps (in which the test points are assigned closer to the real positions) than Isomap. When the number of fingerprints is significantly reduced, the MaxSAT solving becomes faster, and still exhibits robust performance in terms of radio map quality. Figure 3 gives an example of the radio maps produced by (A) Isomap and (B) our method.

Conclusions

We present a novel MaxSAT-based low dimensional neighbor embedding approach for visualization. The approach is guaranteed to provide globally optimal embeddings of high-dimensional similarities onto a discrete grid display, maximizing precision and recall. The method embeds data consistently well in practice, yielding clean embeddings with less artifacts than a state-of-the-art t-distributed stochastic neighbor embedding method. Our approach also allows for iteratively enforcing user feedback (expert knowledge) as additional constraints for refining embeddings. In addition to typical benchmark data, as a show-case we applied the approach to semi-supervised WLAN positioning (mapping high-dimensional RSSI vectors directly to geometrical coordinates) where our method outperforms a state-of-the-art positioning method. Overall, MaxSAT yields powerful new tools for neighbor embedding.

Table 2: Quality of Isomap and MaxSAT radio maps: mean distance of mapped points from recorded positions.

all samples/ prints/keys	**Mean distance**		**MaxSAT**	
	Isomap	MaxSAT	**time** (min)	**cost/softclauses**
540/436/38	210.484	**177.115**	1552.13	0.003
404/300/38	189.367	**178.510**	87.00	0.004
304/200/38	209.822	**164.786**	15.12	0.005
204/100/38	234.310	**177.256**	7.09	0.009
404/300/19	224.688	**204.201**	86.18	0.002
304/200/19	213.637	**174.281**	16.14	0.002
204/100/19	292.846	**275.523**	5.08	0.006
404/300/12	**252.490**	282.756	2931.33	0.001
304/200/12	229.226	**186.638**	57.30	0.002
204/100/12	326.314	**251.408**	4.44	0.004

References

Berg, J., and Järvisalo, M. 2013. Optimal correlation clustering via MaxSAT. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops (ICDMW 2013)*, 750–757. IEEE Press.

Berg, J.; Järvisalo, M.; and Malone, B. 2014. Learning optimal bounded treewidth bayesian networks via maximum satisfiability. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS 2014)*, volume 33 of *JMLR Workshop and Conference Proceedings*, 86–95. JMLR.

Caldas, J.; Gehlenborg, N.; Faisal, A.; Brazma, A.; and Kaski, S. 2009. Probabilistic retrieval and visualization of biologically relevant microarray experiments. *Bioinformatics* 25:145–153.

Chen, Y.; Safarpour, S.; Marques-Silva, J.; and Veneris, A. 2010. Automated design debugging with maximum satisfiability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29(11):1804–1817.

Davies, J., and Bacchus, F. 2013. Exploiting the power of MIP solvers in Maxsat. In *Proceedings of the 16th International Conference on Theory and Applications of Satisfiability Testing (SAT 2013)*, volume 7962 of *Lecture Notes in Computer Science*, 166–181. Springer.

Guerra, J., and Lynce, I. 2012. Reasoning over biological networks using maximum satisfiability. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP 2012)*, volume 7514 of *LNCS*, 941–956. Springer.

Hinton, G., and Roweis, S. T. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press. 833–840.

Jose, M., and Majumdar, R. 2011. Cause clue clauses: error localization using maximum satisfiability. In *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2011)*, 437–446. ACM.

Li, C. M., and Manyà, F. 2009. MaxSAT, hard and soft constraints. In *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press. chapter 19, 613–631.

Nene, S.; Nayar, S.; and Murase, H. 1996. Columbia object image library (COIL-20). Technical Report CUCS-005-96.

Parkinson, H.; Kapushesky, M.; Kolesnikov, N.; Rustici, G.; Shojatalab, M.; Abeygunawardena, N.; Berube, H.; Dylag, M.; Emam, I.; Farne, A.; Holloway, E.; Lukk, M.; Malone, J.; Mani, R.; Pilicheva, E.; Rayner, T.; Rezwan, F.; Sharma, A.; Williams, E.; Bradley, X.; Adamusiak, T.; Brandizi, M.; Burdett, T.; Coulson, R.; Krestyaninova, M.; Kurnosov, P.; Maguire, E.; Neogi, S.; Rocca-Serra, P.; Sansone, S.; Sklyar, N.; Zhao, M.; Sarkans, U.; and Brazma, A. 2009. ArrayExpress update – from an archive of functional genomics experiments to the atlas of gene expression. *Nucleic Acids Research* 37:868–872.

Pulkkinen, T.; Roos, T.; and Myllymäki, P. 2011. Semi-supervised learning for WLAN positioning. In *Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN 2011)*, volume 6791 of *Lecture Notes in Computer Science*, 355–362. Springer.

Quadrianto, N.; Kersting, K.; Tuytelaars, T.; and Buntine, W. 2010. Beyond 2D-grids: a dependence maximization view on image browsing. In *Proceedings of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval (MIR 2010)*, 339–348. ACM.

Samaria, F., and Harter, A. 1994. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*.

Shaw, B., and Jebara, T. 2009. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*, 937–944. ACM.

Tenenbaum, J. B.; de Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.

van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9:2579–2605.

Venna, J.; Peltonen, J.; Nybo, K.; Aidos, H.; and Kaski, S. 2010. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research* 11:451–490.

Zhu, C.; Weissenbacher, G.; and Malik, S. 2011. Post-silicon fault localisation using maximum satisfiability and backbones. In *Proceedings of the 11th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2011)*, 63–66. FMCAD Inc.