



Learning Good Edit Similarities with Generalization Guarantees

Aurélien Bellet, Amaury Habrard, Marc Sebban

► To cite this version:

Aurélien Bellet, Amaury Habrard, Marc Sebban. Learning Good Edit Similarities with Generalization Guarantees. European Conference on Machine Learning, Sep 2011, Athens, Greece. pp.188-203. hal-00608631

HAL Id: hal-00608631

<https://hal.science/hal-00608631>

Submitted on 7 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.

Learning Good Edit Similarities with Generalization Guarantees^{*}

Aurélien Bellet¹, Amaury Habrard², and Marc Sebban¹

¹ Laboratoire Hubert Curien UMR CNRS 5516,
University of Jean Monnet, 42000 Saint-Etienne Cedex 2, France,
{aurelien.bellet, marc.sebban}@univ-st-etienne.fr

² Laboratoire d'Informatique Fondamentale UMR CNRS 6166,
University of Aix-Marseille, 13453 Marseille Cedex 13, France,
amaury.habrard@lif.univ-mrs.fr

Abstract. Similarity and distance functions are essential to many learning algorithms, thus training them has attracted a lot of interest. When it comes to dealing with structured data (e.g., strings or trees), *edit similarities* are widely used, and there exists a few methods for learning them. However, these methods offer no theoretical guarantee as to the generalization performance and discriminative power of the resulting similarities. Recently, a theory of learning with (ϵ, γ, τ) -good similarity functions was proposed. This new theory bridges the gap between the properties of a similarity function and its performance in classification. In this paper, we propose a novel edit similarity learning approach (*GESL*) driven by the idea of (ϵ, γ, τ) -goodness, which allows us to derive generalization guarantees using the notion of uniform stability. We experimentally show that edit similarities learned with our method induce classification models that are both more accurate and sparser than those induced by the edit distance or edit similarities learned with a state-of-the-art method.

Keywords: Edit Similarity Learning, Good Similarity Functions.

1 Introduction

Similarity and distance functions between objects play an important role in many supervised and unsupervised learning methods, among which the popular k -nearest neighbors, k -means and support vector machines. For this reason, a lot of research has gone into automatically learning similarity or distance functions from data, which is usually referred to as *metric learning*. When data consists in numerical vectors, a common approach is to learn the parameters (i.e., the transformation matrix) of a Mahalanobis distance [1–4].

Because they involve more complex procedures, less work has been devoted to learning such functions from structured data (for example strings or trees). Still, there exists a few methods for learning *edit distance*-based functions. Roughly speaking, the edit distance between two objects is the cost of the best sequence of

^{*} Preprint submitted to ECML PKDD 2011.

operations (insertion, deletion, substitution) required to transform an object into another, where an *edit cost* is assigned to each possible operation. Most general-purpose methods for learning the edit cost matrix maximize the likelihood of the data using EM-based iterative methods [5–9], which can imply a costly learning phase. Saigo et al. [10] manage to avoid this drawback in the context of remote homologies detection in protein sequences by applying gradient descent to a specific objective function. Some of the above methods do not guarantee to find the optimal parameters and/or are only based on a training set of *positive* pairs: they do not take advantage of pairs of examples that have different labels. Above all, none of these methods offer theoretical guarantees that the learned edit functions will generalize well to unseen examples (while it is the case for some Mahalanobis distance learning methods [4]) and lead to good performance for the classification or clustering task at hand.

Recently, Balcan et al. [11, 12] introduced a theory of learning with so-called (ϵ, γ, τ) -*good similarity functions* that gives intuitive, sufficient conditions for a similarity function to allow one to learn well. Essentially, a similarity function K is (ϵ, γ, τ) -good if an ϵ proportion of examples are on average 2γ more similar to *reasonable* examples of the same class than to *reasonable* examples of the opposite class, where a τ proportion of examples must be *reasonable*. K does not have to be a metric nor positive semi-definite (PSD). They show that if K is (ϵ, γ, τ) -good, then it can be used to build a linear separator in an explicit projection space that has margin γ and error arbitrarily close to ϵ . This separator can be learned efficiently using a linear program and is supposedly sparse.

In this article, we propose a novel *edit similarity learning* procedure driven by the notion of good similarity function. Our approach (*GESL*, for *Good Edit Similarity Learning*) is formulated as an efficient convex programming approach allowing us to learn the edit costs so as to optimize the (ϵ, γ, τ) -goodness of the resulting similarity function. We provide a bound based on the notion of uniform stability [13] that guarantees that our learned similarity will generalize well and induce low-error classifiers. This bound is independent of the size of the alphabet, making *GESL* suitable for handling problems with large alphabet. To the best of our knowledge, this work is the first attempt to establish a theoretical relationship between a learned edit similarity function and its generalization and discriminative power. We show in a comparative experimental study that *GESL* has fast convergence and leads to more accurate and sparser classifiers than other edit similarities.

This paper is organized as follows. In Section 2, we introduce a few notations, and review the theory of Balcan et al. as well as some prior work on edit similarity learning. In Section 3, which is the core of this paper, we present *GESL*, our approach to learning good edit similarities. We then propose a theoretical analysis of *GESL* based on uniform stability, leading to the derivation of a generalization bound. An experimental evaluation of our approach is provided in Section 4. Finally, we conclude this work by outlining promising lines of research on similarity learning.

2 Notations and Related Work

We consider the following binary classification problem: we are given some labeled examples (x, ℓ) drawn from an unknown distribution P over $X \times \{-1, 1\}$, where X is the instance space. We want to learn a classifier $h : X \rightarrow \{-1, 1\}$ whose error rate is as low as possible using pairwise similarities according to a similarity function $K : X \times X \rightarrow [-1, 1]$. We say that K is symmetric if for all $x, x' \in X$, $K(x, x') = K(x', x)$. K is a valid (or Mercer) kernel if it is symmetric and PSD.

2.1 Learning with *Good* Similarity Functions

In recent work, Balcan et al. [11, 12] introduced a new theory of learning with *good* similarity functions. Their motivation was to overcome two major limitations of kernel theory. First, a *good kernel* is essentially a good similarity function, but the theory talks in terms of margin in an implicit, possibly unknown projection space, which can be a problem for intuition and design. Second, the PSD and symmetry requirement often rules out natural similarity functions for the problem at hand. As a consequence, Balcan et al. proposed the following definition of *good similarity function*.

Definition 1 (Balcan et al. [12]). *A similarity function K is an (ϵ, γ, τ) -good similarity function in hinge loss for a learning problem P if there exists a (random) indicator function $R(x)$ defining a (probabilistic) set of “reasonable points” such that the following conditions hold:*

1. $\mathbf{E}_{(x, \ell) \sim P} [1 - \ell g(x) / \gamma]_+ \leq \epsilon$, where $g(x) = \mathbf{E}_{(\ell', x') \sim P} [\ell' K(x, x') | R(x')]$ and $[1 - c]_+ = \max(0, 1 - c)$ is the hinge loss,
2. $\Pr_{x'} [R(x')] \geq \tau$.

Thinking of this definition in terms of number of margin violations, we can interpret the first condition as *an ϵ proportion of examples x are on average 2γ more similar to random reasonable examples of the same class than to random reasonable examples of the opposite class* and the second condition as *at least a τ proportion of the examples should be reasonable*. Note that other definitions are possible, like those proposed in [14] for unbounded dissimilarity functions. Yet Definition 1 is very interesting in two respects. First, it includes all good kernels as well as some non-PSD similarity functions. In that sense, this is a strict generalization of the notion of good kernel [12]. Second, these conditions are sufficient to learn well, i.e., to induce a linear separator α in an explicit space that has low-error relative to L_1 -margin γ . This is formalized in Theorem 1.

Theorem 1 (Balcan et al. [12]). *Let K be an (ϵ, γ, τ) -good similarity function in hinge loss for a learning problem P . For any $\epsilon_1 > 0$ and $0 \leq \delta \leq \gamma\epsilon_1/4$, let $S = \{x'_1, \dots, x'_d\}$ be a (potentially unlabeled) sample of $d = \frac{2}{\tau} \left(\log(2/\delta) + 16 \frac{\log(2/\delta)}{(\epsilon_1\gamma)^2} \right)$ landmarks drawn from P . Consider the mapping $\phi^S : X \rightarrow \mathbb{R}^d$ defined as follows:*

$\phi_i^S(x) = K(x, x'_i)$, $i \in \{1, \dots, d\}$. Then, with probability at least $1 - \delta$ over the random sample S , the induced distribution $\phi^S(P)$ in \mathbb{R}^d has a linear separator α of error at most $\epsilon + \epsilon_1$ at margin γ .

Therefore, if we are given an (ϵ, γ, τ) -good similarity function for a learning problem P and enough (unlabeled) landmark examples, then with high probability there exists a low-error linear separator α in the explicit “ ϕ -space”, which is essentially the space of the similarities to the d landmarks. As Balcan et al. mention, using d_u unlabeled examples and d_l labeled examples, we can efficiently find this separator $\alpha \in \mathbb{R}^{d_u}$ by solving the following linear program (LP):³

$$\min_{\alpha} \sum_{i=1}^{d_l} \left[1 - \sum_{j=1}^{d_u} \alpha_j \ell_i K(x_i, x'_j) \right]_+ + \lambda \|\alpha\|_1. \quad (1)$$

Note that Problem (1) is essentially a 1-norm SVM problem [15] with an *empirical similarity map* [11], and can be efficiently solved. The L_1 -regularization induces sparsity in α : it allows us to automatically select useful landmarks (the reasonable points), ignoring the others, whose corresponding coordinates in α will be set to zero during learning. We can also control the sparsity of the solution directly: the larger λ , the sparser α . Therefore, one does not need to know in advance the set of reasonable points R , it is automatically worked out while learning α .

Our objective in this paper is to make use of the theory of Balcan et al. to efficiently learn (ϵ, γ, τ) -good edit similarities from data that will lead to effective classifiers. In the next section, we review some past work on edit cost learning.

2.2 String Edit Similarity Learning

The classic edit distance, known as the *Levenshtein distance*, is defined as follows.

Definition 2. The *Levenshtein distance* $e_L(x, x')$ between strings $x = x_1 \dots x_t$ and $x' = x'_1 \dots x'_v$ is the minimum number of edit operations to change x into x' . The allowable operations are insertion, deletion and substitution of a symbol.

e_L can be computed in $O(|x| \cdot |x'|)$ time using dynamic programming. Instead of only counting the minimum number of required operations, we can set a cost (or probability) for each edit operation. These parameters are usually represented as a positive cost matrix C of size $(A + 1) \times (A + 1)$, where A is the size of \mathcal{A} , the alphabet x and x' have been generated from (the additional row and column account for insertion and deletion costs respectively). $C_{i,j}$ gives the cost of the operation changing the symbol c_i into c_j , c_i and $c_j \in \mathcal{A} \cup \{\$\}$, where $\$$ is the empty symbol. Given C , a *generalized* edit similarity e_C can be defined as being the cost corresponding to the sequence of minimum cost. This sequence is called the *optimal edit script*.

³ The original formulation proposed in [12] was actually L_1 -constrained. We transformed it into an equivalent L_1 -regularized one.

Using a matrix C that is appropriately tuned to the considered task can lead to significant improvements in performance. For some applications, such matrices may be available, like BLOSUM in the context of protein sequence alignment [16]. However, in most domains it is not the case, and tuning the costs is difficult. For this reason, methods for learning C from data have attracted a lot of interest. Most general-purpose approaches take the form of probabilistic models. Parameter estimation methods of edit transducers were used to infer generative models [5, 6, 9], discriminative models [7] or tree edit models [8].

Note that the above approaches usually use an Expectation-Maximization (EM)-based algorithm to estimate the parameters of probabilistic models. Beyond the fact that EM is not guaranteed to converge to a global optimum, it can also cause two major drawbacks in the context of edit distance learning. First, since EM is iterative, parameter estimation and distance calculations must be performed several times until convergence, which can be expensive to compute, especially when the size of the alphabet and/or the length of the strings are large. Second, by maximizing the likelihood of the data, one only considers pairs of strings of the same class (*positive pairs*) while it may be interesting to make use of the information brought by pairs of strings that have a different label (*negative pairs*). As a consequence, the above methods “move closer together” examples of the same class, without trying to also “move away from each other” examples of different class. In [17], McCallum et al. consider *discriminative* conditional random fields, dealing with positive and negative pairs in specific states, but still using EM for parameter estimation. To overcome the drawback of iterative approaches for the task of detecting remote homology in protein sequences, Saigo et al. [10] optimize by gradient descent an objective function meant to favor the discrimination between positive and negative examples. But this is done by only using positive pairs of distant homologs.

Despite their diversity, a common feature shared by all of the above approaches is that they do not optimize similarity functions to be (ϵ, γ, τ) -good and thus do not take advantage of the theoretical results of Balcan et al.’s framework. In other words, there is no theoretical guarantee that the learned edit functions will work well for the classification or clustering task at hand. In the next section, we propose a novel approach that bridges this gap.

3 Learning (ϵ, γ, τ) -Good Edit Similarity Functions

What makes the edit costs C hard and expensive to optimize is the fact that the edit distance is based on an optimal script which depends on the edit costs themselves. This is the reason why, as we have seen earlier, iterative approaches are very commonly used to learn C from data. In this section, we take a novel convex programming approach based on the theory of Balcan et al. to learn (ϵ, γ, τ) -good edit similarity functions from both positive and negative pairs without requiring a costly iterative procedure. Moreover, this new framework allows us to derive a generalization bound establishing the convergence of our method and a relationship between the learned similarities and their (ϵ, γ, τ) -goodness.

3.1 An Exponential-based Edit Similarity Function

Let $\#(x, x')$ be a $(A + 1) \times (A + 1)$ matrix whose each component $\#_{i,j}(x, x')$ is the number of times the edit operation (i, j) is used to turn x into x' in the optimal Levenshtein script, $0 \leq i, j \leq A$. We define the following edit function:

$$e_G(x, x') = \sum_{0 \leq i, j \leq A} C_{i,j} \#_{i,j}(x, x').$$

Note that to compute e_G , we do not extract the optimal script with respect to C : we use the Levenshtein script⁴ and apply custom costs C to it. Therefore, since the edit script defined by $\#(x, x')$ is fixed, $e_G(x, x')$ is nothing more than a linear function of the edit costs and can be optimized directly.

Recall that in the framework of Balcan et al., a similarity function must be in $[-1, 1]$. To respect this requirement, we define our similarity function to be:

$$K_G(x, x') = 2e^{-e_G(x, x')} - 1.$$

The motivation for this exponential form is related to the one for using exponential kernels in SVM classifiers: it can be seen as a way to introduce nonlinearity to further separate examples of opposite class while moving closer those of the same class. Note that K_G may not be PSD nor symmetric. However, as we have seen earlier, Balcan et al.’s theory does not require these properties, unlike SVM.

3.2 Learning the Edit Costs: Problem Formulation

We aim at learning an edit cost matrix C so as to optimize the (ϵ, γ, τ) -goodness of K_G . It would be tempting to try to find a way to directly optimize Definition 1. Unfortunately, this is very difficult for two reasons. First, it would result in a nonconvex formulation (summing/subtracting up exponential terms). Second, we do not know the set R of reasonable points in advance (R is inferred when learning the classifier). Instead, we propose to optimize the following criterion:

$$\mathbf{E}_{(x, \ell)} [\mathbf{E}_{(x', \ell')} [[1 - \ell \ell' K_G(x, x') / \gamma]_+ | R(x')]] \leq \epsilon'. \quad (2)$$

Criterion (2) bounds that of Definition 1 due to the convexity of the hinge loss. It is harder to satisfy since the “goodness” is required with respect to each reasonable point instead of considering the average similarity to these points. Clearly, if K_G satisfies (2), then it is (ϵ, γ, τ) -good with $\epsilon \leq \epsilon'$.

Let us consider a training sample of N_T labeled points $T = \{z_i = (x_i, \ell_i)\}_{i=1}^{N_T}$ and a sample of landmark examples $S_L = \{z'_j = (x'_j, \ell'_j)\}_{j=1}^{N_L}$. Note that these examples must be labeled in order to allow us to move closer examples of the same class and to separate points of opposite class. In practice, S_L can be a subsample of the training sample T . Recall that the goodness of a similarity only

⁴ In practice, one could use another type of script. We picked the Levenshtein script because it is a “reasonable” edit script, since it corresponds to a shortest script transforming x into x' .

relies on some relevant subset of examples: the reasonable points. Therefore, in the general case, a relevant strategy does not consist in optimizing the similarity with respect to all the landmarks, but rather to some particular ones allowing a high margin with low violation. In order to model this, we suppose the existence of an indicator matching function $f_{land} : T \times S_L \rightarrow \{0, 1\}$ that associates to each element $x \in T$ a non empty set of landmark points. We say that $x' \in S_L$ is a landmark point for $x \in T$ if and only if $f_{land}(x, x') = 1$. We suppose that f_{land} matches exactly N_L landmark points to each $x \in T$. We will discuss in Section 3.4 how we can build f_{land} .

Our formulation requires the goodness for each (x_i, x'_j) with $f_{land}(x_i, x'_j) = 1$. Therefore, we want $[1 - \ell_i \ell'_j K_G(x_i, x'_j) / \gamma]_+ = 0$, hence $\ell_i \ell'_j K_G(x_i, x'_j) \geq \gamma$. A benefit from using this constraint is that it can easily be turned into an equivalent linear one, considering the following two cases.

1. If $\ell_i \neq \ell'_j$, we get:

$$-K_G(x_i, x'_j) \geq \gamma \iff e^{-e_G(x_i, x'_j)} \leq \frac{1-\gamma}{2} \iff e_G(x_i, x'_j) \geq -\log\left(\frac{1-\gamma}{2}\right).$$

We can use a variable $B_1 \geq 0$ and write the constraint as $e_G(x_i, x'_j) \geq B_1$, with the interpretation that $B_1 = -\log(\frac{1-\gamma}{2})$. In fact, $B_1 \geq -\log(\frac{1}{2})$.

2. Likewise, if $\ell_i = \ell'_j$, we get $e_G(x_i, x'_j) \leq -\log(\frac{1+\gamma}{2})$. We can use a variable $B_2 \geq 0$ and write the constraint as $e_G(x_i, x'_j) \leq B_2$, with the interpretation that $B_2 = -\log(\frac{1+\gamma}{2})$. In fact, $B_2 \in [0, -\log(\frac{1}{2})]$.

The optimization problem *GESL* can then be expressed as follows:

$$\begin{aligned} (GESL) \quad & \min_{C, B_1, B_2} \frac{1}{N_T N_L} \sum_{\substack{1 \leq i \leq N_L, \\ 1 \leq j \leq N_T, \\ f_{land}(x_i, x'_j)=1}} V(C, z_i, z'_j) + \beta \|C\|^2 \\ s.t. \quad & V(C, z_i, z'_j) = \begin{cases} [B_1 - e_G(x_i, x'_j)]_+ & \text{if } \ell_i \neq \ell'_j \\ [e_G(x_i, x'_j) - B_2]_+ & \text{if } \ell_i = \ell'_j \end{cases} \\ & B_1 \geq -\log(\frac{1}{2}), \quad 0 \leq B_2 \leq -\log(\frac{1}{2}), \quad B_1 - B_2 = \eta_\gamma \\ & C_{i,j} \geq 0, \quad 0 \leq i, j \leq A, \end{aligned}$$

where $\beta \geq 0$ is a regularization parameter on edit costs, $\|\cdot\|$ denotes the Frobenius norm (which corresponds to the classical L_2 -norm when considering a matrix as a $n \times n$ vector) and $\eta_\gamma \geq 0$ a parameter corresponding to the desired “margin”. The relationship between the margin γ and η_γ is given by $\gamma = \frac{e^{\eta_\gamma} - 1}{e^{\eta_\gamma} + 1}$.

GESL is a convex program, thus we can efficiently find its global optimum. Using slack variables to express the hinge loss, it has $O(N_T N_L + A^2)$ variables and $O(N_T N_L)$ constraints. Note that *GESL* is quite sparse: each constraint involves at most one string pair and a limited number of edit cost variables, making the problem faster to solve. It is also worth noting that our approach is very flexible. First, it is general enough to be used with any definition of e_G that is based on an edit script (or even a convex combination of edit scripts). Second,

one can incorporate additional convex constraints, which offers the possibility of including background knowledge or desired requirements on C (e.g., symmetry). Lastly, it can be easily adapted to the multi-class case.

In the next section, we derive a generalization bound guaranteeing not only the convergence of our learning method but also the overall goodness of the learned edit similarity function for the task at hand.

3.3 Theoretical Guarantees

The outline of this theoretical part is the following: considering that the pairs (z_i, z'_j) used to learn C in *GESL* are not i.i.d., the classic results of statistical learning theory do not directly hold. To derive a generalization bound, extending the ideas of [4, 13] to string edit similarity, we first prove that our learning method has a uniform stability. This is established in Theorem 2 using Lemma 1 and 2. The stability property allows us to derive our generalization bound (Theorem 4) using the McDiarmid inequality (Theorem 3).

In the following, we suppose every string length bounded by a constant $W > 0$, which is not a strong restriction. This implies that for any string pair, $\|\#(x_1, x_2)\| \leq W$,⁵ since the Levenshtein script contains at most $\max(|x_1|, |x_2|)$ operations. We denote the objective function of *GESL* by:

$$F_T(C) = \frac{1}{N_T} \sum_{k=1}^{N_T} \frac{1}{N_L} \sum_{j=1}^{N_L} V(C, z_k, z'_{k_j}) + \beta \|C\|^2,$$

where z'_{k_j} denotes the j^{th} landmark associated to z_k .

The first term of $F_T(C)$, noted $L_T(C)$ in the following, is the empirical loss over the training sample T . Let us also define the loss over the true distribution P , called $L(C)$, and the estimation error D_T as follows:

$$L(C) = \mathbf{E}_{z_k, z'_j} [V(C, z_k, z'_j)] \quad ; \quad D_T = L(C_T) - L_T(C_T),$$

where C_T denotes the edit cost matrix learned by *GESL* from sample T . Our objective is to derive an upper bound on the generalization loss $L(C_T)$ with respect to the empirical loss $L_T(C_T)$.

A learning algorithm is stable [13] when its output does not change significantly under a small modification of the learning sample. We consider the following definition of uniform stability meaning that the replacement of one example must lead to a variation bounded in $O(1/N_T)$ in terms of infinite norm.

Definition 3 (Jin et al. [4], Bousquet and Elisseeff [13]). *A learning algorithm has a uniform stability in $\frac{\kappa}{N_T}$, where κ is a positive constant, if*

$$\forall (T, z), |T| = N_T, \forall i, \sup_{z_1, z_2} |V(C_T, z_1, z_2) - V(C_{T^{i,z}}, z_1, z_2)| \leq \frac{\kappa}{N_T},$$

where $T^{i,z}$ is the new set obtained by replacing $z_i \in T$ by a new example z .

⁵ Also denoted $\|\#(z_1, z_2)\|$ for the sake of convenience when using labeled strings.

To prove that *GESL* has the property of uniform stability, we need the following two lemmas (proven in Appendices 1 and 2).

Lemma 1. *For any edit cost matrices C, C' and any examples z, z' :*

$$|V(C, z, z') - V(C', z, z')| \leq \|C - C'\|W.$$

Lemma 2. *Let F_T and $F_{T^{i,z}}$ be the functions to optimize, C_T and $C_{T^{i,z}}$ their corresponding minimizers, and β the regularization parameter. Let $\Delta C = (C_T - C_{T^{i,z}})$. For any $t \in [0, 1]$:*

$$\|C_T\|^2 - \|C_T - t\Delta C\|^2 + \|C_{T^{i,z}}\|^2 - \|C_{T^{i,z}} + t\Delta C\|^2 \leq \frac{(2N_T + N_L)t2W}{\beta N_T N_L} \|\Delta C\|.$$

Using Lemma 1 and 2, we can now prove the stability of *GESL*.

Theorem 2. *Let N_T and N_L be respectively the number of training examples and landmark points. Assuming that $N_L = \alpha N_T$, $\alpha \in [0, 1]$, *GESL* has a uniform stability in $\frac{\kappa}{N_T}$, where $\kappa = \frac{2(2+\alpha)W^2}{\beta\alpha}$.*

Proof. Using $t = 1/2$ on the left-hand side of Lemma 2, we get

$$\|C_T\|^2 - \|C_T - \frac{1}{2}\Delta C\|^2 + \|C_{T^{i,z}}\|^2 - \|C_{T^{i,z}} + \frac{1}{2}\Delta C\|^2 = \frac{1}{2}\|\Delta C\|^2.$$

Then, applying Lemma 2, we get

$$\|\Delta C\|^2 \leq \frac{2(2N_T + N_L)W}{\beta N_T N_L} \|\Delta C\| \Rightarrow \|\Delta C\| \leq \frac{2(2N_T + N_L)W}{\beta N_T N_L}.$$

Now, from Lemma 1, we have for any z, z'

$$|V(C_T, z, z') - V(C_{T^{i,z}}, z, z')| \leq \|\Delta C\|W \leq \frac{2(2N_T + N_L)W^2}{\beta N_T N_L}.$$

Replacing N_L by αN_T completes the proof. \square

Now, using the property of stability, we can derive our generalization bound over $L(C_T)$. This is done by using the McDiarmid inequality [18].

Theorem 3 (McDiarmid inequality [18]). *Let X_1, \dots, X_n be n independent random variables taking values in \mathcal{X} and let $Z = f(X_1, \dots, X_n)$. If for each $1 \leq i \leq n$, there exists a constant c_i such that*

$$\sup_{x_1, \dots, x_n, x'_i \in \mathcal{X}} |f(x_1, \dots, x_n) - f(x_1, \dots, x'_i, \dots, x_n)| \leq c_i, \forall 1 \leq i \leq n,$$

$$\text{then for any } \epsilon > 0, \quad \Pr[|Z - \mathbf{E}[Z]| \geq \epsilon] \leq 2 \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

To derive our bound on $L(C_T)$, we just need to replace Z by D_T in Theorem 3 and to bound $\mathbf{E}_T[D_T]$ and $|D_T - D_{T^{i,z}}|$, which is shown by the following lemmas (proven in Appendices 3 and 5).

Lemma 3. *For any learning method of estimation error D_T and satisfying a uniform stability in $\frac{\kappa}{N_T}$, we get $\mathbf{E}_T[D_T] \leq \frac{2\kappa}{N_T}$.*

Lemma 4. *For any edit cost matrix learned by GESL using N_T training examples and N_L landmarks, with $B_\gamma = \max(\eta_\gamma, -\log(1/2))$, we have the following bound:*

$$\forall i, 1 \leq i \leq N_T, \quad \forall z, \quad |D_T - D_{T^{i,z}}| \leq \frac{2\kappa}{N_T} + \frac{(2N_T + N_L)(\frac{2W}{\sqrt{\beta B_\gamma}} + 3)B_\gamma}{N_T N_L}.$$

We are now able to derive our generalization bound over $L(C_T)$.

Theorem 4. *Let T be a sample of N_T randomly selected training examples and let C_T be the edit costs learned by GESL with stability $\frac{\kappa}{N_T}$ using $N_L = \alpha N_T$ landmark points. With probability $1 - \delta$, we have the following bound for $L(C_T)$:*

$$L(C_T) \leq L_T(C_T) + 2\frac{\kappa}{N_T} + \left(2\kappa + \frac{2 + \alpha}{\alpha} \left(\frac{2W}{\sqrt{\beta B_\gamma}} + 3\right) B_\gamma\right) \sqrt{\frac{\ln(2/\delta)}{2N_T}}$$

with $\kappa = \frac{2(2+\alpha)W^2}{\alpha\beta}$ and $B_\gamma = \max(\eta_\gamma, -\log(1/2))$.

Proof. Recall that $D_T = L(C_T) - L_T(C_T)$. From Lemma 4, we get

$$|D_T - D_{T^{i,z}}| \leq \sup_{T, z'} |D_T - D_{T^{i,z'}}| \leq \frac{2\kappa + B}{N_T} \text{ with } B = \frac{(2 + \alpha)}{\alpha} \left(\frac{2W}{\sqrt{\beta B_\gamma}} + 3\right) B_\gamma.$$

Then by applying the McDiarmid inequality, we have

$$\Pr[|D_T - \mathbf{E}_T[D_T]| \geq \epsilon] \leq 2 \exp \left(-\frac{2\epsilon^2}{\sum_{i=1}^{N_T} \frac{(2\kappa+B)^2}{N_T^2}} \right) \leq 2 \exp \left(-\frac{2\epsilon^2}{\frac{(2\kappa+B)^2}{N_T}} \right). \quad (3)$$

By fixing $\delta = 2 \exp \left(-\frac{2\epsilon^2}{(2\kappa+B)^2/N_T} \right)$, we get $\epsilon = (2\kappa + B) \sqrt{\frac{\ln(2/\delta)}{2N_T}}$. Finally, from (3), Lemma 3 and the definition of D_T , we have with probability at least $1 - \delta$:

$$D_T < \mathbf{E}_T[D_T] + \epsilon \Rightarrow L(C_T) < L_T(C_T) + 2\frac{\kappa}{N_T} + (2\kappa + B) \sqrt{\frac{\ln(2/\delta)}{2N_T}}. \quad \square$$

This bound outlines three important features of our approach. First, it has a convergence in $O(\sqrt{\frac{1}{N_T}})$, which is classical with the notion of uniform stability. Second, this rate of convergence is independent of the alphabet size, which means that our method should scale well to large alphabet problems. Lastly, thanks to the relation between the optimized criterion and Definition 1 that we established earlier, this bound also ensures the goodness in generalization of the learned similarity function. Therefore, by Theorem 1, it guarantees that the similarity will induce low-error classifiers for the classification task at hand.

3.4 Discussion on the matching function

The question of how one should define the matching function f_{land} relates to the open question of building the training pairs in many metric or similarity learning problems. In some applications, it may be trivial: e.g., a misspelled word and its correction. Otherwise, popular choices are to pair each example with its nearest neighbor or to consider all possible pairs. In our case, matching each example with every landmark may result in a similarity function that performs very poorly, because requiring the goodness over all landmarks (including irrelevant ones) defines an over-constrained problem and does not capture the essence of Definition 1. Remember that on average, examples should be more similar to reasonable points of the same class than to reasonable points of the opposite class. In that sense, reasonable points “represent” the data well. Since classes have intra-class variability, a given reasonable point can only account for a subset of the data. Therefore, reasonable points must be somewhat *complementary*.

Keeping this in mind, we propose the following strategy, used in the experiments. Assuming an even distribution of classes in T , we use a positive parameter $P \leq N_T/2$ to pair each example with its P nearest neighbors of the same class in T and its P farthest neighbors of the opposite class in T , using the Levenshtein distance. Therefore, we have $N_L = 2P$ with $N_L = \alpha N_T$, $0 < \alpha \leq 1$ (where α is typically closer to 0 than to 1). In other words, we essentially take a few landmarks that are already good representatives of a given example and optimize the edit costs so that they become even better representatives. Note that the choice of the Levenshtein distance to determine the neighbors is consistent with our choice to define e_G according to the Levenshtein script.

4 Experimental Results

In this section, we provide an experimental evaluation of the approach presented in Section 3. Using the learning rule (1) of Balcan et al., we compare three edit similarity functions:⁶ (i) K_G , learned by *GESL*,⁷ (ii) the Levenshtein distance e_L , and (iii) an edit similarity function p_e learned with the method of Oncina and Sebban [7].⁸ The task is to learn a model to classify words as either English or French. We use the 2,000 top words lists from Wiktionary.⁹

First, we assess the convergence rate of the two considered edit cost learning methods (i and iii). We keep aside 600 words as a validation set to tune the parameters, using 5-fold cross-validation and selecting the value offering the

⁶ A similarity function that is not in $[-1, 1]$ can be normalized.

⁷ In this series of experiments, we constrained the cost matrices to be symmetric in order not to be dependent on the order in which the examples are considered.

⁸ We used their software SEDiL, available online at <http://labh-curien.univ-st-etienne.fr/SEDiL/>

⁹ These lists are available at http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists. We only considered unique words (i.e., not appearing in both lists) of length at least 4, and we also got rid of accent and punctuation marks. We ended up with about 1,300 words of each language over an alphabet of 26 symbols.

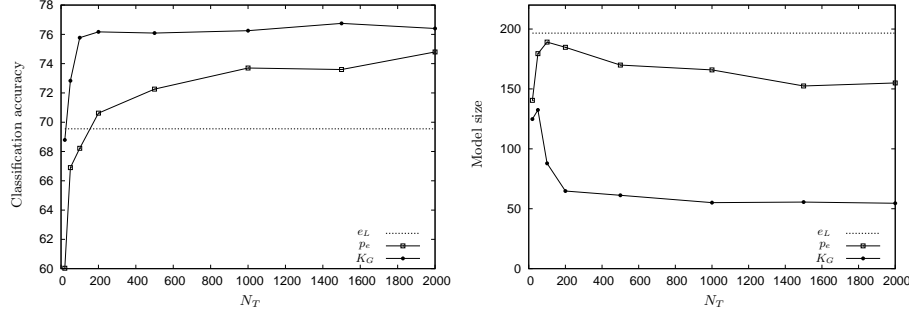


Fig. 1. Learning the costs: accuracy and sparsity results with respect to N_T .

best classification accuracy. We build bootstrap samples T from the remaining 2,000 words to learn the edit costs, as well as 600 words to train the separator α and 400 words to test its performance. Figure 1 shows the accuracy and sparsity results of each method with respect to N_T , averaged over 5 runs. We see that K_G leads to more accurate models than e_L and p_e for every size $N_T > 20$. The difference is statistically significant: the Student’s t -test yields $p < 0.01$. At the same time, K_G requires considerably less reasonable points (thus speeding up classification). This clearly indicates that *GESL* leads to a better similarity than (ii) and (iii). Moreover, the convergence rate of *GESL* is very fast, considering that $(26+1)^2 = 729$ costs must be learned: it needs very few examples to outperform the Levenshtein distance, and about 200 examples to reach convergence. This provides experimental evidence that our method scales well with the size of the alphabet, as suggested by the generalization bound derived in Section 3.3. On the other hand, (iii) seems to suffer from the large number of costs to estimate: it needs a lot more examples to outperform Levenshtein (about 200) and convergence seems to be only reached at 1,000.

Now, we assess the performance of the three edit similarities with respect to the number of examples d_l used to learn the separator α . For K_G and p_e , we use the matrix that performed best in the previous experiment. Taking our set of 2,000 words, we keep aside 400 examples to test the models and build bootstrap samples from the remaining 1,600 words to learn α . Figure 2 shows the accuracy and sparsity results of each method with respect to d_l , averaged over 5 runs. Again, K_G outperforms e_L and p_e for every size d_l (the difference is statistically significant with $p < 0.01$ using a Student’s t -test) while always leading to much sparser models. Moreover, the size of the models induced by K_G stabilizes for $d_l \geq 400$ while the accuracy still increases. This is not the case for the models induced by e_L and p_e , whose size keeps growing. To sum up, the best matrix learned by *GESL* outperforms the best matrix learned by [7], which had been proven to perform better than other state-of-the-art methods.

Finally, one may wonder what kind of words are selected as reasonable points in the models. The intuition is that they should be some sort of “discriminative prototypes” the classifier is based on. Table 1 gives an example of a set of 11

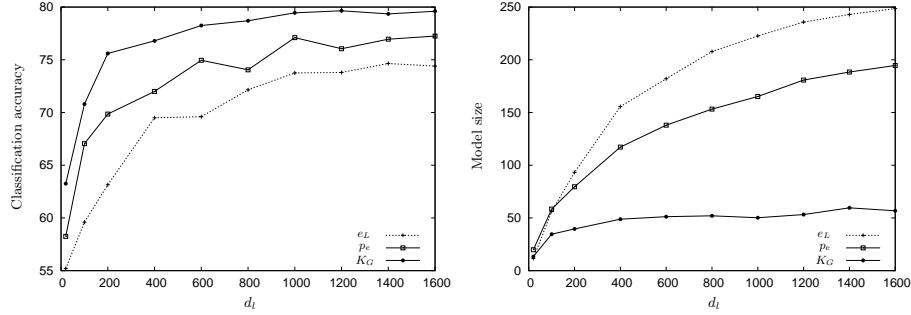


Fig. 2. Learning the separator: accuracy and sparsity results with respect to d_l .

English	French
high showed holy	economiques americaines decouverte
liked hardly	britannique informatique couverture

Table 1. Example of a set of 11 reasonable points.

reasonable points obtained with K_G using a set of 1,200 examples to learn α .¹⁰ This small set actually carries a lot of discriminative patterns (shown in Table 2 along with their number of occurrences in each class over the entire dataset). For example, words ending with *ly* correspond to English words, while those ending with *que* characterize French words. Note that Table 1 also reflects the fact that English words are shorter on average (6.99) than French words (8.26) in the dataset, but the English (resp. French) reasonable points are significantly shorter (resp. longer) than the average (mean of 5.00 and 10.83 resp.), which allows better discrimination.

5 Conclusion and Future Work

In this work, we proposed a novel approach to the problem of learning edit similarities from data that induces (ϵ, γ, τ) -goodness. We derived a generalization bound using the notion of uniform stability that is independent from the size of the alphabet, making it suitable for problems involving large vocabularies. This

¹⁰ We used a high λ value in order to get a small set, thus making the analysis easier.

	w	y	k	q	nn	gh	ai	ed\$	ly\$	es?\$	ques?\$	^h
English	146	144	83	14	5	34	39	151	51	265	0	62
French	7	19	5	72	35	0	114	51	0	630	43	14

Table 2. Some discriminative patterns extracted from the reasonable points of Table 1 (^: start of word, \$: end of word, ?: 0 or 1 occurrence of preceding letter).

bound is related to the goodness of the resulting similarity, which gives guarantees that the similarity will induce accurate models for the task at hand. We experimentally showed that it is indeed the case and that the induced models are also sparser than if we use other (standard or learned) edit similarities. Our approach is flexible enough to be straightforwardly generalized to tree edit similarity learning: one just has to redefine e_G to be a tree edit script. Considering that tree edit distances generally run in cubic time and that the methods for learning tree edit similarities available in the literature are mostly EM-based (thus requiring the distances to be recomputed many times), this seems a very promising avenue to explore. Finally, learning (ϵ, γ, τ) -good Mahalanobis distance could also be considered.

A Appendices

A.1 Proof of Lemma 1

Proof. $|V(C, z, z') - V(C', z, z')| \leq |\sum_{0 \leq i, j \leq A} (C_{i,j} - C'_{i,j}) \#_{i,j}(z, z')| \leq \|C - C'\| \|\#(z, z')\|$. The first inequality uses the 1-lipschitz property of the hinge loss and the fact that B_1 's and B_2 's cancel out. The second one comes from the Cauchy-Schwartz inequality.¹¹ Finally, since $\|\#(z, z')\| \leq W$, the lemma holds. \square

A.2 Proof of Lemma 2

Proof. Let $B = L_T(C_T + t\Delta C) - L_{T^{i,z}}(C_T + t\Delta C) - (L_T(C_T) - L_{T^{i,z}}(C_T))$. Since L_T , F_T , $L_{T^{i,z}}$ and $F_{T^{i,z}}$ are convex functions and using the fact that C_T and $C_{T^{i,z}}$ are minimizers of F_T and $F_{T^{i,z}}$ respectively, we get¹² for any $t \in [0, 1]$:

$$\beta (\|C_T\|^2 - \|C_T - t\Delta C\|^2 + \|C_{T^{i,z}}\|^2 - \|C_{T^{i,z}} + t\Delta C\|^2) \leq B.$$

Then, using the previous upper bound B , we get

$$\begin{aligned} B &\leq |L_T(C_T + t\Delta C) - L_{T^{i,z}}(C_T + t\Delta C) + L_{T^{i,z}}(C_T) - L_T(C_T)| \\ &\leq \frac{2(N_T - 1) + N_L}{N_T N_L} \sup_{\substack{z_1, z_2 \in T \\ z_3, z_4 \in T^{i,z}}} |V(C_T + t\Delta C, z_1, z_2) - V(C_T, z_1, z_2) + \\ &\quad V(C_T, z_3, z_4) - V(C_T + t\Delta C, z_3, z_4)| \\ &\leq \frac{2(N_T - 1) + N_L}{N_T N_L} t \|\Delta C\| \sup_{\substack{z_1, z_2 \in T \\ z_3, z_4 \in T^{i,z}}} (\|\#(z_1, z_2)\| + \|\#(z_3, z_4)\|) \\ &\leq \frac{(2N_T + N_L)t2W}{N_T N_L} \|\Delta C\|. \end{aligned}$$

The second line is obtained by the fact that every z_k in T , $z_k \neq z_i$, has at most two landmark points different between T and $T^{i,z}$, and z and z_i at most N_L different landmarks. To complete the proof, we reorder the terms and use the 1-lipschitz property, Cauchy-Schwartz, triangle inequalities and $\|\#(z, z')\| \leq W$. \square

¹¹ 1-lipschitz implies $|[X]_+ - [Y]_+| \leq |X - Y|$, Cauchy-Schwartz $|\sum_{i=1}^n x_i y_i| \leq \|x\| \|y\|$.

¹² Due to the limitation of space, the details of this construction are not presented in this paper. We advise the interested reader to have a look at Lemma 20 in [13].

A.3 Proof of Lemma 3

$$\begin{aligned}
\text{Proof. } \mathbf{E}_T[D_T] &\leq \mathbf{E}_T[\mathbf{E}_{z,z'}[V(C_T, z, z')] - L_T(C_T)] \\
&\leq \mathbf{E}_{T,z,z'}[|V(C_T, z, z') - \frac{1}{N_T} \sum_{k=1}^{N_T} \frac{1}{N_L} \sum_{j=1}^{N_L} V(C_T, z_k, z'_{k_j})|] \\
&\leq \mathbf{E}_{T,z,z'}[\frac{1}{N_T} \sum_{k=1}^{N_T} (V(C_T, z, z') - V(C_T, z_k, z') + V(C_T, z_k, z') - \frac{1}{N_L} \sum_{j=1}^{N_L} V(C_T, z_k, z'_{k_j}))].
\end{aligned}$$

Since, T, z and z' are i.i.d. from distribution P , we do not change the expected value by replacing one point with another and thus

$$\mathbf{E}_{T,z,z'}[|V(C_T, z, z') - V(C_T, z_k, z')|] = \mathbf{E}_{T,z,z'}[|V(C_T, z, z') - V(C_{T^k}, z, z')|].$$

It suffices to apply this trick twice, combined with the triangle inequality and the property of stability in $\frac{\kappa}{N_T}$ to lead to the lemma. \square

A.4 Lemma 5

In order to bound $|D_T - D_{T^i,z}|$, we need to bound $\|C_T\|$.

Lemma 5. *Let (C_T, B_1, B_2) an optimal solution learned by GESL from a sample T , and let $B_\gamma = \max(\eta_\gamma, -\log(1/2))$, then $\|C_T\| \leq \sqrt{\frac{B_\gamma}{\beta}}$.*

Proof. Since (C_T, B_1, B_2) is an optimal solution then the value reached by F_T is lower than the one obtained with $(\mathbf{0}, B_\gamma, 0)$, where $\mathbf{0}$ denotes the null matrix:

$$\sum_{k=1}^{N_T} \frac{1}{N_T} \sum_{j=1}^{N_L} \frac{1}{N_L} V(C, z_k, z'_{k_j}) + \beta \|C_T\|^2 \leq \sum_{k=1}^{N_T} \frac{1}{N_T} \sum_{j=1}^{N_L} \frac{1}{N_L} V(\mathbf{0}, z_k, z'_{k_j}) + \beta \|\mathbf{0}\|^2 \leq B_\gamma.$$

For the last inequality, note that $V(\mathbf{0}, z_k, z'_{k_j})$ is bounded either by B_γ or 0. Since $\sum_{k=1}^{N_T} \frac{1}{N_T} \sum_{j=1}^{N_L} \frac{1}{N_L} V(C, z_k, z'_{k_j}) \geq 0$, we get $\beta \|C_T\|^2 \leq B_\gamma$. \square

A.5 Proof of Lemma 4

Proof. First, we derive a bound on $|D_T - D_{T^i,z}|$.

$$\begin{aligned}
|D_T - D_{T^i,z}| &= |L(C_T) - L_T(C_T) - (L(C_{T^i,z}) - L_{T^i,z}(C_{T^i,z}))| \\
&\leq |L(C_T) - L(C_{T^i,z})| + |L_T(C_{T^i,z}) - L_T(C_T)| + |L_{T^i,z}(C_{T^i,z}) - L_T(C_{T^i,z})| \\
&\leq \mathbf{E}_{z_1, z_2}[|V(C_T, z_1, z_2) - V(C_{T^i,z}, z_1, z_2)|] + \\
&\quad \frac{1}{N_T} \sum_{k=1}^{N_T} \frac{1}{N_L} \sum_{j=1}^{N_L} |V(C_{T^i,z}, z_k, z'_{k_j}) - V(C_T, z_k, z'_{k_j})| + |L_{T^i,z}(C_{T^i,z}) - L_T(C_{T^i,z})| \\
&\leq 2\frac{\kappa}{N_T} + |L_{T^i,z}(C_{T^i,z}) - L_T(C_{T^i,z})|. \text{ (by using the hypothesis of stability twice)}
\end{aligned}$$

Now, proving Lemma 4 boils down to bounding the last term above. Using similar arguments to the proof of Lemma 2,

$$|L_{T^i,z}(C_{T^i,z}) - L_T(C_{T^i,z})| \leq \frac{(2N_T + N_L)}{N_T N_L} \sup_{\substack{z_1, z_2 \in T \\ z_3, z_4 \in T^i, z}} |V(C_{T^i,z}, z_1, z_2) - V(C_{T^i,z}, z_3, z_4)|.$$

We study two cases that need the 1-lipschitz property of hinge loss and Lemma 5. If $\ell_{z_1}\ell_{z_2} = \ell_{z_3}\ell_{z_4}$, $|V(C_{T^{i,z}}, z_1, z_2) - V(C_{T^{i,z}}, z_3, z_4)| \leq \sqrt{\frac{B_\gamma}{\beta}}W$. Otherwise, if $\ell_{z_1}\ell_{z_2} \neq \ell_{z_3}\ell_{z_4}$, note that $|B_1 + B_2| = \eta_\gamma + 2B_2 \leq 3B_\gamma$. Hence we get

$$|V(C_{T^{i,z}}, z_1, z_2) - V(C_{T^{i,z}}, z_3, z_4)| \leq \sqrt{\frac{B_\gamma}{\beta}}2W + 3B_\gamma. \quad \square$$

References

1. Yang, L., Jin, R.: Distance Metric Learning: A Comprehensive Survey. Technical report, Dep. of Comp. Science and Eng., Michigan State University (2006)
2. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proc. of the Int. Conf. on Machine Learning (ICML). (2007) 209–216
3. Weinberger, K.Q., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. J. of Mach. Learn. Res. (JMLR) **10** (2009) 207–244
4. Jin, R., Wang, S., Zhou, Y.: Regularized distance metric learning: Theory and algorithm. In: Adv. in Neural Inf. Proc. Sys. (NIPS). (2009) 862–870
5. Ristad, E.S., Yianilos, P.N.: Learning String-Edit Distance. In: IEEE Trans. on Pattern Analysis and Machine Intelligence. Volume 20. (1998) 522–532
6. Bilenko, M., Mooney, R.J.: Adaptive Duplicate Detection Using Learnable String Similarity Measures. In: Proc. of the Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD). (2003) 39–48
7. Oncina, J., Sebban, M.: Learning Stochastic Edit Distance: application in handwritten character recognition. Pattern Recognition **39**(9) (2006) 1575–1587
8. Bernard, M., Boyer, L., Habrard, A., Sebban, M.: Learning probabilistic models of tree edit distance. Pattern Recognition **41**(8) (2008) 2611–2629
9. Takasu, A.: Bayesian Similarity Model Estimation for Approximate Recognized Text Search. In: Proc. of the Int. Conf. on Doc. Ana. and Reco. (2009) 611–615
10. Saigo, H., Vert, J.P., Akutsu, T.: Optimizing amino acid substitution matrices with a local alignment kernel. BMC Bioinformatics **7**(246) (2006) 1–12
11. Balcan, M.F., Blum, A.: On a Theory of Learning with Similarity Functions. In: Proc. of the Int. Conf. on Machine Learning (ICML). (2006) 73–80
12. Balcan, M.F., Blum, A., Srebro, N.: Improved Guarantees for Learning via Similarity Functions. In: Proc. of the Conf. on Learning Theory (COLT). (2008) 287–298
13. Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research **2** (2002) 499–526
14. Wang, L., Yang, C., Feng, J.: On Learning with Dissimilarity Functions. In: Proc. of the Int. Conf. on Machine Learning (ICML). (2007) 991–998
15. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm Support Vector Machines. In: Adv. in Neural Inf. Proc. Sys. (NIPS). Volume 16. (2003) 49–56
16. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. In: Proc. of the National Academy of Sciences of the United States of America. Volume 89. (1992) 10915–10919
17. McCallum, A., Bellare, K., Pereira, F.: A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance. In: Conference on Uncertainty in AI. (2005) 388–395
18. McDiarmid, C.: On the method of bounded differences. In: Surveys in Combinatorics. Cambridge University Press (1989) 148–188