Fast and Provably Effective Multi-view Classification with Landmark-based SVM

Valentina Zantedeschi[1], Rémi Emonet[1], and Marc Sebban[1]

Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
<name>.<surname>@univ-st-etienne.fr

Abstract. We introduce a fast and theoretically founded method for learning landmark-based SVMs in a multi-view classification setting which leverages the complementary information of the different views and linearly scales with the size of the dataset. The proposed method – called MVL-SVM – applies a non-linear projection to the dataset through multi-view similarity estimates w.r.t. a small set of randomly selected landmarks, before learning a linear SVM in this latent space joining all the views. Using the uniform stability framework, we prove that our algorithm is robust to slight changes in the training set leading to a generalization bound depending on the number of views and landmarks. We also show that our method can be easily adapted to a missing-view scenario by only reconstructing the similarities to the landmarks. Empirical results, both in complete and missing view settings, highlight the superior performances of our method, in terms of accuracy and execution time, w.r.t. state of the art techniques.

Keywords: Multi-view Learning · Linear SVM · Landmark induced Latent Space · Uniform Stability · Missing Views.

1 Introduction

Machine learning has mainly focused, during the past decades, on settings where training data is embedded in a single feature set. However, data collected nowadays is rarely of a single nature. They are rather observed in multiple, possibly heterogeneous views, where each view can take the form of a different source of information. Examples of multi-view datasets are documents translated in different languages, corpora of pictures with descriptive captions, clips with both audio and video streams and so on. Dealing with such scenarios led to the development of the multi-view learning setting [29,26,22] facing new challenges and requiring scientific breakthroughs. Basically, the need for designing multi-view algorithms relies on the observation that standard learning methods with good performance on single-view problems are, in most cases, inefficient in a multi-view setting [11,23,14]. Indeed, the views of an instance don't necessarily stand-alone because they might individually carry insufficient information

about the task at hand. Even worse, they can be noisy or missing for a part of the training set. Thus, learning a model jointly on the ensemble of views has been proved to be more expressive than view-specific models, because it exploits the possible complementarity between views [29].

The simplest solution to tackle multi-view problems consists in working on the concatenated space of views, i.e. treating each view as a subset of features. However, as the nature of the views can be heterogeneous, i.e. their corresponding features might lie in different input spaces, such a solution is often unfeasible. Moreover, it does not take into account the statistical specificities of each view and can suffer from the curse of dimensionality. A rich literature of methods has been proposed over the years to provide solutions for extracting information from multiple sources. Common multi-view state of the art approaches learn a set of single-view models either by *co-training* [5], in the attempt to capture both the commonalities and idiosyncrasies of the views, or by *co-regularization* [21,11] over the predictions, aiming at maximizing their agreement (see [29,26,22] for surveys). However, because of the computational overhead originated by training and testing with multiple models, these methods are generally slower than standard single-view algorithms.

A few techniques [14,18,15] have also been proposed suggesting to address the problem in a unified space common to all views, allowing us to learn a single model while exploiting the different sources of information. However, this interesting idea faces a major issue: the cost required to extract the complementary information usually results in algorithms nonetheless barely competitive in terms of execution time. Following this promising line of work, we propose in this paper a new latent space-based approach, called MVL-SVM, which leverages the complementary information and which is fast, scalable and provably effective. As shown in Fig.1, we base our work on Support Vector Machines (SVMs) [9] which are well known for their robustness, simplicity, efficiency as well as their theoretical foundations via generalization guarantees. In order to keep the time complexity and memory usage low, we formulate our problem as a Linear SVM in a joint space created by comparing the instances, a view at a time, to a small set of randomly selected landmarks, also observed in multiple views. The instance/landmark comparison is carried out by mean of similarity functions, such as the RBF kernel, each defined on a view. Doing so, we solve a linearized joint problem over all views, in which the statistical characteristics of the views are recoded in similarity estimates with points spread over their spaces. Additionally, by applying non-linear mappings, we efficiently capture the non-linearities and multi-modalities of the view spaces while avoiding the drawbacks of Kernel SVMs (see [20,16,2,6]). Such benefits would not be possible without projecting the points on landmarks: the mapping ensures that the algorithm works on homogeneous features and it also controls the dimensionality of the projected space. To theoretically validate our method, we derive a tight generalization bound by proving its stability w.r.t. changes in the training set, utilizing the framework of the Uniform Stability [7]. Finally, we propose an im-

putation technique for adapting MVL-SVM to the missing-view context, which exploits the information coming from the landmarks.

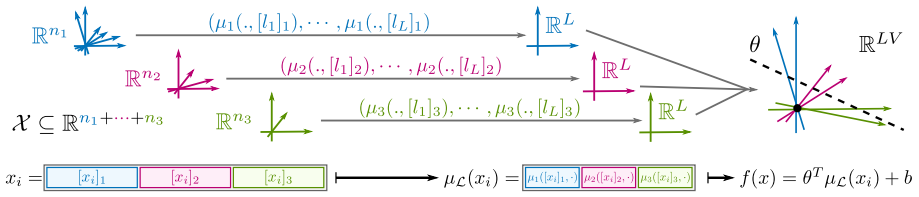


Fig. 1: Overview of the proposed MVL-SVM method. From V views (3 here) of possibly different nature, points are projected on randomly selected landmarks $l_1, \cdots, l_L$ using view specific non-linear mappings $\mu_1, \cdots, \mu_V$. Then, a linear separator is learned in $\mathbb{R}^{LV}$, the joint space of projections.

To sum up, our contribution is three-fold:

1. We introduce a simple, fast and scalable multi-view learning algorithm which benefits from a latent space constructed from similarities to a small set of landmarks. We also show that our approach can be adapted to a missing-view scenario.
2. Using the uniform stability framework, we show that our algorithm is robust to slight changes in the training set leading to a generalization bound that converges uniformly with the number of training examples and that directly depends on the number of views and landmarks.
3. Our experimental results highlight that MVL-SVM allows us to reach very competitive performance in much less time than state of the art methods, overcoming the main issue related to classic latent space-based approaches.

The remainder of the document is organized as follows: Section 2 is devoted to the related work; In Section 3, we present MVL-SVM's algorithm before deriving in Section 4 generalization guarantees in the form of an upper bound on the true risk; An extension of our approach to the missing-view scenario is presented in Section 5; Our experimental results are reported in Section 6.

2 Related Work

The key to effectively tackling multi-view problems is arguably exploiting the diversity between views. As mentioned earlier, the different views rarely contain, alone, sufficient information for the task at hand and leveraging their complementarity is imperative. We can distinguish two principal families of approaches which address multi-view problems: those which optimize a set of single-view learners and combine their predictions, and those which learn a single model in a common space shared by all views.

Co-training and co-regularization methods [5,11] belong to the first category. Basically, they train multiple view-specific models either by alternatively optimizing them, "teaching" one another, or by fostering their smoothness in predictions. The final step of such techniques consists in aggregating the predictions of the view-specific classifiers, for instance by majority vote [21,11] or by weighted majority vote [19,13]. Note that these methods usually face the following issues: their performances are degraded by the computational overload of training and testing multiple learners; also, by usually making the assumption that the views' common information is the only worth keeping, they boil down to denoising the single views from their uncorrelated information. Yet, it is worth noticing that the information relevant to the task is not necessarily the one the views share, but the one that can be extracted by aggregating the views' incomplete information.

The second category of multi-view learning algorithms contains methods working on Vector-valued Kernel Hilbert Spaces (vvRKHS) [17], whose reproducing kernel outputs, for a pair of multi-view points, a matrix of similarities, each component weighting the similarity of the points observed in a pair of views. These methods are extremely powerful, because they are able to keep the statistical specificities of each view and to extract the complementary information from the diversity of the sources. Of particular interest is Multi-view Metric Learning (MVML [14]) which combines vvRKHS with Metric Learning [27,3] and has proved to outperform Kernel-based state of the art methods, such as Multiple Kernel Learning [12]. MVML jointly learns a classifier and a kernel matrix encoding the within-view and between-view relationships. Although the computations are sped up by working on an approximated Gram matrix, obtained through the Nyström technique [25], this powerful approach is not sufficiently competitive in terms of execution time. To overcome this complexity issue of kernel-based methods, L^3-SVM has been recently proposed in [28] for single-view classification as a different way to take advantage of the discriminatory capabilities of kernels while being fast and scalable. Through clustering and projections on *landmarks*, this algorithm speeds up the learning process while training expressive classifiers, competitive with Kernel-SVMs. This algorithm also comes with a generalization bound on the true risk, even though it is derived independently from the number of clusters. In this paper, we aim at (i) benefiting from this promising landmarks-based SVM paradigm, (ii) adapting it to the multi-view scenario and (iii) deriving theoretical guarantees which take into account both the number of landmarks and views.

Another open problem in multi-view learning is how to deal with realizations of the points that are partially incomplete, i.e. some views of multiple instances are missing. In order to apply a multi-view algorithm, one might have to discard the points with missing views, which may result in a loss of performance, or to complete them using different techniques while trying not to introduce bias. Common practices consist in replacing the missing values with zeros or with the mean or median values of the considered feature. On the other hand, multi-view kernel specific techniques have been proposed to complete the Gram matrices

of incomplete views. By making the assumption that similarities between points should be consistent from one view to another, the missing values of a view's Gram matrix are inferred by aligning its eigen-space to the ones of the other views. This can be done by Graph Laplacian regularization [24] (finding the matrix that minimizes its product with the Graph Laplacian matrix of a complete reference view) or by learning convex combinations of normalized kernel matrices [4]. A first limitation of such approaches comes from the fact that they cannot be applied on non square matrices. This prevents us from using them on matrices containing the similarities to a subset of points, like in landmarks-based SVM approaches. Beyond this constraint, the assumption that views are strongly similar and the constraint of having the points altogether observed in a view seem too strong. Another multi-view imputation technique relies on the existence of view generating functions for approximating the missing values. For example, in [1], the authors resort to translation functions for documents in multiple-languages. Unfortunately, depending on the application at hand, such functions are not always available.

In this paper, we make use of the information coming from a small set of randomly selected landmarks to impute the missing values. As for Laplacian imputation [24], we do not need to reconstruct the actual missing features of a point, but only its similarities w.r.t. the landmarks, which drastically simplifies the problem. Through Least Square minimization, we impute the missing similarities by learning the linear combinations of the landmarks projected in the latent space.

3 Multi-view Landmarks-based SVM (MVL-SVM)

3.1 Notations and Problem Statement

We consider the problem of learning from a dataset $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$ of m instances i.i.d. according to a joint distribution $\mathcal{D}$ and observed in a multi-view space of V views, so that $x_i \in \mathcal{X} \subseteq \mathbb{R}^{n_1+\cdots+n_V}$, in which views are potentially of different dimensionality, and $y_i \in \mathcal{Y} = \{-1, 1\}$. In the following, we will use the notation $[x_i]_v$ to refer to the realization of point x_i in the view v. Moreover, we denote $\mathcal{L} = \{l_p\}_{p=1}^L \in \mathcal{X}^L$, a set of L landmarks of the input space selected randomly from the training sample.

We aim at learning a classifier $f : \mathcal{X} \to \mathbb{R}$ in the joint space defined by the different views as follows (see also Fig.1):

$$f(x) = \theta^T \mu_{\mathcal{L}}(x) + b \tag{1}$$

where $\theta \in \mathbb{R}^{LV}$ is a vector of weights, each associated to a view v of a landmark p and $\mu_{\mathcal{L}}(x_i) = [\mu_1([x_i]_1, [l_1]_1), \ldots, \mu_1([x_i]_1, [l_L]_1), \ldots, \mu_V([x_i]_V, [l_L]_V)]$ can be interpreted as the mapping function from the input space $\mathcal{X}$ to a new landmark space $\mathcal{H} \subseteq \mathbb{R}^{LV}$. The sign of the function is retained for prediction

($\hat{y} = sign(f(x))$), i.e. test examples need to be projected as well on the latent space. Notice that each point is compared to the set of landmarks one view at a time and that the problem is now linear in the space $\mathcal{H}$. To capture the non-linearities of the space, we rely on the choice of view-specific score functions $\mu_v : \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \to \mathbb{R}$ between representations of points in a given view.

The choice of projecting the dataset on selected landmarks is crucial for the discriminatory power of the resulting classifier. As a matter of fact, it enables to express the statistical peculiarities of a view space through similarity estimates and additionally it allows us to work on a latent space common to all views, which has multiple benefits: firstly, it allows to control the dimensionality of the space by choosing the number of landmarks; secondly, it enables to learn a unique classifier, avoiding the problem of combining the outputs of view-specific models; lastly, and most importantly, it loosens the assumptions on the relationship between view information, especially the one on their correlation.

3.2 Optimization Problem and Algorithm

As for standard SVM, our objective function consists in maximizing the margin between the class hyperplanes while minimizing a surrogate function of the classification error:

$$F(f) = \frac{1}{2} \|f\|^2 + \frac{c}{m} \sum_{i=1}^{m} \ell(f, z_i) \tag{2}$$

where $\ell(f, z) = \max(0, 1 - yf(x))$ is the hinge loss. We formulate the multi-view classification problem as a soft-margin SVM learning that we solve in its primal form:

$$\begin{aligned} &\underset{\theta, b, \xi}{\arg\min} \frac{1}{2} \|\theta\|^2 + \frac{c}{m} \sum_{i=1}^{m} \xi_i \\ s.t.\ \ &y_i \left(\theta^T \mu_{\mathcal{L}}(x_i) + b\right) \geq 1 - \xi_i \ ; \ \xi_i \geq 0 \ \forall i = 1..m. \end{aligned} \tag{3}$$

The main difference with standard-SVM is the working input space and its interpretation. Basically, we learn how to linearly combine the point-landmark similarities, describing how they should change over the views for a class. The pseudo-code of MVL-SVM is reported in Algorithm 1.

To recapitulate, our landmark-induced latent space allows us to efficiently extract the complementarity between views while capturing their statistical peculiarities. Moreover, MVL-SVM's flexibility makes it suited to deal with multiple not necessarily correlated views, potentially heterogeneous and of different dimensionality. This flexibility, combined with its scalability, makes MVL-SVM applicable to a wide set of problems.

Input: a sample $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m \subseteq \mathbb{R}^{n_1+\cdots+n_V} \times \{-1, 1\}$
and a set of view-specific score functions $\{\mu_v : \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \to \mathbb{R}\}_{v=1}^V$
1. Select $\mathcal{L} = \{l_p\}_{p=1}^L$ uniformly from $\{x_i\}_{i=1}^m$;
2. Project $\mathcal{S}$ on the latent space:
for $i = 1$ **to** m **do**
$\mu_{\mathcal{L}}(x_i) = [\mu_1([x_i]_1, [l_1]_1), \ldots, \mu_1([x_i]_1, [l_L]_1), \ldots, \mu_V([x_i]_V, [l_L]_V)]$
end for
3. Learn $\theta \in \mathbb{R}^{LV}$ as the minimizer of Problem (3);
4. Use $sign(\theta^T \mu_{\mathcal{L}}(x) + b)$ for prediction.

Algorithm 1: MVL-SVM algorithm.

4 Theoretical Results

Since the parameters θ and b are optimized from a finite set of training examples, a key question is how the learned model behaves at test time. Using the theoretical framework of the Uniform Stability [7], we analyze in this section the generalization properties of our algorithm by deriving an upper bound on its true risk. We will see that the stability of our method and, consequently, its generalization capabilities, depend on the choice of the projection functions, the number of selected landmarks and the characteristic of the dataset, such as the number of views and the size of the training set.

4.1 MVL-SVM's uniform stability

An algorithm is said to enjoy uniform stability if it outputs similar solutions from slightly different datasets. Let S be the original dataset and S^i the set obtained after replacing the i^{th} sample z_i of S by a new sample z_i' drawn according to the unknown underlying distribution $\mathcal{D}$. We say that an algorithm is uniformly stable if, on a new instance, the difference between the loss suffered by the solution f learned from S and the loss suffered by the solution f^i learned from S^i converges in $O(\frac{1}{m})$. More formally,

Definition 1. (Uniform Stability) *A learning algorithm A has uniform stability $2\frac{\beta}{m}$ w.r.t. the loss function ℓ with $\beta \in \mathbb{R}^+$ if*

$$\sup_{z \sim \mathcal{D}} |\ell(f, z) - \ell(f^i, z)| \leq 2\frac{\beta}{m}.$$

The uniform stability is directly implied by the triangle inequality if

$$\sup_{z \sim \mathcal{D}} |\ell(f, z) - \ell(f^{\setminus i}, z)| \leq \frac{\beta}{m}$$

where $f^{\setminus i}$ is learned on $S^{\setminus i}$, the set S without the i^{th} instance z_i.

The notion of σ-admissibility is helpful for studying the uniform stability of an algorithm. In order for the algorithm to be stable, it is necessary to prove

that, for a given point, the difference between its loss function evaluated for any two possible hypotheses is bounded by the difference of hypotheses' predictions, scaled by a constant.

Definition 2. (σ-admissibility) *A loss function $\ell(f, z)$ is σ-admissible w.r.t. f if it is convex w.r.t. its first argument and $\forall f_1, f_2$ and $\forall z = (x, y) \in \mathcal{Z}$:*

$$|\ell(f_1, z) - l(f_2, z)| \leq \sigma |f_1(x) - f_2(x)|.$$

In our case, and according to [7], we know that the hinge loss is 1-admissible. We can now present our main theoretical result.

Theorem 1. Uniform Stability *Given the inverse regularizer weight c (from Eq. (3)), MVL-SVM has uniform stability $\frac{cLVM^2}{m}$, where $M = 1$ if μ_v uses the RBF kernel.*

Proof. As $\ell(f, z)$ is 1-admissible, $\forall z = (x, y) \in \mathcal{Z}$,

$$|\ell(f^{\setminus i}, z) - \ell(f, z)| \leq |f^{\setminus i}(x) - f(x)| = |\Delta f(x)| \tag{4}$$

with $\Delta f = f^{\setminus i} - f$. By denoting $\Delta\theta = \theta^{\setminus i} - \theta$, we can derive, $\forall z = (x, y) \in \mathcal{Z}$,

$$\begin{aligned}
|\Delta f(x)| &= |\theta^{\setminus i} \mu_{\mathcal{L}}(x)^T - \theta \mu_{\mathcal{L}}(x)^T| \\
&= |(\theta^{\setminus i} - \theta) \mu_{\mathcal{L}}(x)^T| \\
&\leq \left\| \theta^{\setminus i} - \theta \right\| \|\mu_{\mathcal{L}}(x)\| && (5) \\
&\leq \|\Delta\theta\| \, \|\mu_{\mathcal{L}}(x)\| \\
&\leq \|\Delta\theta\| \sqrt{LV} \, \|\mu_{\mathcal{L}}(x)\|_\infty && (6) \\
&\leq \|\Delta\theta\| \sqrt{LV} \max_{l,v} (\mu_v([x]_v, [l]_v)) \\
&\leq \|\Delta\theta\| \sqrt{LV} M && (7)
\end{aligned}$$

with $M = \max_{l,v}(\mu_v([x]_v, [l]_v))$.

Eq. (5) is due to the Cauchy-Swartz inequality and Eq. (6) is because $\|\mu_{\mathcal{L}}(x)\| \leq \sqrt{LV} \, \|\mu_{\mathcal{L}}(x)\|_\infty$ recalling that $\mu_{\mathcal{L}}(x) \in \mathbb{R}^{LV}$.

The value of M depends on the chosen scores functions $\{\mu_v\}_{v=1}^V$. For instance, if all μ_v are the RBF kernel $M = 1$.

From Lemma 21 of [7] we get:

$$2 \|\Delta\theta\|^2 \leq \frac{c}{m} |\Delta f(x_i)|.$$

Then, by instantiating Eq. (7) for $x = x_i$, we get

$$\|\Delta\theta\|^2 \leq \frac{c}{2m} |\Delta f(x_i)| \leq \frac{c}{2m} \|\Delta\theta\| \sqrt{LV} M$$

and as $\|\Delta\theta\| > 0$, we obtain

$$\|\Delta\theta\| \leq \frac{c}{2m}\sqrt{LV}M. \tag{8}$$

So, plugging Eq. (8) in Eq. (7), we get

$$\forall z = (x, y), \;\; |\Delta f(x)| \leq \|\Delta\theta\| \sqrt{LV}M \leq \frac{cLVM^2}{2m}$$

which, with Eq. (4), gives the $\frac{cLVM^2}{m}$ uniform stability. □

Note that the stability of MVL-SVM depends on the number of landmarks L. Our method is stable only if $L \ll \frac{m}{V}$, which is not a strong condition considering that usually $m \gg V$. Moreover, this bound expresses that, the smaller L, the more stable the algorithm. This is consistent with the fact that L controls the dimensionality of the projected space in which the multi-view model is learned.

4.2 Generalization bound

From [7], we know that:

Theorem 2. *Let A be an algorithm with uniform stability $\frac{2\beta}{m}$ w.r.t. a loss ℓ such that $0 \leq \ell(f, z) \leq E$, for f the minimizer of F and $\forall z \in \mathcal{Z}$. Then, for any i.i.d. sample S of size m and for any $\delta \in (0, 1)$, with probability $1 - \delta$:*

$$R_{\mathcal{D}}(f) \leq \hat{R}_S(f) + \frac{2\beta}{m} + \big(4\beta + E\big)\sqrt{\frac{\ln\frac{1}{\delta}}{2m}}$$

where $R_{\mathcal{D}}(f)$ is the true risk on distribution $\mathcal{D}$ and $\hat{R}_S(f)$ is the empirical risk on sample S.

Corollary 1. *The generalization bound of MVL-SVM derived using the Uniform Stability framework is as follows:*

$$R_{\mathcal{D}}(f) \leq \hat{R}_S(f) + \frac{cLVM^2}{m} + \Big(2cLVM^2 + 1 + 2c\sqrt{LV}M\Big)\sqrt{\frac{\ln\frac{1}{\delta}}{2m}}.$$

Proof. The constant E can be estimated by considering the following:

$$F(f) \leq F(\mathbf{0})$$

$$\frac{1}{2}\|\theta\|^2 + \frac{c}{m}\sum_{i=1}^{m}\max(0, 1 - y_i(\theta\mu_{\mathcal{L}}(x_i)^T)) \leq \frac{1}{2}\|\mathbf{0}\|^2 + \frac{c}{m}\sum_{i=1}^{m}\max(0, 1 - y_i(\mathbf{0}\mu_{\mathcal{L}}(x_i)^T))$$

$$\frac{1}{2}\|\theta\|^2 \leq c \tag{9}$$

$$\|\theta\|^2 \leq 2c$$

Eq. (9) is because $\forall a, b, c \in \mathbb{R}^+$, $a + b \leq c$ implies that $b \leq c$. Thus,

$$\begin{aligned} \ell(f, z) &= \max(0, 1 - y\theta\mu_{\mathcal{L}}(x)^T) \\ &\leq 1 + |\theta\mu_{\mathcal{L}}(x)^T| \\ &\leq 1 + \|\theta\| \, \|\mu_{\mathcal{L}}(x)\| \\ &\leq 1 + 2c\sqrt{LV}M = E \end{aligned} \tag{10}$$

Eq. (10) comes again from the Cauchy-Swartz inequality. □

5 Learning with Missing Views

Up to this section, we have made the implicit assumption that all the instances were observed in all the views. Because it is common in real-case scenarios that some points are observed only in a subset of views, we now illustrate how to adapt our formulation to the so-called missing-view setting.

The formulation from Eq. 3 is applicable only when all the points of the training and test sets are observed in all the views. To extend our method to the context of missing views, we apply a reconstruction step before learning. As we want to preserve the scalability of our approach, we do not impute missing values in the original input space: we rather design a dedicated method that imputes missing values by directly leveraging the information coming from the set of landmarks. We simply formulate our imputation as a Least Square over the known values as follows:

$$\arg\min_{R} \|M - RP\|^2_{\mathcal{F}'} \tag{11}$$

with M the $m \times LV$ matrix of projection values, P the $L \times LV$ matrix of projected landmarks, R the unknown $m \times L$ reconstruction matrix and $\|.\|_{\mathcal{F}'}$ the Frobenius norm considering only the non-missing values (in our case, the missing values are those of M). The problem from Eq. (11) boils down to learning linear combinations of landmark similarities over all the views and, for this reason, all the views of the landmarks need to be known. Doing so, we avoid estimating the actual missing features and we directly impute the view-dependent similarities between points and landmarks.

It is worth noting that each point projection is reconstructed independently and that the system is always (over-)determined for each point, as at least one block of size L of the point projection is known (at least one view's features are given) and the number of unknowns is L.

6 Experimental Results

In this section, we report and analyze the performances of our method w.r.t. the state of the art algorithms, in terms of both classification accuracy and training

and testing execution times. We perform two sets of experiments: (i) learning with complete views and (ii) learning with missing views. We will specifically study the behavior of MVL-SVM w.r.t. the number of landmarks keeping in mind that the larger the number of landmarks, the better the discriminatory power of the classifier, but the slower the learning process.

An implementation of our method, based on the Liblinear library [10], together with the other existing algorithms (when the codes are open-source) is available at https://github.com/vzantedeschi/multiviewLSVM.

6.1 Datasets, Methods and Experimental Setup

For these experiments, we employ two multi-class datasets that provide multi-view representations of the instances:

- Flower17 [1] contains 1360 pictures of 17 categories of flowers, which come with 7 different distance matrices between pictures (*i.e.* the 7 views);
- uWaveGesture [8] is formed by 4478 vectors describing 8 different gestures as captured by 3 accelerometers (the 3 views).

In order to prove the significance of embedding the datasets in a single space, we compare methods that learn a single classifier on a latent space and methods that learn a set of single-views classifiers. Moreover, we principally compare MVL-SVM to SVM-based approaches, to highlight the interest of using landmark-mappings. Multi-class classification is carried-out through the one-vs-all procedure.

We report the results of the following baselines:

- **MVML** [14] that optimizes over both the classifier and the metric matrix, and which is designed to make the most of the between-view and within-view relationships;
- the co-regularization technique **SVM-2k** [11], which regularizes over the predictions enforcing their smoothness. Originally designed for 2 view learning, we adapted this algorithm to work with $V \geq 2$ views by learning a SVM-2k for every pair of views and combining their predictions using a majority vote;
- **SVMs** which consists in learning a Kernel-SVM per view and aggregating their predictions by majority vote.

All the previous methods, and ours, utilize the Radial Basis Function (RBF, squared exponential) kernel for comparing the points, with a radius that we fixed equal to the square root of the number of features. We make use of the 3 train-val-test splits provided for Flower17, and of the train-test split for uWaveGesture, tuning by cross-validation over the training set. We repeat each experiment 5 times, reporting the average test value and its standard deviation when it is not null. For MVL-SVM, at each iteration we randomly select a new set of landmarks to underline how the chosen landmarks affect the expressiveness of

[1] http://www.robots.ox.ac.uk/∼vgg/data/flowers/17/

the latent space. We tune the hyper-parameters of the methods by grid-search over the following set-values: for MVML, we evaluate $\lambda \in \{10^{-8}, \dots, 10\}$ and $\eta \in \{10^{-3}, \dots, 10^{2}\}$, as indicated in the original paper; for SVM-2K, we consider c_1, c_2 and $d \in \{10^{-4}, \dots, 1\}$ and fix $\varepsilon = 10^{-3}$; for both SVMs and MVL-SVM, we consider $c \in \{10^{-3}, \dots, 10^{4}\}$.

6.2 Learning with Complete Views

In this first experiment, we compare the methods on complete datasets, where all the points are observed on all the views. In particular, we study the impact of the dimensionality of our latent space, controlled by the number of landmarks, on the performances of MVL-SVM. As the rank of the Nyström-approximated Gram matrix of MVML and the number of landmarks of MVL-SVM are comparable, because they both measure the number of computed similarities, we draw them on the same axis and compare these two methods also on this criterion. We explore values from 10 to the size of the training set (validation set not included). Because of MVML's huge computational complexity (see Fig. 3), its results in Figure 2 are truncated at a smaller approximation level.

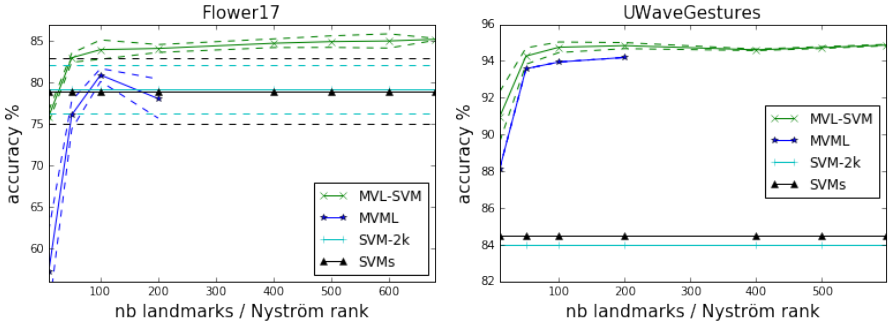


Fig. 2: Average test accuracies (with standard deviations) w.r.t. the number of landmarks/Nyström rank.

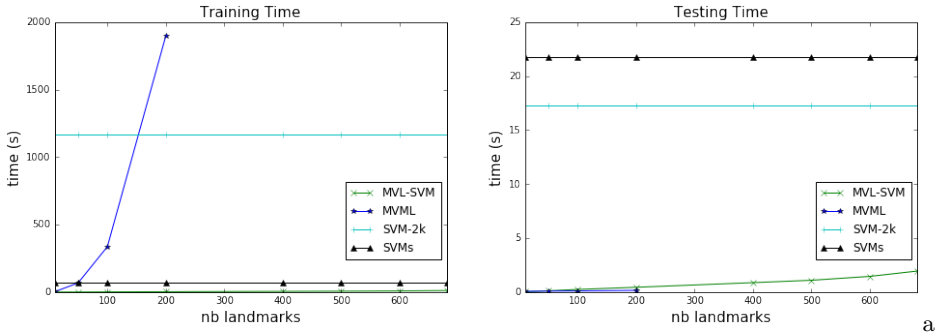
Figure 2 shows the test accuracies on both datasets. It is manifest how working on a latent space is of great benefit: both methods that exploit this idea show significant better test accuracies than those that learn view-specific classifiers, especially for the uWaveGestures dataset where views are very complementary. It is worth noting that MVL-SVM is able to reach the best performance even with a small number of landmarks (10 for uWaveGestures and 50 for Flower17).

Moreover, majority-vote techniques seem more sensitive to the choice of points selected for training (see Flower17) than MVL-SVM, which is consistently robust to the variations in the set of landmarks.

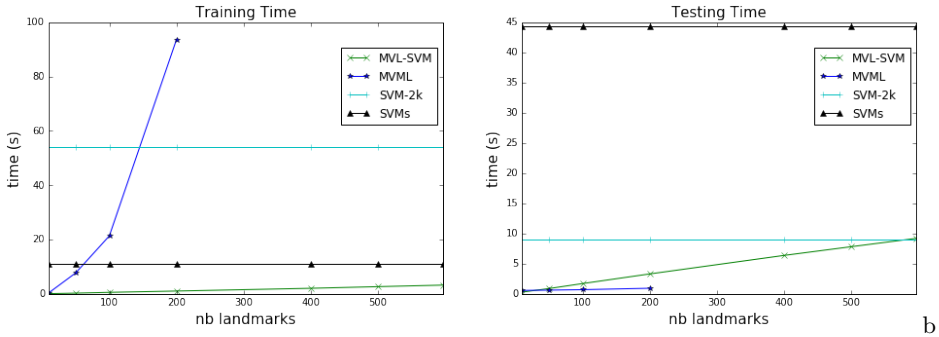
Figures 3 and 4 highlight the other important advantage of MVL-SVM: its fastness. At training time, MVL-SVM's execution time is linear in the number of

landmarks and several magnitudes smaller than baselines' times. At test time, MVL-SVM is only slightly beaten by MVML, but it could be accounted to optimizations in the code. Notice how learning multiple learners (SVM-2k and SVMs) considerably slows down both training and test steps. Handling multiple models is, indeed, a heavy overhead.

Overall, MVL-SVM achieves significantly better test accuracy that the considered baselines, even with a limited number of landmarks, while training several order of magnitude faster and being comparably fast at test time.



(a) Flower17.



(b) uWaveGestures.

Fig. 3: Training and test times w.r.t. the number of landmarks. MVL-SVM is very fast and scales linearly with the number of landmarks, unlike MVML.

6.3 Learning with Missing Views

With this second series of experiments, we aim at evaluating the validity of the imputation technique proposed in Section 5. We make use of the two previously described datasets that we modify for the current task: we drop random views of their points with a ratio of missing views over total number of views (mV)

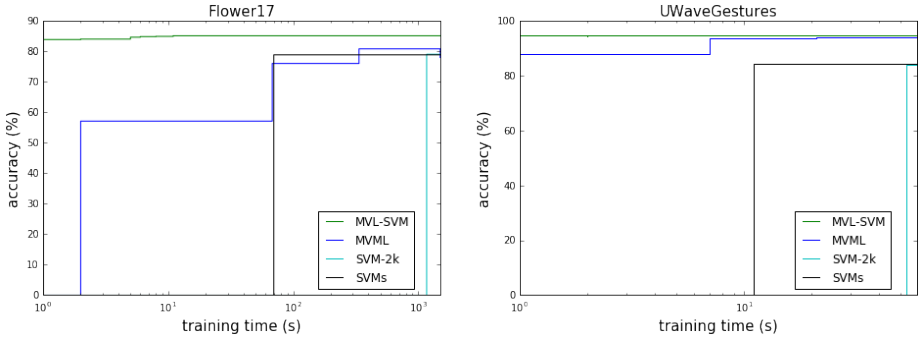


Fig. 4: Average test accuracies w.r.t. the training time. Compared to the other methods, MVL-SVM reaches high accuracy even with very low computational budget. The x axis is in logarithmic scale.

varying in the interval $[0, 0.5]$. For MVL-SVM, the number of landmarks L is fixed to 200. In Figure 5, we draw the test accuracies in this new setting for both datasets, comparing MVL-SVM to **SVMs** both with and without any reconstruction technique. When no imputation is applied as preprocessing, the points with missing views are dropped for MVL-SVM, while for **SVMs**, as it deals with a view at a time, they are still used for training the view-specific models corresponding to the available views. For **SVMs**, we impute the missing values using Graph Laplacian imputation [24] by fixing the Gram matrix of the view with the most points as the reference view for reconstructing all the other views. Remark that the points missing from the reference view will not have their views reconstructed, which might explain the drop in accuracy of **SVMs** for a ratio bigger than 0.3 for Flower17.

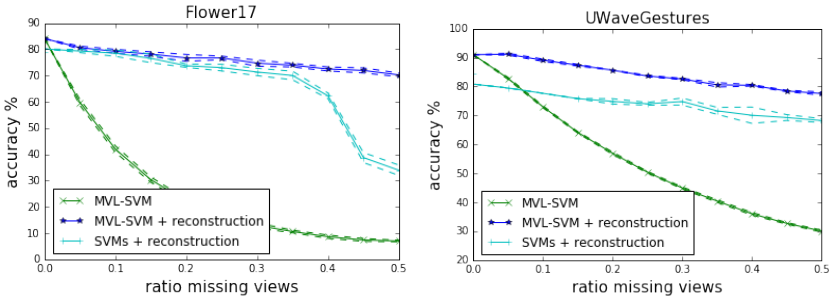


Fig. 5: Test accuracies (with standard deviations) w.r.t. the ratio of missing views, using 200 landmarks for MVL-SVM. Imputation of missing value is critical for MVL-SVM to achieve good accuracy when facing missing views. Thanks to the proposed missing value imputation, MVL-SVM remains more accurate even in case of missing views.

Notice how preprocessing the dataset is fundamental for applying MVL-SVM to the missing-view scenario. This is not surprising as, using a latent space, we can train the model only on points observed in all the views. Even if the accuracy of both methods (with reconstruction) still slightly decays with the ratio of missing views, the gain in performances is dramatic.

7 Conclusion and Perspectives

We proposed MVL-SVM, an effective technique for tackling multi-view problems, training a linear-SVM on a landmark-induced latent space, unifying the view information, constructed by applying non-linear multi-view similarity estimates between the instances and a set of randomly selected landmarks. We additionally introduced an imputation technique making it suited for the missing-view context. We also showed MVL-SVM's validity, from both theoretical and empirical point of view: we derived a generalization bound using the uniform stability framework, and we showed empirically that our approach outperforms the considered baselines in terms of accuracy while being several order of magnitude faster. MVL-SVM rely on a set of landmarks that is shared for all views. According to the application at hand, it might be interesting to consider more landmarks in some of the views, and future work includes considering different landmarks in the views. Additionally, by using block-sparsity in the final linear-separator, automated landmark selection could be achieved, giving MVL-SVM an even better test-time execution speed. The missing view imputation technique can also be improved by considering a joint optimization of the reconstruction matrix R and the linear classifier (θ, b).

References

1. Amini, M., Usunier, N., Goutte, C.: Learning from multiple partially observed views-an application to multilingual text categorization. In: Advances in neural information processing systems. pp. 28–36 (2009)
2. Bakır, G., Bottou, L., Weston, J.: Breaking svm complexity with cross training. Advances in neural information processing systems **17**, 81–88 (2005)
3. Bellet, A., Habrard, A., Sebban, M.: A survey on metric learning for feature vectors and structured data (2013), http://arxiv.org/abs/1306.6709
4. Bhadra, S., Kaski, S., Rousu, J.: Multi-view kernel completion. Machine Learning **106**(5), 713–739 (2017)
5. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. pp. 92–100. ACM (1998)
6. Bordes, A., Bottou, L., Gallinari, P.: Sgd-qn: Careful quasi-newton stochastic gradient descent. Journal of Machine Learning Research **10**(Jul), 1737–1754 (2009)
7. Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research **2**(Mar), 499–526 (2002)
8. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The ucr time series classification archive (July 2015), www.cs.ucr.edu/~eamonn/time_series_data/

9. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research **9**, 1871–1874 (2008)
11. Farquhar, J., Hardoon, D., Meng, H., Shawe-taylor, J.S., Szedmak, S.: Two view learning: Svm-2k, theory and practice. In: Advances in neural information processing systems. pp. 355–362 (2006)
12. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. Journal of machine learning research **12**(Jul), 2211–2268 (2011)
13. Goyal, A., Morvant, E., Germain, P., Amini, M.R.: Pac-bayesian analysis for a two-step hierarchical multiview learning approach. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 205–221. Springer (2017)
14. Huusari, R., Kadri, H., Capponi, C.: Multi-view metric learning in vector-valued kernel spaces. In: Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. pp. 415–424 (2018)
15. Kadri, H., Ayache, S., Capponi, C., Koço, S., Dupé, F.X., Morvant, E.: The multi-task learning view of multimodal data. In: Asian Conference on Machine Learning. pp. 261–276 (2013)
16. Ladicky, L., Torr, P.: Locally linear support vector machines. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 985–992 (2011)
17. Micchelli, C.A., Pontil, M.: On learning vector-valued functions. Neural computation **17**(1), 177–204 (2005)
18. Minh, H.Q., Bazzani, L., Murino, V.: A unifying framework for vector-valued manifold regularization and multi-view learning. In: ICML (2). pp. 100–108 (2013)
19. Minh, H.Q., Bazzani, L., Murino, V.: A unifying framework in vector-valued reproducing kernel hilbert spaces for manifold regularization and co-regularized multi-view learning. Journal of Machine Learning Research **17**(25), 1–72 (2016)
20. Steinwart, I.: Sparseness of support vector machines. Journal of Machine Learning Research **4**(Nov), 1071–1105 (2003)
21. Sun, S.: Multi-view laplacian support vector machines. In: International Conference on Advanced Data Mining and Applications. pp. 209–222. Springer (2011)
22. Sun, S.: A survey of multi-view machine learning. Neural Computing and Applications **23**(7-8), 2031–2038 (2013)
23. Tang, J., Tian, Y., Zhang, P., Liu, X.: Multiview privileged support vector machines. IEEE transactions on neural networks and learning systems (2017)
24. Trivedi, A., Rai, P., Daumé III, H., DuVall, S.L.: Multiview clustering with incomplete views. In: NIPS Workshop (2010)
25. Williams, C.K., Seeger, M.: Using the nyström method to speed up kernel machines. In: Advances in neural information processing systems. pp. 682–688 (2001)
26. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. CoRR **abs/1304.5634** (2013), http://arxiv.org/abs/1304.5634
27. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. Michigan State Universiy **2**(2) (2006)
28. Zantedeschi, V., Emonet, R., Sebban, M.: L^3-SVMs: Landmarks-based linear local support vectors machines. arXiv preprint arXiv:1703.00284 (2017)
29. Zhao, J., Xie, X., Xu, X., Sun, S.: Multi-view learning overview: Recent progress and new challenges. Information Fusion **38**, 43–54 (2017)