Multi-view Machines

Bokai Cao
University of Illinois at Chicago
Chicago, IL, USA
caobokai@uic.edu

Hucheng Zhou
Microsoft Research Asia
Beijing, China
huzho@microsoft.com

Philip S. Yu
University of Illinois at Chicago
Chicago, IL, USA
psyu@cs.uic.edu

ABSTRACT

For a learning task, data can usually be collected from different sources or be represented from multiple views. For example, laboratory results from different medical examinations are available for disease diagnosis, and each of them can only reflect the health state of a person from a particular aspect/view. Therefore, different views provide complementary information for learning tasks. An effective integration of the multi-view information is expected to facilitate the learning performance. In this paper, we propose a general predictor, named multi-view machines (MVMs), that can effectively include all the possible interactions between features from multiple views. A joint factorization is embedded for the full-order interaction parameters which allows parameter estimation under sparsity. Moreover, MVMs can work in conjunction with different loss functions for a variety of machine learning tasks. A stochastic gradient descent method is presented to learn the MVM model. We further illustrate the advantages of MVMs through comparison with other methods for multi-view classification, including support vector machines (SVMs), support tensor machines (STMs) and factorization machines (FMs).

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithm, Experimentation, Performance

Keywords

Data Mining, Multi-view Learning, Factorization

1. INTRODUCTION

In the era of big data, information is available not only in great volume but also in multiple representations/views from a variety of sources or feature subsets. Generally, different views provide complementary information for learning tasks. Thus, multi-view learning can facilitate the learning process and is prevalent in a wide range of application domains. For example, to fulfil an accurate disease diagnosis, one should consider laboratory results from different medical examinations, including clinical, imaging, immunologic, serologic and cognitive measures. For the business on the web, it is critical to estimate the probability that the display of an ad to a specific user when s/he searches for a query will lead to a click. This process involves three entities: users, ads, and queries. An effective integration of the features describing these different entities is directly related to a precise targeting of the advertising system.

One of the key challenges of multi-view learning is to model the interactions between different views, wherein complementary information is contained. Conventionally, multiple kernel learning algorithms combine kernels that naturally correspond to different views to improve the learning performance [5]. Basically, the coefficients are learned based on the usefulness/informativeness of the associated views, and thus the correlations are considered at the view-level. These approaches, however, fail to explicitly explore the correlations between features. In contrast to modeling on views, another direction for modeling multi-view data is to directly consider the abundant correlations between features from different views.

In this paper, we propose a novel model for multi-view learning, called multi-view machines (MVMs). The main advantages of MVMs are outlined as follows:

- MVMs include all the possible interactions between features from multiple views, ranging from the first-order interactions (*i.e.*, contributions of single features) to the highest order interactions (*i.e.*, contributions of combinations of features from each view).
- MVMs jointly factorize the interaction parameters in different orders to allow parameter estimation under sparsity.
- MVMs are a general predictor that can work with different loss functions (*e.g.*, square error, hinge loss, logit loss) for a variety of machine learning tasks.

2. MULTI-VIEW CLASSIFICATION

We first state the problem of multi-view classification and introduce the notation. Table 1 lists some basic symbols that will be used throughout the paper.

Table 1: Symbols.

Symbol	Definition and Description
s	each lowercase letter represents a scale
$\mathbf{v}$	each boldface lowercase letter represents a vector
$\mathbf{M}$	each boldface capital letter represents a matrix
$\mathcal{T}$	each calligraphic letter represents a tensor, set or space
$\langle\cdot,\cdot\rangle$	denotes inner product
$\circ$	denotes tensor product or outer product
$\times_k$	denotes mode-k product
$\lvert\cdot\rvert$	denotes absolute value
$\lVert\cdot\rVert_F$	denotes (Frobenius) norm of vector, matrix or tensor

Suppose each instance has representations in m different views, *i.e.*, $\mathbf{x}^T = \left(\mathbf{x}^{(1)^T}, ..., \mathbf{x}^{(m)^T}\right)$, where $\mathbf{x}^{(v)} \in \mathbb{R}^{I_v}$, I_v is the dimensionality of the v-th view. Let $d = \sum_{v=1}^m I_v$, so $\mathbf{x} \in \mathbb{R}^d$. Considering the problem of click through rate (CTR) prediction for advertising display, for example, an instance corresponds to an *impression* which involves a user, an ad, and a query. Therefore, if $\mathbf{x}^T = \left(\mathbf{x}^{(1)^T}, \mathbf{x}^{(2)^T}, \mathbf{x}^{(3)^T}\right)$ is an impression, $\mathbf{x}^{(1)}$ contains information of the user profile, $\mathbf{x}^{(2)}$ is associated with the ad information, and $\mathbf{x}^{(3)}$ is the description from the query aspect. The result of an impression is *click* or *non-click*.

Given a training set with n labeled instances represented from m views: $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, ..., n\}$, in which $\mathbf{x}_i^T = \left(\mathbf{x}_i^{(1)^T}, ..., \mathbf{x}_i^{(m)^T}\right)$ and $y_i \in \{-1, 1\}$ is the class label of the i-th instance. For CTR prediction problem, $y = 1$ denotes *click* and $y = -1$ denotes *non-click* in an impression. The task of multi-view classification is to learn a function $f : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_m} \to \{-1, 1\}$ that correctly predicts the label of a test instance.

In addition, we introduce the concept of tensors which are higher order arrays that generalize the notions of vectors (first-order tensors) and matrices (second-order tensors), whose elements are indexed by more than two indexes. We state the definition of tensor product and mode-k product which will be used to formulate our proposed model.

Definition 2.1 (Tensor Product or Outer Product). *The tensor product $\mathcal{X} \circ \mathcal{Y}$ of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_m}$ and another tensor $\mathcal{Y} \in \mathbb{R}^{I'_1 \times \cdots \times I'_{m'}}$ is defined by*

$$(\mathcal{X} \circ \mathcal{Y})_{i_1, ..., i_m, i'_1, ..., i'_{m'}} = x_{i_1, ..., i_m} y_{i'_1, ..., i'_{m'}} \qquad (1)$$

for all index values.

Definition 2.2 (Mode-k Product). *The mode-k product $\mathcal{X} \times_k \mathbf{M}$ of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_m}$ and a matrix $\mathbf{M} \in \mathbb{R}^{I'_k \times I_k}$ is defined by*

$$(\mathcal{X} \times_k \mathbf{M})_{i_1, ..., i_{k-1}, j, i_{k+1}, ..., i_m} = \sum_{i_k=1}^{I_k} x_{i_1, ..., i_m} m_{j, i_k} \qquad (2)$$

for all index values.

3. MULTI-VIEW MACHINE MODEL

3.1 Model Formulation

The key challenge of multi-view classification is to model the interactions between features from different views, wherein complementary information is contained. Here, we consider nesting all interactions up to mth-order between m views:

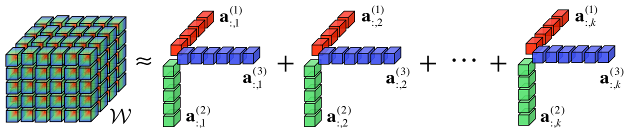

Figure 1: CP factorization. The third-order ($m = 3$) tensor $\mathcal{W}$ is approximated by k rank-one tensors. The f-th factor tensor is the tensor product of three vectors, *i.e.*, $\mathbf{a}^{(1)}_{:,f} \circ \mathbf{a}^{(2)}_{:,f} \circ \mathbf{a}^{(3)}_{:,f}$.

$$\begin{aligned}\hat{y} = & \underbrace{\beta_0}_{\text{bias}} + \underbrace{\sum_{v=1}^{m} \sum_{i_v=1}^{I_v} \beta_{i_v}^{(v)} x_{i_v}^{(v)}}_{\text{first-order interactions}} \\ & + \underbrace{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \beta_{i_1, i_2}^{(1,2)} x_{i_i}^{(1)} x_{i_2}^{(2)} + \cdots + \sum_{i_{m-1}=1}^{I_{m-1}} \sum_{i_m=1}^{I_m} \beta_{i_{m-1}, i_m}^{(m-1,m)} x_{i_{m-1}}^{(m-1)} x_{i_m}^{(m)}}_{\text{second-order interactions}} \\ & + \cdots + \underbrace{\sum_{i_1=1}^{I_1} \cdots \sum_{i_m=1}^{I_m} \beta_{i_1, ..., i_m} \left(\prod_{v=1}^{m} x_{i_v}^{(v)}\right)}_{m\text{th-order interactions}} \qquad (3)\end{aligned}$$

Let us add an extra feature with constant value 1 to the feature vector $\mathbf{x}^{(v)}$, *i.e.*, $\mathbf{z}^{(v)^T} = (\mathbf{x}^{(v)^T}, 1) \in \mathbb{R}^{I_v+1}, \forall v = 1, ..., m$. Then, Eq. (3) can be effectively rewritten as:

$$\hat{y} = \sum_{i_1=1}^{I_1+1} \cdots \sum_{i_m=1}^{I_m+1} w_{i_1, ..., i_m} \left(\prod_{v=1}^{m} z_{i_v}^{(v)}\right) \qquad (4)$$

where $w_{I_1+1, ..., I_m+1} = \beta_0$ and $w_{i_1, ..., i_m} = \beta_{i_1, ..., i_m}, \forall i_v \le I_v$. For $w_{i_1, ..., i_m}$ with some indexes satisfying $i_v = I_v + 1$, it encodes lower order interaction between views whose $i_v \le I_v$. Hereinafter, let $w_{i_p}^{(p)}$ denote $w_{i_1, ..., i_m}$ where only $i_p \le I_p$ and $i_v = I_v + 1, v \ne p$, and let $w_{i_p, i_q}^{(p,q)}$ denote $w_{i_1, ..., i_m}$ where $i_p \le I_p$, $i_q \le I_q$ and $i_v = I_v + 1, v \notin \{p, q\}$ for other $m - 2$ views, *etc.*

The number of parameters in Eq. (4) is $\prod_{v=1}^m (I_v + 1)$, which can make the model prone to overfitting and ineffective on sparse data. Therefore, we assume that the effect of interactions has a low rank and the mth-order tensor $\mathcal{W} = \{w_{i_1, ..., i_m}\} \in \mathbb{R}^{(I_1+1) \times \cdots \times (I_m+1)}$ can be factorized into k factors:

$$\mathcal{W} = \mathbf{C} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_m \mathbf{A}^{(m)} \qquad (5)$$

where $\mathbf{A}^{(v)} \in \mathbb{R}^{(I_v+1) \times k}$, and $\mathbf{C} \in \mathbb{R}^{k \times \cdots \times k}$ is the identity tensor, *i.e.*, $c_{i_1, ..., i_m} = \delta(i_1 = \cdots = i_m)$. Basically, it is a CANDECOMP/PARAFAC (CP) factorization [4] as shown in Figure 1, with element-wise notation $w_{i_1, ..., i_m} = \sum_{f=1}^k \prod_{v=1}^m a_{i_v, f}^{(v)}$. The number of model parameters is reduced to $k \sum_{v=1}^m (I_v + 1) = k(m + d)$. It transforms Eq. (4) into:

$$\hat{y} = \sum_{i_1=1}^{I_1+1} \cdots \sum_{i_m=1}^{I_m+1} \left(\prod_{v=1}^{m} z_{i_v}^{(v)}\right) \left(\sum_{f=1}^{k} \prod_{v=1}^{m} a_{i_v, f}^{(v)}\right) \qquad (6)$$

We name this model as multi-view machines (MVMs). As shown in Figure 2, the full-order interactions between mul-

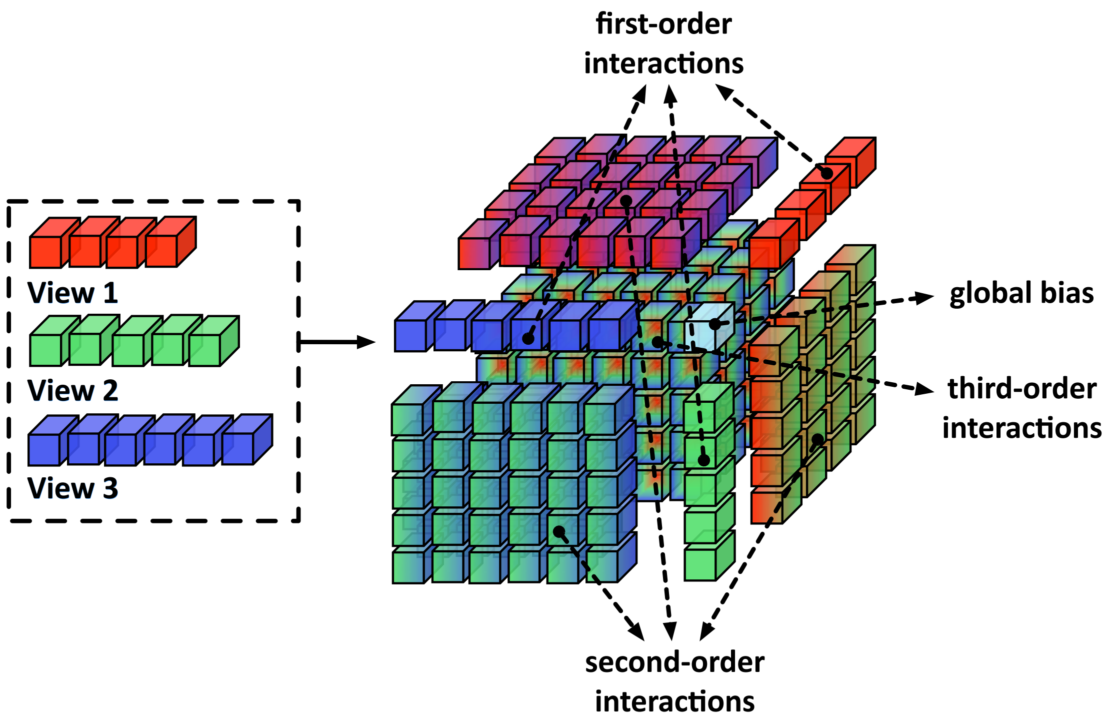

Figure 2: Multi-view machines. All the interactions of different orders between multiple views are modeled in a single tensor and share the same set of latent factors.

tiple views are modeled in a single tensor, and they are factorized collectively. The model parameters that have to be estimated are:

$$\mathbf{A}^{(v)} \in \mathbb{R}^{(I_v+1)\times k},\ v = 1, ..., m \tag{7}$$

where the i_v-th row $\mathbf{a}_{i_v}^{(v)T} = (a_{i_v,1}^{(v)}, ..., a_{i_v,k}^{(v)})$ within $\mathbf{A}^{(v)}$ describes the i_v-th feature in the v-the view with k factors. Let the last row $\mathbf{a}_{I_v+1}^{(v)}{}^T$ denote the *bias factor* from the v-th view, since it is always combined with $z_{I_v+1}^{(v)} = 1$ in Eq. (6). Hence,

$$w_{I_1+1,...,I_m+1} = \sum_{f=1}^{k} \prod_{v=1}^{m} a_{I_v+1,f}^{(v)} \tag{8}$$

is the *global bias*, denoted as w_0 hereinafter.

Moreover, MVMs are flexible in the order of interactions of interests. That is to say, when there are too many views available for a learning task and interactions between some of them may obviously be physically meaningless, or sometimes the very high order interactions may be intuitively uninterpretable, it is not desirable to include these potentially redundant interactions in the model. In such scenarios, one can (1) partition (overlapping) groups of views, (2) construct multiple MVMs on these view groups where the full-order interactions within each group are included, and (3) implement a coupled matrix/tensor factorization [3]. This implementation excludes those cross-group interactions. Although MVMs are feasible in any order of interactions, that is outside the scope of this paper; our focus is on investigating how to effectively explore the full-order interactions within a given set of views.

3.2 Time Complexity

Next, we show how to make MVMs applicable from a computational point of view. The straightforward time complexity of Eq. (6) is $O(k\prod_{v=1}^{m}(I_v+1))$. However, we observe that there is no model parameter which directly depends on the interactions between variables (*e.g.*, a parameter with an index $(i_1, ..., i_m)$), due to the factorization of the interactions. Therefore, the time complexity can be largely reduced.

LEMMA 3.1. *The model equation of MVMs can be computed in linear time $O(k(m+d))$.*

PROOF. The interactions in Eq. (6) can be reformulated as:

$$\begin{aligned}&\sum_{i_1=1}^{I_1+1}\cdots\sum_{i_m=1}^{I_m+1}\left(\prod_{v=1}^{m} z_{i_v}^{(v)}\right)\left(\sum_{f=1}^{k}\prod_{v=1}^{m} a_{i_v,f}^{(v)}\right)\\ =&\sum_{f=1}^{k}\sum_{i_1=1}^{I_1+1}\cdots\sum_{i_m=1}^{I_m+1}\left(\prod_{v=1}^{m} z_{i_v}^{(v)} a_{i_v,f}^{(v)}\right)\\ =&\sum_{f=1}^{k}\left(\sum_{i_1=1}^{I_1+1} z_{i_1}^{(1)} a_{i_1,f}^{(1)}\right)\cdots\left(\sum_{i_m=1}^{I_m+1} z_{i_m}^{(m)} a_{i_m,f}^{(m)}\right)\end{aligned} \tag{9}$$

This equation has only linear complexity in both k and I_v. Thus, its time complexity is $O(k(m+d))$, which is in the same order of the number of parameters in the model. □

4. LEARNING MULTI-VIEW MACHINES

To learn the parameters in MVMs, we consider the following regularization framework:

$$\underset{\Theta}{\operatorname{argmin}} \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathcal{L}(\hat{y}(\mathbf{x}|\Theta), y) + \lambda\Omega(\Theta) \tag{10}$$

where Θ represents all the model parameters, $\mathcal{L}(\cdot)$ is the loss function, $\Omega(\cdot)$ is the regularization term, and λ is the trade-off between the empirical loss and the risk of overfitting.

Importantly, MVMs can be used to perform a variety of machine learning tasks, depending on the choices of the loss function. For example, to conduct regression, the square error is a popular choice:

$$\mathcal{L}^S(\hat{y}(\mathbf{x}|\Theta), y) = (\hat{y}(\mathbf{x}|\Theta) - y)^2 \tag{11}$$

and for classification problems, we can use the logit loss:

$$\mathcal{L}^L(\hat{y}(\mathbf{x}|\Theta), y) = \log(1 + \exp(-y \cdot \hat{y}(\mathbf{x}|\Theta))) \tag{12}$$

or the hinge loss:

$$\mathcal{L}^H(\hat{y}(\mathbf{x}|\Theta), y) = \max(0, 1 - y \cdot \hat{y}(\mathbf{x}|\Theta)) \tag{13}$$

The regularization term is chosen based on our prior knowledge about the model parameters. Typically, we can apply L2-norm:

$$\Omega^{L2}(\Theta) = \|\Theta\|_2^2 = \sum\nolimits_i \theta_i^2 \tag{14}$$

or L1-norm:

$$\Omega^{L1}(\Theta) = \|\Theta\|_1 = \sum\nolimits_i |\theta_i| \approx \sum\nolimits_i \sqrt{\theta_i^2 + \epsilon^2} \tag{15}$$

where ϵ is a very small number to make the L1-norm term differentiable.

The model parameters $\Theta = \{\mathbf{A}^{(v)}|\ v = 1, ..., m\}$ can be learned efficiently by alternating least square (ALS), stochastic gradient descent (SGD), L-BFGS, *etc.*, for a variety of loss functions, including square error, hinge loss, logit loss, *etc.* From Eq. (9), the gradient of the MVM model is:

$$\begin{aligned}\frac{\partial \hat{y}(\mathbf{x}|\Theta)}{\partial \theta} =& z_{i_v}^{(v)} \left(\sum_{i_1=1}^{I_1+1} z_{i_1}^{(1)} a_{i_1,f}^{(1)}\right) \cdots \left(\sum_{i_{v-1}=1}^{I_{v-1}+1} z_{i_{v-1}}^{(v-1)} a_{i_{v-1},f}^{(v-1)}\right) \\ & \left(\sum_{i_{v+1}=1}^{I_{v+1}+1} z_{i_{v+1}}^{(v+1)} a_{i_{v+1},f}^{(v+1)}\right) \cdots \left(\sum_{i_m=1}^{I_m+1} z_{i_m}^{(m)} a_{i_m,f}^{(m)}\right)\end{aligned} \tag{16}$$

where $\theta = a_{i_v,f}^{(v)}$, and $z_{i_v}^{(v)} = 1$ if $i_v = I_v + 1$, otherwise $z_{i_v}^{(v)} = x_{i_v}^{(v)}$. It validates that MVMs possess the multilinearity property, because the gradient along θ is independent of the value of θ itself.

Note that in Eq. (16), the sum $\sum_{i_v=1}^{I_v+1} z_{i_v}^{(v)} a_{i_v,f}^{(v)}$ can be precomputed and reused for updating the f-th factor of all the features. Hence, each gradient can be computed in $O(m)$. In an iteration, including the precomputation time, all the $k(m + d)$ parameters can be updated in $O(mk(m + d))$. It can be even reduced under sparsity, where most of the elements in $\mathbf{x}$ (or $\mathbf{z}$) are 0 and thus, the sums have only to be computed over the non-zero elements.

It is straightforward to embed Eq. (16) into the gradient of the loss functions *e.g.*, Eqs. (11)-(13), for direct optimization, as follows:

$$\frac{\partial \mathcal{L}^S(\hat{y}(\mathbf{x}|\Theta), y)}{\partial \theta} = 2(\hat{y}(\mathbf{x}|\Theta) - y) \cdot \frac{\partial \hat{y}(\mathbf{x}|\Theta)}{\partial \theta} \tag{17}$$

$$\frac{\partial \mathcal{L}^L(\hat{y}(\mathbf{x}|\Theta), y)}{\partial \theta} = \frac{-y \exp(-y \cdot \hat{y}(\mathbf{x}|\Theta))}{1 + \exp(-y \cdot \hat{y}(\mathbf{x}|\Theta))} \cdot \frac{\partial \hat{y}(\mathbf{x}|\Theta)}{\partial \theta} \tag{18}$$

$$\frac{\partial \mathcal{L}^H(\hat{y}(\mathbf{x}|\Theta), y)}{\partial \theta} = \begin{cases} -y \cdot \frac{\partial \hat{y}(\mathbf{x}|\Theta)}{\partial \theta} & \text{if } y \cdot \hat{y}(\mathbf{x}|\Theta) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Moreover, the gradient of the regularization term $\Omega(\Theta)$ can be derived:

$$\frac{\partial \Omega^{L2}(\Theta)}{\partial \theta} = 2\theta \tag{20}$$

$$\frac{\partial \Omega^{L1}(\Theta)}{\partial \theta} = \frac{\theta}{\sqrt{\theta^2 + \epsilon^2}} \tag{21}$$

The SGD optimization method for MVMs is summarized in Algorithm 1, where the model parameters are first initialized from a zero-mean normal distribution with standard deviation σ, and the gradients in line 8 can be computed according to Eqs. (11)-(13) and Eqs. (20)-(21). Moreover, rather than specifying a learning rate η beforehand, we can use a line search to determine it in the optimization process. The regularization parameter λ can be searched on a held-out validation set. Considering the number of factors k, the performance can usually be improved with larger k, at the cost of more parameters which can make the learning much harder in terms of both runtime and memory [10].

Algorithm 1 Stochastic Gradient Descent for MVMs

```
Input: Training data $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, ..., n\}$, number
    of factors $k$, regularization parameter $\lambda$, learning rate $\eta$,
    standard deviation $\sigma$
Output: Model parameters $\Theta = \{\mathbf{A}^{(v)} \in \mathbb{R}^{(I_v+1)\times k}|\ v =$
    $1, ..., m\}$
 1: Initialize $\mathbf{A}^{(v)} \sim \mathcal{N}(0, \sigma)$
 2: repeat
 3:   for $(\mathbf{x}, y) \in \mathcal{D}$ do
 4:     for $v := 1$ to $m$ do
 5:       for $i_v := 1$ to $I_v + 1$ do
 6:         if $z_{i_v}^{(v)} \neq 0$ then
 7:           for $f := 1$ to $k$ do
 8:             $\theta \leftarrow \theta - \eta(\frac{\partial \mathcal{L}(\hat{y}(\mathbf{x}|\Theta),y)}{\partial \theta} + \lambda \frac{\partial \Omega(\Theta)}{\partial \theta})$ where
                $\theta = a_{i_v}^{(v)}$
 9:           end for
10:         end if
11:       end for
12:     end for
13:   end for
14: until convergence
```

5. RELATED WORK

In this section, we discuss and compare our proposed MVM model with other methods (and extensions) for multi-view classification, including support vector machines (SVMs), support tensor machines (STMs) and factorization machines (FMs).

5.1 SVM Model

Vapnik introduced support vector machines (SVMs) [9] based on the maximum-margin hyperplane. Essentially, SVMs

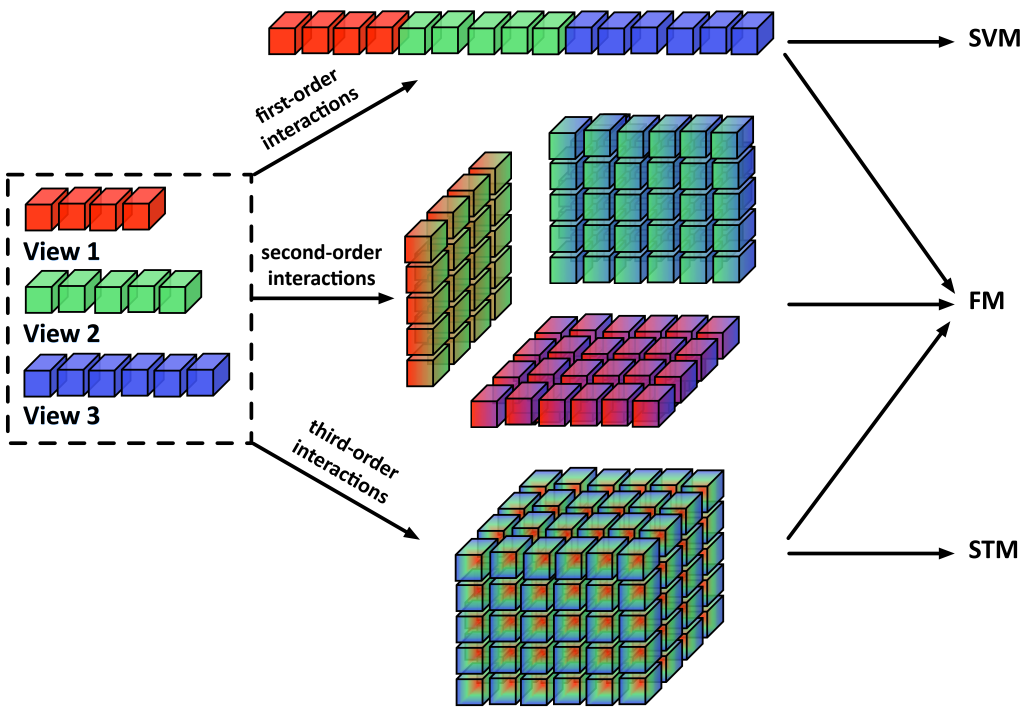

Figure 3: Related work (and extensions) on modeling the interactions between multiple views. In general, the linear SVM model is limited to the first-order interactions; the STM model explores only the highest order interactions; in spite of including all the interactions in different orders, the FM model is not sufficiently factorized compared to our proposed MVM model.

integrate the hinge loss and the L2-norm regularization. The decision function with a linear kernel is[1]:

$$\hat{y} = w_0 + \sum_{i=1}^{d} w_i x_i \tag{22}$$

In the multi-view setting, $\mathbf{x}$ is simply a concatenation of features from different views, *i.e.*, $\mathbf{x}^T = \left(\mathbf{x}^{(1)T}, ..., \mathbf{x}^{(m)T}\right)$, as shown in Figure 3. Thus, Eq. (22) is equivalent to:

$$\hat{y} = w_0 + \sum_{v=1}^{m} \sum_{i_v=1}^{I_v} w_{i_v}^{(v)} x_{i_v}^{(v)} \tag{23}$$

Obviously, no interactions between views are explored in Eq. (23). By restricting $i_v = I_v + 1$ for any $m-1$ indexes of $w_{i_1,\ldots,i_m}$ in Eq. (4), *i.e.*, removing factorization and higher order interactions from MVMs, we obtain the linear SVMs:

$$\begin{aligned}\hat{y} &= w_{I_1+1,\ldots,I_m+1} + \sum_{i_1=1}^{I_1} w_{i_1,I_2+1,\ldots,I_m+1} z_{i_1}^{(1)} \\ &\quad + \cdots + \sum_{i_m=1}^{I_m} w_{I_1+1,\ldots,I_{m-1}+1,i_m} z_{i_m}^{(m)} \\ &= w_0 + \sum_{i_1=1}^{I_1} w_{i_1}^{(1)} x_{i_1}^{(1)} + \cdots + \sum_{i_m=1}^{I_m} w_{i_m}^{(m)} x_{i_m}^{(m)}\end{aligned} \tag{24}$$

[1]The sign function is omitted, because the analysis and conclusions can easily extend to other generalized linear models, *e.g.*, logistic regression.

Throught the employment of a nonlinear kernel, SVMs can implicitly project data from the feature space into a more complex high-dimensional space, which allows SVMs to model higher order interactions between features. However, as discussed in [6], all interaction parameters of nonlinear SVMs are completely independent. In contrast, the interaction parameters of MVMs are collectively factorized and thus dependencies exist when interactions share the same feature.

For nonlinear SVMs, there must be enough instances $\mathbf{x} \in \mathcal{D}$ where $x_{i_p}^{(p)} \neq 0$ and $x_{i_q}^{(q)} \neq 0$ to reliably estimate the second-order interaction parameter $w_{i_p,i_q}^{(p,q)}$. The instances with either $x_{i_p}^{(p)} = 0$ or $x_{i_q}^{(q)} = 0$ cannot be used for estimating $w_{i_p,i_q}^{(p,q)}$. That is to say, on a sparse dataset where there are too few or even no cases for some higher order interactions, nonlinear SVMs are likely to degenerate into linear SVMs.

The factorization of interactions in Eq. (5) benefits MVMs for parameter estimation under sparsity, since the latent factor $\mathbf{a}_{i_p}^{(p)}$ can be learned from any instances whose $x_{i_p}^{(p)} \neq 0$, which allows the second-order interaction $w_{i_p,i_q}^{(p,q)}$ can be approximated from instances whose $x_{i_p}^{(p)} \neq 0$ *or* $x_{i_q}^{(q)} \neq 0$ rather than instances whose $x_{i_p}^{(p)} \neq 0$ *and* $x_{i_q}^{(q)} \neq 0$. Therefore, the interaction parameters in MVMs can be effectively learned without direct observations of such interactions in a training set of sparse data.

5.2 STM Model

Cao et al. investigated multi-view classification by modeling interactions between views as a tensor, *i.e.*, $\mathcal{X} = \mathbf{x}^{(1)} \circ \cdots \circ \mathbf{x}^{(m)} \in \mathbb{R}^{I_1 \times \cdots \times I_m}$ [2] and solved the problem in the framework of support tensor machines (STMs) [8]. Basically, as shown in Figure 3, only the highest order interactions are explored:

$$\hat{y} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_m=1}^{I_m} w_{i_1,\ldots,i_m} \left(\prod_{v=1}^{m} x_{i_v}^{(v)} \right) \tag{25}$$

where $w_{i_1,\ldots,i_m} = \prod_{v=1}^{m} w_{i_v}^{(v)}$, *i.e.*, a rank-one decomposition of the tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times \cdots \times I_m}$ [2].

However, estimating a lower order interaction (*e.g.*, a pairwise one) reliably is easier than estimating a higher order one, and lower order interactions can usually explain the data sufficiently [7, 1]. Thus, it is critical to include the lower order interactions in MVMs. Moreover, instead of a rank-one decomposition, we apply a higher rank decomposition of $\mathcal{W} \in \mathbb{R}^{(I_1+1) \times \cdots \times (I_m+1)}$ to capture more latent factors and thereby achieving a better approximation to the original interaction parameters.

5.3 FM Model

Rendle introduced factorization machines (FMs) [6] that combine the advantages of SVMs with factorization models. The model equation for a second-order FM is as follows:

$$\hat{y} = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i x_j \tag{26}$$

where $d = \sum_{v=1}^{m} I_v$ and $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{f=1}^{k} v_{i,f} v_{j,f}$.

However, the pairwise interactions between all the features are included in FMs without consideration of the view segmentation. In the multi-view setting, there can be redundant correlations between features within the same view which are thereby unworthy of consideration. The coupled group lasso model proposed in [10] is essentially an application of the second-order FMs to multi-view classification. To achieve this purpose, we can simply modify Eq. (26) as:

$$\begin{aligned}\hat{y} = w_0 &+ \sum_{v=1}^{m} \sum_{i_v=1}^{I_v} w_{i_v}^{(v)} x_{i_v}^{(v)} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \left\langle \mathbf{v}_{i_1}^{(1)}, \mathbf{v}_{i_2}^{(2)} \right\rangle x_{i_1}^{(1)} x_{i_2}^{(2)} \\ &+ \cdots + \sum_{i_{m-1}=1}^{I_{m-1}} \sum_{i_m=1}^{I_m} \left\langle \mathbf{v}_{i_{m-1}}^{(m-1)}, \mathbf{v}_{i_m}^{(m)} \right\rangle x_{i_{m-1}}^{(m-1)} x_{i_m}^{(m)} \end{aligned} \tag{27}$$

The pairwise interaction parameter $w_{i_p,i_q}^{(p,q)} = \left\langle \mathbf{v}_{i_p}^{(p)}, \mathbf{v}_{i_q}^{(q)} \right\rangle$ in Eq. (27) indicates that $w_{i_p,i_q}^{(p,q)}$ can be learned from instances whose $x_{i_p}^{(p)} \neq 0$ and some $x_{i_v}^{(v)} \neq 0$ (sharing $\mathbf{v}_p$), or $x_{i_q}^{(q)} \neq 0$ and some $x_{i_v}^{(v)} \neq 0$ (sharing $\mathbf{v}_q$), which makes FMs more robust under sparsity than SVMs where only instances with $x_{i_p}^{(p)} \neq 0$ and $x_{i_q}^{(q)} \neq 0$ can be used to learn $w_{i_p,i_q}^{(p,q)}$.

The main difference between FMs and MVMs is that the interaction parameters in different orders are completely independent in FMs, *e.g.*, the first-order interaction $w_{i_v}^{(v)}$ and the second-order interaction $\mathbf{v}_{i_v}^{(v)}$ in Eq. (27). On the contrary, in MVMs, all the orders of interactions share the same set of latent factors, *e.g.*, $\mathbf{a}_{i_v}^{(v)}$ in Eq. (6). For example, the combination of $\mathbf{a}_{i_v}^{(v)}$ and the bias factors from other $m-1$ views, *i.e.*, $\mathbf{a}_{I_1+1}^{(1)}, \ldots, \mathbf{a}_{I_{v-1}+1}^{(v-1)}, \mathbf{a}_{I_{v+1}+1}^{(v+1)}, \ldots, \mathbf{a}_{I_m+1}^{(m)}$, approximates the first-order interaction $w_{i_v}^{(v)}$. Similarly, we can obtain the second-order interaction $w_{i_p,i_q}^{(p,q)}$ by combining $\mathbf{a}_{i_p}^{(p)}$, $\mathbf{a}_{i_q}^{(q)}$ and other $m-2$ bias factors.

Such difference is more significant for higher order FMs, as shown in Figure 3. Assuming the same number of factors in different orders of interactions, the number of parameters to be estimated in a mth-order FM is $1 + (1 + (m-1)k) \sum_{v=1}^{m} I_v = (k(m-1)+1)d + 1$ which can be much larger than $k(m+d)$ in MVMs, when there are many views (*i.e.*, a large m). Therefore, compared to MVMs, FMs are not fully factorized.

6. CONCLUSION

In this paper, we have proposed a multi-view machine (MVM) model and presented an efficient inference method based on stochastic gradient descent. In general, MVMs can be applied to a variety of supervised machine learning tasks, including classification and regression, and are particularly designed for data that is composed of features from multiple views, between which the interactions are effectively explored. In contrast to other models that explore only the partial interactions or factorize the interactions in different orders separately, MVMs jointly factorize the full-order interactions and thereby benefiting the parameter estimation under sparsity.

7. REFERENCES

[1] Yuanzhe Cai, Miao Zhang, Dijun Luo, Chris Ding, and Sharma Chakravarthy. Low-order tensor decompositions for social tagging recommendation. In *WSDM*, pages 695–704. ACM, 2011.

[2] Bokai Cao, Lifang He, Xiangnan Kong, Philip S. Yu, Zhifeng Hao, and Ann B. Ragin. Tensor-based multi-view feature selection with applications to brain diseases. In *ICDM*, pages 40–49. IEEE, 2014.

[3] Liangjie Hong, Aziz S Doumith, and Brian D Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *WSDM*, pages 557–566. ACM, 2013.

[4] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[5] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[6] Steffen Rendle. Factorization machines. In *ICDM*, pages 995–1000. IEEE, 2010.

[7] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, pages 81–90. ACM, 2010.

[8] Dacheng Tao, Xuelong Li, Weiming Hu, Stephen Maybank, and Xindong Wu. Supervised tensor learning. In *ICDM*, pages 8–pp. IEEE, 2005.

[9] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

[10] Ling Yan, Wu-jun Li, Gui-Rong Xue, and Dingyi Han. Coupled group lasso for web-scale CTR prediction in display advertising. In *ICML*, pages 802–810, 2014.