• Facebook recently announced 300 million users (Sept, 2009)

• Social Media has overtaken porn as the #1 activity on the Web

• % of companies using LinkedIn as a primary tool to find employees….80%

• The #2 largest search engine in the world is YouTube

• There are over 200,000,000 Blogs and 54% post content or tweet daily

• 25% of Americans in the past month said they watched a short video...on their phone

More than 1.5 million pieces of content (web links, news stories, blog posts, notes, photos, etc.) are shared on Facebook...daily.
Web Objects

Products
- amazon.com

Video
- YouTube
- twitter

Photo
- flickr
- facebook

Research Papers
- citeulike

$10^3 - 10^9$
Web Objects

Need to classify web objects into semantic categories

- Index and organize web objects efficiently
- Discover interesting patterns from web objects
- Browse and search of web objects conveniently
Web Objects Classification

“Harry Potter” DVD
Class: “Movies & TV”

amazon.com

The fifth book of “Harry Potter”
Class: “Books”

“Harry Potter” Halloween costume
Class: “Apparel & Accessories”
Challenging task for the specific characteristics of the data.

- LACK OF FEATURES
  Limited text description & difficulty on extracting content features of video/image
Web Objects Classification

Challenging task for the specific characteristics of the data.

- **LACK OF FEATURES**
  Limited text description & difficulty on extracting content features of video/image

- **LACK OF INTERCONNECTIONS**
  “Michael Jordan”
Web Objects Classification

Challenging task for the specific characteristics of the data.

• LACK OF FEATURES
  Limited text description & difficulty on extracting content features of video/image

• LACK OF INTERCONNECTIONS
  “Michael Jordan”

• LACK OF LABELS
  Difficulty to create a large training set
Web Objects Classification

Limited text description
Rich semantic feature space
Isolated settings of web objects
Labeled examples in some domains

Social tags
Overcome the difficulties of web object classification

Heterogeneous objects on Web are tagged by users, with keywords freely chosen from their own vocabulary
Users provides enriched semantic features for web object classification.
Social Tags

LACK OF INTERCONNECTIONS

New link structure of web objects
LACK OF LABELS

Heterogenous types of web objects are connected through common tags

Giulia Mialich – Luca Rossi
This is the first work to explore social tag data for web object classification.

Investigated for a long time: - web page classification
- multimedia classification

WEB PAGE CLASSIFICATION
- textual feature based
- hyperlink
- html & metadata
- query log

MULTIMEDIA OBJECT CLASSIFICATION
- text features
- contextual information
Related work

Authors propose a general theoretic framework for explicitly modeling tagging behaviors and web object classification problem.

Social Tag can benefit:

- web search
- information retrieval
- semantic web
- web page clustering
- user interest mining
Bao et al. Observe that the social annotation can benefit web search in two aspects:

1. Annotations are good summaries of corresponding web pages
   [Amazon’s homepage: shopping, books, amazon, music, store]
   Similar or closely related annotations are usually given to the same web pages
   SocialSimRank (SSR)

2. The count of annotations indicates the popularity of web pages from users’ point of view
   SocialPageRank (SPR)
Innovatively social tag exploration for web object classification.

They propose an iterative algorithm which solves the problem efficiently, significantly outperforming the state-of-the-art methods that don’t use tags as bridges.
Simplifying assumption: 2 types of objects, S and T. S objects are already labeled.
Social Tagging Graph

C = \{c_1, c_2, \ldots, c_k\} is the set of all the categories

V_T^l

G = (V,E) is the social tagging graph, where V is the set of all the objects (plus tags) and E is the set of edges between an object and its tags
C = \{c_1, c_2, \ldots, c_k\} is the set of all the categories

\[ V_u \]

G = (V,E) is the social tagging graph, where V is the set of all the objects (plus tags) and E is the set of edges between an object and its tags
C = \{c_1, c_2, \ldots, c_k\} is the set of all the categories

G = (V,E) is the social tagging graph, where V is the set of all the objects (plus tags) and E is the set of edges between an object and its tags
C = \{c_1, c_2, \ldots, c_k\} is the set of all the categories

G = (V, E) is the social tagging graph, where V is the set of all the objects (plus tags) and E is the set of edges between an object and its tags
Web users are likely to select similar tags for objects belonging to the same semantic category, independent of the type.

⇒ tags can be used as a “bridge” to semantically connect objects
The label assigned by the classifier should be consistent. This consistency can be captured by the following 3 properties:

- Category assignment of a vertex in $V_s$ or $V_{lt}$ should not deviate much from its original label as long as we trust the initial labeling.
The label assigned by the classifier should be consistent. This consistency can be captured by the following 3 properties:

Category assignment of a vertex in Vut should take into account any prior knowledge
The label assigned by the classifier should be consistent.
This consistency can be captured by the following 3 properties:

Category assignment of any vertex of the graph should be as consistent as possible with its neighbors’ labels.
The Optimization Framework

- $f_u$: a $k$-dimension vector that represents the class distribution of vertex $u \in V$, where $k$ is the number of categories. $f_u[i]$ represents the possibility that $u$ belongs to category $i$, s.t. $\sum_{i=1}^{k} f_u[i] = 1$. We denote $\{f_u\}_{u \in V}$ as $f$.

- $f_u^*$: the optimal solution of $f_u$

- $\hat{f}_u$: for $u \in V_S \cup V_T^l$, $\hat{f}_u$ is the class distribution estimated from the original category labels of vertex $u$. For $u \in V_T^u$, $\hat{f}_u$ is the class distribution estimated from some prior knowledge of the unlabeled object $u$ (e.g., the label assignments by a domain classifier).

- $w_{uv}$: a weight of the importance of edge $(u, v)$. Given an object $u$ and its associated tag $v$, $w_{uv}$ is the frequency that $v$ is used to tag $u$. 

Giulia Mialich – Luca Rossi
The Optimization Framework

\[
O(f) = \alpha \sum_{u \in V_S} \|f_u - \hat{f}_u\|^2 \\
+ \beta \sum_{u \in V^l_T} \|f_u - \hat{f}_u\|^2 \\
+ \gamma \sum_{u \in V^u_T} \|f_u - \hat{f}_u\|^2 + \\
+ \sum_{(u, v) \in E} w_{uv} \|f_u - f_v\|^2
\]
The Optimization Framework

1. \( \sum_{u \in V_S} \| f_u - \hat{f}_u \|^2 \) means that the category of a vertex in \( V_S \) should not deviate much from its original label(s).

2. \( \sum_{u \in V_T^l} \| f_u - \hat{f}_u \|^2 \) means that the category of a vertex in \( V_T^l \) should keep close to its initial label(s).

3. \( \sum_{u \in V_T^u} \| f_u - \hat{f}_u \|^2 \) means that the category of a vertex in \( V_T^u \) should keep close to the prior knowledge if any.

4. \( \sum_{(u,v) \in E} w_{uv} \| f_u - f_v \|^2 \) makes sure that the class distribution of the vertices are smooth over the whole graph, i.e., the class distribution of a vertex is consistent with its neighbors.
Our target is to find

\[ f^* = \arg \min \ O(f) \]

Based on this class distribution we can state that,
given an object \( o \), its class \( c \) is

\[ c = \arg \max \ \frac{P(o|c)}{P(o)} = \arg \max \ \frac{P(c|o)}{P(c)} \]
Our target is to find

\[ f^* = \arg \min O(f) \]

Based on this class distribution we can state that, given an object \( o \), its class \( c \) is

\[ c = \arg \max_{1 \leq i \leq k} \frac{f_u^*[i]}{\sum_{u' \in V^l_T \cup V^u_T} f_{u'}^*[i]} \]
Any problem so far?

Any problem so far?

Oh yes...

finding a closed-form solution to this problem requires \textbf{inverting a huge matrix} with the size of all the objects and tags

Giulia Mialich – Luca Rossi
Any problem so far?

Why not using a smart iterative algorithm instead? I’ve got an idea: let’s differentiate $O(f)$ with respect to the 4 types of vertices and update $f$ by setting the differentiated result to zero!
Toward an Iterative Algorithm

\[
\frac{\partial O}{\partial s} = 2\alpha(f_s - \hat{f}_s) + 2 \sum_{v \in V_{tag}} w_{sv}(f_s - f_v) = 0
\]

\[
f_s = \frac{\alpha}{\alpha + \sum_{v \in V_{tag}} w_{sv}} \hat{f}_s + \frac{\sum_{v \in V_{tag}} w_{sv}f_v}{\alpha + \sum_{v \in V_{tag}} w_{sv}} \quad (3)
\]

\[
\frac{\partial O}{\partial l} = 2\beta(f_l - \hat{f}_l) + 2 \sum_{v \in V_{tag}} w_{lv}(f_l - f_v) = 0
\]

\[
f_l = \frac{\beta}{\beta + \sum_{v \in V_{tag}} w_{lv}} \hat{f}_l + \frac{\sum_{v \in V_{tag}} w_{lv}f_v}{\beta + \sum_{v \in V_{tag}} w_{lv}} \quad (4)
\]

\[
\frac{\partial O}{\partial u} = 2\gamma(f_u - \hat{f}_u) + 2 \sum_{v \in V_{tag}} w_{uv}(f_u - f_v) = 0
\]

\[
f_u = \frac{\gamma}{\gamma + \sum_{v \in V_{tag}} w_{uv}} \hat{f}_u + \frac{\sum_{v \in V_{tag}} w_{uv}f_v}{\gamma + \sum_{v \in V_{tag}} w_{uv}} \quad (5)
\]
3.3 The Optimization Framework

With the intuitions discussed in Section 3.2, we now define the objective function.

\[ \frac{\partial O}{\partial v} = -2 \sum_{s \in V_S} w_{sv}(f_s - f_v) - 2 \sum_{l \in V_T^l} w_{lv}(f_l - f_v) - 2 \sum_{u \in V_T^u} w_{uv}(f_u - f_v) = 0 \]

\[ f_v = \frac{\sum_{s \in V_S} w_{sv} f_s + \sum_{l \in V_T^l} w_{lv} f_l + \sum_{u \in V_T^u} w_{uv} f_u}{\sum_{s \in V_S} w_{sv} + \sum_{l \in V_T^l} w_{lv} + \sum_{u \in V_T^u} w_{uv}} \] (6)
Algorithm 1: Iterative Algorithm

**Input:** category size $k$, class labels $C(x)$ for $x \in V_S \cup V_T^l \cup V_T^u$

**Output:** class labels $\tilde{C}(x)$ for $x \in V_T^u$

// Initialization
1 foreach $x \in V_S \cup V_T^l \cup V_T^u$ do
2 $\hat{f}_x[C(x)] \leftarrow 1$
3 foreach $x \in V$ do
4 foreach $i \leftarrow 1$ to $k$ do $f_x[i] \leftarrow 1/k$

// Iteration
5 repeat
6 foreach $x \in V_S$ do
7 $f_x' \leftarrow \frac{\alpha}{\alpha + \sum_{v \in V_{tag}} w_{xv} f_v} \hat{f}_x + \frac{\sum_{v \in V_{tag}} w_{xv} f_v}{\alpha + \sum_{v \in V_{tag}} w_{xv}}$
8 foreach $x \in V_T^l$ do
9 $f_x' \leftarrow \frac{\beta}{\beta + \sum_{v \in V_{tag}} w_{xv} f_v} \hat{f}_x + \frac{\sum_{v \in V_{tag}} w_{xv} f_v}{\beta + \sum_{v \in V_{tag}} w_{xv}}$
10 foreach $x \in V_T^u$ do
11 $f_x' \leftarrow \frac{\gamma}{\gamma + \sum_{v \in V_{tag}} w_{xv} f_v} \hat{f}_x + \frac{\sum_{v \in V_{tag}} w_{xv} f_v}{\gamma + \sum_{v \in V_{tag}} w_{xv}}$
12 foreach $x \in V$ do
13 $f_x \leftarrow f_x'$
14 until converged ;

// Get Class Label
15 foreach $x \in V_T^u$ do
16 $\tilde{C}(x) = \arg \max_{1 \leq i \leq k} \frac{f_x[i]}{\sum_{u \in V_T^l \cup V_T^u} f_u[i]}$
17

\[ \sum_{s \in V_S} w_{sx} f_s + \sum_{l \in V_T^l} w_{lx} f_l + \sum_{u \in V_T^u} w_{ux} f_u \]
The vector encoding the prior knowledge and the class distribution of each vertex $u \in V$ are initialized.
The class distribution of each vertex $u$ is repeatedly updated according to its neighbors vertices.
If $u$ is not a tag, the class distribution of object $u$ is updated from the class distribution of the associated tags according to (3), (4) and (5).
if u is a tag, its class distribution is updated according to the class distributions of the connected objects (based on equation (6))

Note how the tags act as a bridge of belief propagation
Complexity Analysis

Lines 1-4 take $O(k |V|)$
Lines 6-11 take $O(2k |E|)$

$\Rightarrow O(k |V| + \text{iter} |E|)$

where $k$ is the number of categories and $\text{iter}$ is the number of iterations
Tuning the parameters

\[ \alpha=0, \beta\neq 0 \text{ and } \gamma=0 \]
Tuning the parameters

\[ \alpha \neq 0, \beta = 0 \text{ and } \gamma = 0 \]
Tuning the parameters

\[ \alpha \neq 0, \beta \neq 0 \text{ and } \gamma \neq 0 \]
Web products classification (from Amazon). Web pages collected from ODP are used as external resource for helping classification.

Tags for web pages are collected from delicious.

<table>
<thead>
<tr>
<th>ODP:Shopping</th>
<th>Amazon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Count</td>
</tr>
<tr>
<td>Publications/Books</td>
<td>558</td>
</tr>
<tr>
<td>Consumer_Electronics</td>
<td>494</td>
</tr>
<tr>
<td>Health</td>
<td>1009</td>
</tr>
<tr>
<td>Home_and_Garden</td>
<td>1976</td>
</tr>
<tr>
<td>Jewelry</td>
<td>452</td>
</tr>
<tr>
<td>Music</td>
<td>527</td>
</tr>
<tr>
<td>Office</td>
<td>77</td>
</tr>
<tr>
<td>Pet</td>
<td>443</td>
</tr>
</tbody>
</table>
Experiments setup: the measure

\[
\pi_i = \frac{TP_i}{TP_i + FP_i}, \quad \rho_i = \frac{TP_i}{TP_i + FN_i}
\]

\[
F_i = \frac{2\pi_i \rho_i}{\pi_i + \rho_i}, \quad F(\text{macro-averaged}) = \frac{\sum_{i=1}^{M} F_i}{M}
\]

Macro-averaged scores (MacroF1) are influenced by the performance in rare categories.
Experiments setup: the measure

\[ \pi = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FP_i)} , \quad \rho = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FN_i)} . \]

\[ F\text{(micro-averaged)} = \frac{2\pi \rho}{\pi + \rho} . \]

Micro-averaged scores (MicroF1) tend to be dominated by the performance on common categories.
The Tag-based classification Model (TM) presented in the paper is compared with SVM (Support Vector Machine) and HG (Harmonic Gaussian field method). Both are used with the title or the tag of the products as features.
Experiments: overall performance

We show that our algorithm is effective when there is few or no labeled data. It is because there is no prior knowledge of the degree of trust of information from another domain (web page). Therefore, from the experiment result, we can safely draw the conclusion that tags are meaningful features for object classification. To consider such a case, we propose a classification model based on social tagging, which explicitly represents the interconnections of objects through the social tag structure. The link structure of objects and tags, which is used as training data, from 0 to 5, which means there is no or fully trust the existing labels. For example, 5% means 5 percent of labeled data is used as training data, and 95% is used as testing data.

5.2.3 Exploring the interconnections of objects

The link structure of objects and tags is used as training data, from 0 to 5, which means there is no or fully trust the existing labels. For example, 5% means 5 percent of labeled data is used as training data, and 95% is used as testing data. In this way, our classification model can achieve an impressive classification result (i.e., web page). The results of different classification methods are shown in Table 2. From the table, for each classification model, using tags as the features space achieves much better result than using titles as the edge weight in the graph.

5.2.4 Handling lack of labeling issue

There are many cases, such as the expansion pace of the Web, where labeled data are not available for web objects. Besides, prior knowledge of web objects or certain categories can be inferred from the link structure of objects and tags, which e

Table 2. From the table, for each classification model, using tags as the feature space performs better than using titles as the feature space. In the experiment, we set \( \alpha = 1000, \beta = \infty, \gamma = 0.1 \).

<table>
<thead>
<tr>
<th>Label Ratio</th>
<th>Measure</th>
<th>1%</th>
<th>5%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MicroF1</td>
<td>MacroF1</td>
<td>MicroF1</td>
</tr>
<tr>
<td>SVM+TITLE</td>
<td>0.4233</td>
<td>0.3812</td>
<td>0.5967</td>
</tr>
<tr>
<td>SVM+TAG</td>
<td>0.4045</td>
<td>0.4059</td>
<td>0.6397</td>
</tr>
<tr>
<td>HG+TITLE</td>
<td>0.6251</td>
<td>0.6038</td>
<td>0.6778</td>
</tr>
<tr>
<td>HG+TAG</td>
<td>0.7174</td>
<td>0.7127</td>
<td>0.7856</td>
</tr>
<tr>
<td>TM\textsuperscript{5}</td>
<td>0.7870</td>
<td>0.7872</td>
<td>0.8027</td>
</tr>
</tbody>
</table>

\( \textsuperscript{5} \alpha = 1000, \beta = \infty, \gamma = 0.1 \)
Experiments: Tag vs Title

<table>
<thead>
<tr>
<th>p%</th>
<th>SVM+TITLE</th>
<th>SVM+TAG</th>
<th>HG+TITLE</th>
<th>HG+TAG</th>
<th>TM(^\text{6})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MicroF1</td>
<td>MacroF1</td>
<td>MicroF1</td>
<td>MacroF1</td>
<td>MicroF1</td>
</tr>
<tr>
<td>5%</td>
<td>0.5967</td>
<td>0.6091</td>
<td>0.6397</td>
<td>0.6435</td>
<td>0.6778</td>
</tr>
<tr>
<td>10%</td>
<td>0.6700</td>
<td>0.6789</td>
<td>0.7168</td>
<td>0.7334</td>
<td>0.6937</td>
</tr>
<tr>
<td>15%</td>
<td>0.7181</td>
<td>0.7218</td>
<td>0.7417</td>
<td>0.7366</td>
<td>0.7139</td>
</tr>
<tr>
<td>20%</td>
<td>0.7343</td>
<td>0.7399</td>
<td>0.7674</td>
<td>0.7722</td>
<td>0.7152</td>
</tr>
<tr>
<td>25%</td>
<td>0.7545</td>
<td>0.7597</td>
<td>0.7763</td>
<td>0.7780</td>
<td>0.7131</td>
</tr>
</tbody>
</table>

\[^6\alpha = 0, \beta = \infty, \gamma = 0\]
Experiments: 2 domains

If the two domains are similar, we could trust even more on the homogeneous labeled objects and unlabeled objects. If the type and use our model to explore the link structure among enough labels, we should rely on the labeling of the same type. The results suggest that if there are no or a bridge to transform knowledge, then even less than 5% in this specific case, we could trust more on the knowledge from heterogeneous objects. If the test category information from heterogeneous objects can also help classification through tags when there is not sufficient, then even less than 5% in this specific case, we could trust even more on the knowledge from web page to product via social tags. MicroF1 = 0.7594 and MacroF1 = 0.7606) by transferring both homogeneous and heterogeneous objects empirically.

Table 4: Comparison of classification results using:

<table>
<thead>
<tr>
<th>p%</th>
<th>HG+TITLE MicroF1</th>
<th>HG+TAG MicroF1</th>
<th>α = 1000 MicroF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NA</td>
<td>NA</td>
<td>0.7594</td>
</tr>
<tr>
<td>1%</td>
<td>0.6251</td>
<td>0.7174</td>
<td>0.7708</td>
</tr>
<tr>
<td>2%</td>
<td>0.6499</td>
<td>0.7510</td>
<td>0.7771</td>
</tr>
<tr>
<td>3%</td>
<td>0.6368</td>
<td>0.7695</td>
<td>0.7774</td>
</tr>
<tr>
<td>4%</td>
<td>0.6503</td>
<td>0.7566</td>
<td>0.7885</td>
</tr>
<tr>
<td>5%</td>
<td>0.6778</td>
<td>0.7856</td>
<td>0.7872</td>
</tr>
</tbody>
</table>

Figure 3: Sensitivity of parameter α's that we choose here. It illustrates the idea that as the parameter increases to 5%, we find that the classification performance decreases. We can set a threshold for α, where α = 0.001, 0.01, 0.1, 1, and ∞ = 1000. From Figure 4, we find that our algorithm converges quickly at about 10 rounds of iterations.

Table 3: Comparison of title feature and tag feature

<table>
<thead>
<tr>
<th>p%</th>
<th>HG+TITLE MicroF1</th>
<th>HG+TAG MicroF1</th>
<th>α = 1000 MicroF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NA</td>
<td>NA</td>
<td>0.7594</td>
</tr>
<tr>
<td>1%</td>
<td>0.6251</td>
<td>0.7174</td>
<td>0.7708</td>
</tr>
<tr>
<td>2%</td>
<td>0.6499</td>
<td>0.7510</td>
<td>0.7771</td>
</tr>
<tr>
<td>3%</td>
<td>0.6368</td>
<td>0.7695</td>
<td>0.7774</td>
</tr>
<tr>
<td>4%</td>
<td>0.6503</td>
<td>0.7566</td>
<td>0.7885</td>
</tr>
<tr>
<td>5%</td>
<td>0.6778</td>
<td>0.7856</td>
<td>0.7872</td>
</tr>
</tbody>
</table>

α = 0.01+(SVM+TAG) performs better than SVM+TAG. The result with HG+TAG as prior is better than the one with SVM+TAG as prior as expected, since HG+TAG is better than SVM+TAG. Compared with SVM+TAG and HG+TAG, our model using prior knowledge outperforms the one without prior (i.e., HG+TAG as prior) in most of the cases. The larger γ, the more we trust the prior knowledge. Therefore, our model performs better with γ = 0 and β = 0 and γ = 0.1. From Figure 4, we find that our algorithm converges quickly at about 10 rounds of iterations.

Table 5 shows that the performance of our model using prior knowledge is considered as the prior for our model.
Experiments: sensitivity of alpha

![Graph showing sensitivity of Micro F1 and Macro F1 to parameter alpha]

- Micro F1
- Macro F1

Figure 3: Sensitivity of paramter α

- Micro F1
- Macro F1

Table 3: Comparison of title feature and tag feature

<table>
<thead>
<tr>
<th>α</th>
<th>Micro F1</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7885</td>
<td>0.7774</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7891</td>
<td>0.7769</td>
</tr>
<tr>
<td>0.01</td>
<td>0.7856</td>
<td>0.7766</td>
</tr>
<tr>
<td>0.001</td>
<td>0.7606</td>
<td>0.7719</td>
</tr>
<tr>
<td>0</td>
<td>0.7872</td>
<td>0.7719</td>
</tr>
<tr>
<td>∞</td>
<td>0.7606</td>
<td>0.7719</td>
</tr>
</tbody>
</table>

Table 4: Comparison of classification results using both homogeneous and heterogeneous objects

- SVM+TAG
- HG+TITLE
- HG+TAG

<table>
<thead>
<tr>
<th>p%</th>
<th>Micro F1</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0.6778</td>
<td>0.753</td>
</tr>
<tr>
<td>3%</td>
<td>0.6360</td>
<td>0.755</td>
</tr>
<tr>
<td>2%</td>
<td>0.6368</td>
<td>0.756</td>
</tr>
<tr>
<td>1%</td>
<td>0.6334</td>
<td>0.757</td>
</tr>
<tr>
<td>0.1%</td>
<td>0.758</td>
<td>0.758</td>
</tr>
</tbody>
</table>

Table 5 shows that the performance of our model using prior knowledge outperforms the one without prior (i.e., SVM+TAG as prior as expected, since HG+TAG is better than SVM+TAG).

Heterogeneous knowledge performs better than the one without prior (i.e., SVM+TAG and HG+TAG to classify the remaining products). The classification results are considered as the prior for our model.
Experiments: prior knowledge

<table>
<thead>
<tr>
<th>Measure</th>
<th>5% MicroF1</th>
<th>5% MacroF1</th>
<th>10% MicroF1</th>
<th>10% MacroF1</th>
<th>15% MicroF1</th>
<th>15% MacroF1</th>
<th>20% MicroF1</th>
<th>20% MacroF1</th>
<th>25% MicroF1</th>
<th>25% MacroF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma=0$</td>
<td>0.7918</td>
<td>0.7919</td>
<td>0.8005</td>
<td>0.7996</td>
<td>0.8187</td>
<td>0.8199</td>
<td>0.8217</td>
<td>0.8231</td>
<td>0.8259</td>
<td>0.8273</td>
</tr>
<tr>
<td>SVM+TAG</td>
<td>0.6397</td>
<td>0.6435</td>
<td>0.7168</td>
<td>0.7334</td>
<td>0.7417</td>
<td>0.7366</td>
<td>0.7674</td>
<td>0.7722</td>
<td>0.7763</td>
<td>0.7780</td>
</tr>
<tr>
<td>$(\gamma=0.001)$+(SVM+TAG)</td>
<td>0.7938</td>
<td>0.7914</td>
<td>0.8000</td>
<td>0.7987</td>
<td>0.8214</td>
<td>0.8198</td>
<td>0.8229</td>
<td>0.8238</td>
<td>0.8281</td>
<td>0.8295</td>
</tr>
<tr>
<td>$(\gamma=0.1)$+(SVM+TAG)</td>
<td>0.7964</td>
<td>0.7932</td>
<td>0.8013</td>
<td>0.8005</td>
<td>0.8199</td>
<td>0.8184</td>
<td>0.8223</td>
<td>0.8231</td>
<td>0.8292</td>
<td>0.8306</td>
</tr>
<tr>
<td>$(\gamma=1)$+(SVM+TAG)</td>
<td>0.7796</td>
<td>0.7673</td>
<td>0.8096</td>
<td>0.8109</td>
<td>0.8251</td>
<td>0.8201</td>
<td>0.8272</td>
<td>0.8277</td>
<td>0.8355</td>
<td>0.8364</td>
</tr>
<tr>
<td>$\gamma=0$</td>
<td>0.6878</td>
<td>0.6846</td>
<td>0.7704</td>
<td>0.7803</td>
<td>0.7913</td>
<td>0.7843</td>
<td>0.8033</td>
<td>0.8051</td>
<td>0.8165</td>
<td>0.8163</td>
</tr>
<tr>
<td>HG+TAG</td>
<td>0.7856</td>
<td>0.7859</td>
<td>0.7915</td>
<td>0.7864</td>
<td>0.7921</td>
<td>0.7908</td>
<td>0.8025</td>
<td>0.8004</td>
<td>0.8109</td>
<td>0.8079</td>
</tr>
<tr>
<td>$(\gamma=0.001)$+(HG+TAG)</td>
<td>0.7968</td>
<td>0.7973</td>
<td>0.8038</td>
<td>0.8026</td>
<td>0.8214</td>
<td>0.8228</td>
<td>0.8251</td>
<td>0.8263</td>
<td>0.8300</td>
<td>0.8316</td>
</tr>
<tr>
<td>$(\gamma=0.1)$+(HG+TAG)</td>
<td>0.8012</td>
<td>0.8028</td>
<td>0.8056</td>
<td>0.8040</td>
<td>0.8222</td>
<td>0.8233</td>
<td>0.8249</td>
<td>0.8286</td>
<td>0.8313</td>
<td>0.8329</td>
</tr>
<tr>
<td>$(\gamma=1)$+(HG+TAG)</td>
<td>0.8038</td>
<td>0.8043</td>
<td>0.8174</td>
<td>0.8151</td>
<td>0.8233</td>
<td>0.8238</td>
<td>0.8296</td>
<td>0.8301</td>
<td>0.8381</td>
<td>0.8387</td>
</tr>
<tr>
<td>$\gamma=0$</td>
<td>0.7950</td>
<td>0.7951</td>
<td>0.8036</td>
<td>0.7982</td>
<td>0.8082</td>
<td>0.8065</td>
<td>0.8206</td>
<td>0.8192</td>
<td>0.8339</td>
<td>0.8308</td>
</tr>
</tbody>
</table>
Experiments: convergence

![Accuracy Change vs Iteration Graph]

Figure 4: Accuracy change in 10 iterations

- **Scenario 1**: α = 1000, γ = 0
- **Scenario 2**: α = 1000, γ = 0.1
- **Scenario 3**: α = 1000, γ = 0.01

Accuracy Change

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10^-2</td>
<td>10^-2</td>
<td>10^-2</td>
</tr>
<tr>
<td>2</td>
<td>10^-1</td>
<td>10^-1</td>
<td>10^-1</td>
</tr>
<tr>
<td>3</td>
<td>10^-2</td>
<td>10^-2</td>
<td>10^-2</td>
</tr>
<tr>
<td>4</td>
<td>10^-3</td>
<td>10^-3</td>
<td>10^-3</td>
</tr>
<tr>
<td>5</td>
<td>10^-4</td>
<td>10^-4</td>
<td>10^-4</td>
</tr>
<tr>
<td>6</td>
<td>10^-5</td>
<td>10^-5</td>
<td>10^-5</td>
</tr>
<tr>
<td>7</td>
<td>10^-6</td>
<td>10^-6</td>
<td>10^-6</td>
</tr>
<tr>
<td>8</td>
<td>10^-7</td>
<td>10^-7</td>
<td>10^-7</td>
</tr>
<tr>
<td>9</td>
<td>10^-8</td>
<td>10^-8</td>
<td>10^-8</td>
</tr>
<tr>
<td>10</td>
<td>10^-9</td>
<td>10^-9</td>
<td>10^-9</td>
</tr>
<tr>
<td>11</td>
<td>10^-10</td>
<td>10^-10</td>
<td>10^-10</td>
</tr>
</tbody>
</table>

Giulia Mialich – Luca Rossi
Conclusions

- Web object classification: An emerging task and increasingly important
- Web object classification problem can take advantage from social tags in three aspects
  - represent web objects in a meaningful feature space
  - interconnect objects to indicate implicit relationship
  - bridging heterogeneous objects so that category information can be propagated from one domain to another
- The proposed method significantly outperforms the state-of-the-art of general classification methods
- In this model, it is only considered the setting of two types of web objects
  - It would be interesting to generalize the model to manage multi-types of objects