Semi-Supervised Learning (SSL)
Conclusions: Beta version 0.1

- We propose a **semi-supervised boosting** algorithm
Conclusions: Beta version 0.1

- We propose a semi-supervised boosting algorithm
- which solves multi-class problems without decomposing them into binary tasks.
We propose a semi-supervised boosting algorithm which solves multi-class problems without decomposing them into binary tasks.

Additionally, our algorithm scales very well with respect to the number of both labeled and unlabeled samples.
Outline
Semi-supervised learning is a class of machine learning techniques that make use of both labeled and unlabeled data for training.

- There exists many SSL methods, see:
  - X. Zhu, “Semi-Supervised Learning Survey”, 2008 and
Motivations

Many successful SSL methods do not scale very well w.r.t. the number of unlabeled samples, or are very sensitive to the choice of hyper-parameters (G. Mann, A. McCallum, ICML 2007). Expect to see $O(n^3)$ many times.
Motivations

- Many successful SSL methods do not scale very well w.r.t. the number of unlabeled samples, or are very sensitive to the choice of hyper-parameters (G. Mann, A. McCallum, ICML 2007). Expect to see $O(n^3)$ many times.
- Usually multi-class problems are solved via 1-vs-all and occasionally with 1-vs-1 decompositions.
What is wrong with 1-vs-all?

- Do you want to repeat a slow method a few more of times?

Calibration problems (B. Schoelkopf, A. Smola, 2002).

Artificial unbalanced binary problems.

There exists slow multi-class SSL methods, see the details in the paper.
What is wrong with 1-vs-all?

- Do you want to repeat a slow method a few more of times?
- **Calibration** problems (B. Schoelkopf, A. Smola, 2002).

Artificial unbalanced binary problems. There exists slow multi-class SSL methods, see the details in the paper.
What is wrong with 1-vs-all?

- Do you want to repeat a slow method a few more times?
- **Calibration** problems (B. Schoelkopf, A. Smola, 2002).
- Artificial **unbalanced** binary problems.
What is wrong with 1-vs-all?

- Do you want to repeat a slow method a few more times?
- **Calibration** problems (B. Schoelkopf, A. Smola, 2002).
- Artificial **unbalanced** binary problems.

- There exists slow multi-class SSL methods, see the details in the paper.
Multi-Class Semi-Supervised Boosting

Multi-class classifier: \( f(x) = [f_1(x), \ldots, f_K(x)]^T \).
Multi-Class Semi-Supervised Boosting

Multi-class classifier: \( f(x) = [f_1(x), \ldots, f_K(x)]^T \). 

**Overall Loss**

\[
\mathcal{L}(f(x), X) = \sum_{(x,y) \in X_l} \ell(f(x)) + \alpha \sum_{x \in X_u} \ell_c(f(x)) + \beta \sum_{x \in X_u} \ell_m(f(x)) \tag{1}
\]

- Labeled
- Unlabeled
Multi-Class Semi-Supervised Boosting

Multi-class classifier: $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_K(\mathbf{x})]^T$.

**Overall Loss**

$$
\mathcal{L}(\mathbf{f}(\mathbf{x}), \mathcal{X}) = \sum_{(\mathbf{x}, y) \in \mathcal{X}_l} \ell(\mathbf{f}(\mathbf{x})) + \alpha \sum_{\mathbf{x} \in \mathcal{X}_u} \ell_c(\mathbf{f}(\mathbf{x})) + \beta \sum_{\mathbf{x} \in \mathcal{X}_u} \ell_m(\mathbf{f}(\mathbf{x}))
$$  \hspace{1cm} (1)

- **Labeled**
- **Unlabeled**

**Boosting Model**

$$
\mathbf{f}(\mathbf{x}) = \nu \sum_{t=1}^{T} \mathbf{g}^t(\mathbf{x})
$$  \hspace{1cm} (2)
Fisher-Consistent Loss Functions

\[ R(T_\alpha) \leq R_{emp}(T_\alpha) + \frac{\ln W - \ln n}{s} \left( 1 + \sqrt{1 + \frac{2R_{emp}(T_\alpha)}{\ln W - \ln n}} \right) \]

ALL YOUR BAYES ARE BELONG TO US

Vladimir Vapnik (picture courtesy of Yann LeCun)
Fisher-Consistent Loss Functions

Margin Vector

\( f(x) \) is a **universal margin vector**, if \( \forall x : \sum_{i=1}^{K} f_i(x) = 0. \)
Fisher-Consistent Loss Functions

**Margin Vector**

\( f(x) \) is a universal margin vector, if \( \forall x : \sum_{i=1}^{K} f_i(x) = 0. \)

**Fisher-Consistent Loss**

\( \ell(\cdot) \) is Fisher-consistent, if the minimization of the expected risk:

\[
\hat{f}(x) = \arg \min_{f(x)} \int (x, y) \ell(f_y(x)) p(y, x) d(x, y) \tag{3}
\]

has a unique solution and

\[
C(x) = \arg \max_i \hat{f}_i(x) = \arg \max_i p(y = i | x). \tag{4}
\]
Fisher-Consistent Loss Functions

**Margin Vector**

\( f(x) \) is a **universal margin vector**, if \( \forall x : \sum_{i=1}^{K} f_i(x) = 0 \).

**Fisher-Consistent Loss**

\( \ell(\cdot) \) is **Fisher-consistent**, if the minimization of the expected risk:

\[
\hat{f}(x) = \arg \min_{f(x)} \int_{(x,y)} \ell(f_y(x))p(y,x)d(x,y)
\]

has a unique solution and

\[
C(x) = \arg \max_i \hat{f}_i(x) = \arg \max_i p(y = i|x).
\]
Margin Assumption
Margin Assumption

Put the decision boundary over low-density regions of features space. This is equivalent to maximizing the margin of the unlabeled samples.

Example

Transductive Support Vector Machines (TSVM, T. Joachims, ICML 1999) uses this loss function for the binary SVM classifier $h(x)$

$$
\ell_u(h(x)) = \max(0, 1 - |h(x)|)
$$

(5)

Multi-Class Unlabeled Margin

We propose to maximize the multi-class margin of the unlabeled samples by using

$$
\ell_m(f(x)) = \max(0, M - \max_i (f_i(x))).
$$

(6)
Manifold Assumption
Manifold Assumption

Enforce the classifier to predict similar labels for similar unlabeled samples.

Example

Graph-based methods, such as Laplacian SVM (Belkin et al., JMLR 2006), use this loss function for the binary SVM classifier $h(x)$

$$
\ell_u(h(x)) = \sum_{x' \in \mathcal{X}_u, x' \neq x} s(x, x') \| h(x) - h(x') \|^2.
$$

Cluster Prior

We enforce the multi-class classifier to have a consistent probabilistic estimates over regions of feature space formed by similar samples, i.e. clusters.
Cluster Priors

\[ p(y|x) = \text{[red, blue, green]} \]

\[
\begin{align*}
1, 0, 0 \\
0.33, 0, 0.66 \\
0, 0, 1
\end{align*}
\]
Cluster Priors

\[ p(y|x) = [\text{red}, \text{blue}, \text{green}] \]

\[ [1, 0, 0] \quad \text{red} \]

\[ [0, 1, 0] \quad \text{blue} \]

\[ [0.6, 0, 0.4] \quad \text{green} \]

\[ [0, 0, 1] \quad \text{grey} \]
Cluster Priors
Cluster Priors

∀ \mathbf{x} \in \mathcal{X}_u, \forall i \in \{1, \cdots, K\} : p_p(y = i|\mathbf{x}) .

We use the Kullback-Leibler (KL) divergence

$$\mathcal{D}_c(\mathbf{f}(\mathbf{x})) = -\mathbf{p}_p^T \mathbf{f}(\mathbf{x}) + \log \sum_{j=1}^{K} e^{f_j(\mathbf{x})} .$$

(8)
Cluster Priors

Let $\forall x \in \mathcal{X}_u, \forall i \in \{1, \cdots , K\} : p_p(y = i|x)$.

We use the Kullback-Leibler (KL) divergence

$$\ell_c(f(x)) = -p_p^T f(x) + \log \sum_{j=1}^{K} e^{f_j(x)}.$$  (8)

- Use any clustering method which suits your application.
- Use similarity functions if it helps clustering to recover the manifolds.
Cluster Priors

∀x ∈ X_u, ∀i ∈ {1, ⋯, K} : p_p(y = i|x) .

We use the Kullback-Leibler (KL) divergence

\[ \ell_c(f(x)) = -p_p^T f(x) + \log \sum_{j=1}^{K} e^{f_j(x)}. \] (8)

- Use any clustering method which suits your application.
- Use similarity functions if it helps clustering to recover the manifolds.
- Use any other source of information in form of priors: label prior, knowledge transfer, human prior knowledge.
Learning with Functional Gradient Descent

\[ F^*(x) = F_0(x) - \nu \sum_{t=1}^{T} \frac{\partial L}{\partial F}|(F_{t-1}(x)) \]

RMSBoost

Learning task for \( t^{th} \) boosting stage becomes

\[
 g^t(x) = \arg \max_{g(x)} \sum_{(x,y) \in \mathcal{X}_l} e^{-f_y(x)} y^T g(x) + \sum_{x \in \mathcal{X}_u} (\alpha \Delta p + \beta m)^T g(x). \quad (9)
\]

Theorem

The solution using a multi-class classifier \( C(x) \in \{1, \ldots, K\} \) is

\[
 C_t(x) = \arg \min_{C(x)} \sum_{(x,y) \in \mathcal{X}_l} I(C(x) \neq y) + \sum_{x \in \mathcal{X}_u} w_u I(C(x) \neq z) \quad (10)
\]

where \( w_l = e^{-f_y(x)} \) is the weight for a labeled sample, \( z = \arg \max_i (\alpha \Delta p_i + \beta m_i) \) and \( w_u = \alpha \Delta p + \beta m \) are the pseudo-label and weight for an unlabeled sample, respectively.
RMSBoost

Learning task for $t^{th}$ boosting stage becomes

$$g^t(x) = \arg\max_{g(x)} \sum_{(x,y) \in \mathcal{X}_l} e^{-f_y(x)} y^T g(x) + \sum_{x \in \mathcal{X}_u} (\alpha \Delta p + \beta m)^T g(x). \quad (9)$$

**Theorem**

The solution using a multi-class classifier $C(x) \in \{1, \cdots, K\}$ is

$$C_t(x) = \arg\min_{C(x)} \sum_{(x,y) \in \mathcal{X}_l} w_l \mathbb{I}(C(x) \neq y) + \sum_{x \in \mathcal{X}_u} w_u \mathbb{I}(C(x) \neq z) \quad (10)$$

where $w_l = e^{-f_y(x)}$ is the weight for a labeled sample, $z = \arg\max_i (\alpha \Delta p_i + \beta m_i)$ and $w_u = \alpha \Delta p_z + \beta m_z$ are the pseudo-label and weight for an unlabeled sample, respectively.
Experimental Settings

- **RMSBoost** is compared with:
  - Kernel SVM
  - Multi-Switch TSVM (Sindhwani and Keerthi, SIGIR 2006)
  - SERBoost (Saffari et al., ECCV 2008)
  - RMBoost
Experimental Settings

- **RMSBoost** is compared with:
  - Kernel SVM
  - Multi-Switch TSVM (Sindhwani and Keerthi, SIGIR 2006)
  - SERBoost (Saffari et al., ECCV 2008)
  - RMBoost

- Base learners are tiny **extremely randomized forests**, each consisting of 10 trees.
Experimental Settings

- **RMSBoost** is compared with:
  - Kernel SVM
  - Multi-Switch TSVM (Sindhwani and Keerthi, SIGIR 2006)
  - SERBoost (Saffari et al., ECCV 2008)
  - RMBoost

- Base learners are tiny extremely randomized forests, each consisting of 10 trees.

- Boosting iterations set to be 10000.
Experimental Settings

- **RMSBoost** is compared with:
  - Kernel SVM
  - Multi-Switch TSVM (Sindhwani and Keerthi, SIGIR 2006)
  - SERBoost (Saffari et al., ECCV 2008)
  - RMBoost

- Base learners are tiny extremely randomized forests, each consisting of 10 trees.

- Boosting iterations set to be 10000.

- Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.
Experimental Settings

- **RMSBoost** is compared with:
  - Kernel SVM
  - Multi-Switch TSVM (Sindhwani and Keerthi, SIGIR 2006)
  - SERBoost (Saffari et al., ECCV 2008)
  - RMBoost

- Base learners are tiny extremely randomized forests, each consisting of 10 trees.

- Boosting iterations set to be **10000**.

- Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.

- All boosting and RF methods are implemented in C++ and use **ATLAS** subroutines.
5% of the training data is chosen randomly to form the labeled set, the rest 95% is used as unlabeled set.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Train</th>
<th># Test</th>
<th># Class</th>
<th># Feat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter</td>
<td>15000</td>
<td>5000</td>
<td>26</td>
<td>16</td>
</tr>
<tr>
<td>SensIt (com)</td>
<td>78823</td>
<td>19705</td>
<td>3</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table:** Data sets for the machine learning experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>AML</th>
<th>SVM</th>
<th>TSVM</th>
<th>SER</th>
<th>RMB</th>
<th>RMSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter</td>
<td>72.3</td>
<td>70.3</td>
<td>65.9</td>
<td>76.5</td>
<td>74.4</td>
<td>79.9</td>
</tr>
<tr>
<td>SensIt</td>
<td>79.5</td>
<td>80.2</td>
<td>79.9</td>
<td>81.9</td>
<td>79.0</td>
<td>83.7</td>
</tr>
</tbody>
</table>

**Table:** Classification accuracy (in %).
Standard bag-of-words using quantized SIFT on a regular grid at multiple scales.

Images are represented by $L_1$-normalized 2-level spatial pyramids.

For SVM, pyramid $\chi^2$ kernel is used.
PASCAL 2006 Object Categorization Dataset

![Graph showing class accuracy against labeled sample ratio for various methods: RMSB, SER, AML, SVM, and TSVM. The graph demonstrates the performance of these methods on the VOC2006 dataset, with SVM generally performing better than the others at lower labeled sample ratios.](image)
PASCAL 2006 Object Categorization Dataset

VOC2006, $r=0.5$

Class. Acc.
Grad.: Labeled
Grad.: Unlabeled

Acc. and Grad.
With our current GPU implementation of random forest, one can get a 10 to 20 times speed up here. An additional 5 times speed up can be achieved by reducing the iterations to 2000.
With our current GPU implementation of random forest, once can get a 10 to 20 times speed up here. An additional 5 times speed up can be achieved by reducing the iterations to 2000.
Conclusions: Release version 1.0

- We proposed a multi-class semi-supervised boosting method based on margin maximizing and cluster prior regularizations.
- By directly addressing the multi-class problem and using efficient base learners, such as random forests, we showed that our algorithm not only out-performs other supervised and semi-supervised methods, but also achieves a high level of computational efficiency.
- Additionally, our method provides a mean to incorporate other knowledge sources, such as label priors, knowledge transfer priors, or human knowledge.
DAS-Forests

Hope to see many of you at Kyoto.

Amir Saffari, Christian Leistner, Horst Bischof
Learning with Functional Gradient Descent

\[ X^* = X_0 - \nu \sum_{t=1}^{T} L'(X_{t-1}) \]

Friedman et al., Annals of Applied Statistics, 2000
Learning with Functional Gradient Descent

\[ F^*(x) = F_0(x) - \nu \sum_{t=1}^{T} \frac{\partial L}{\partial F}(F_{t-1}(x)) \]

Friedman et al., Annals of Applied Statistics, 2000
PASCAL 2006 Object Categorization Dataset

VOC2006, \( r = 0.5 \)

Class. Acc.

\( \alpha \)

0.00 0.02 0.04 0.06 0.08 0.10

\( \nu \)

0.30 0.35 0.40 0.45 0.50 0.55 0.56 0.57

Amir Saffari, Christian Leistner, Horst Bischof (Institute for Computer Graphics and Vision, Graz University of Technology, Austria)
Experimental Settings

- RMSBoost is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.
- Base learners are tiny extremely randomized forests, each consisting of 10 trees.
RMSBoost is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.

Base learners are tiny extremely randomized forests, each consisting of 10 trees.

Boosting iterations set to be 10000.
Experimental Settings

- **RMSBoost** is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.
- Base learners are tiny extremely randomized forests, each consisting of 10 trees.
- Boosting iterations set to be 10000.
- Parameters are selected via 10-fold cross-validation.

Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.

For binary classification methods, we used a 1-vs-all strategy.

All results reported are average of 10 independent runs.

All boosting and RF methods are implemented in C++ and use ATLAS subroutines.
Experimental Settings

- RMSBoost is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.
- Base learners are tiny extremely randomized forests, each consisting of 10 trees.
- Boosting iterations set to be 10000.
- Parameters are selected via 10-fold cross-validation.
- Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.
Experimental Settings

- **RMSBoost** is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.
- Base learners are tiny extremely randomized forests, each consisting of 10 trees.
- Boosting iterations set to be 10000.
- Parameters are selected via 10-fold cross-validation.
- Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.
- For binary classification methods, we used a 1-vs-all strategy.
Experimental Settings

- **RMSBoost** is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.
- Base learners are tiny extremely randomized forests, each consisting of 10 trees.
- Boosting iterations set to be 10000.
- Parameters are selected via 10-fold cross-validation.
- Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.
- For binary classification methods, we used a 1-vs-all strategy.
- All results reported are average of 10 independent runs.
Experimental Settings

- **RMSBoost** is compared with: AdaBoost.ML, Kernel SVM, Multi-Switch TSVM, SERBoost, RMBoost.
- Base learners are tiny extremely randomized forests, each consisting of 10 trees.
- Boosting iterations set to be 10000.
- Parameters are selected via 10-fold cross-validation.
- Results of hierarchical k-means is averaged 10 times to estimate the cluster priors.
- For binary classification methods, we used a 1-vs-all strategy.
- All results reported are average of 10 independent runs.
- All boosting and RF methods are implemented in C++ and use ATLAS subroutines.
Amir Saffari, Christian Leistner, Horst Bischof (Institute for Computer Graphics and Vision, Graz University of Technology, Austria)

Regularized Multi-Class Semi-Supervised Boosting

PASCAL 2006 Object Categorization Dataset

VOC2006, \( r = 0.5 \)

Class. Acc.
Grad.: Labeled
Grad.: Unlabeled

Weights
Correct
Outliers
Exponential Loss

Example

The exponential loss $\ell(f(x)) = e^{-f(x)}$, is a Fisher-consistent loss, its estimated conditional probabilities can be written as

$$\hat{p}(y = i|x) = \frac{e^{f_i(x)}}{\sum_{j=1}^{K} e^{f_j(x)}},$$

which is a symmetric multiple logistic transformation.

The empirical risk is

$$\mathcal{L}(f(x), \mathcal{X}_l) = \sum_{(x,y) \in \mathcal{X}_l} e^{-f_y(x)}.$$  

Cluster Priors

∀x ∈ X_u, ∀i ∈ {1, · · · , K} : p_p(y = i|x).

We use the Kullback-Leibler (KL) divergence to measure the deviation of the model w.r.t. cluster prior

\[ \ell_c(f(x)) = D(p_p || \hat{p}) = -H(p_p) + H(p_p, \hat{p}). \] (13)

Using symmetric multiple logistic transformation as the probabilistic estimates of the model

\[ \ell_c(f(x)) = -p_p^T f(x) + \log \sum_{j=1}^{K} e^{f_j(x)}. \] (14)