Semi-supervised learning in natural language processing

Anders Søgaard

Course outline

1. Supervised learning
2. Unsupervised learning and semi-supervised learning
3. Learning from weighted data and transfer learning
4. Applications to dependency parsing
5. **Transfer learning in the blind**

Out-of-vocabulary effects

- One of the main reasons for performance drops when evaluating supervised NLP models on out-of-domain data is out-of-vocabulary (OOV) effects (Blitzer et al., 2007; Daume and Jagarlamudi, 2011).
- Spelling expansion, morphological expansion, dictionary term expansion, proper name transliteration, correlation analysis, and word clustering still leave us with "removed" dimensions.
- This is a potential source of error.

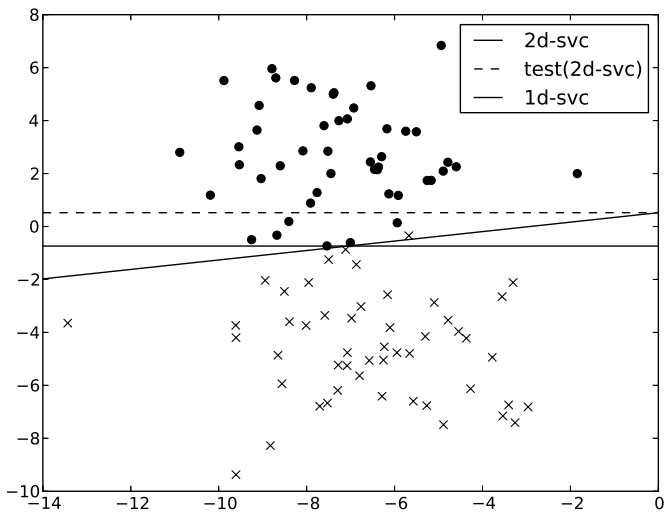

Figure: Optimal decision boundary is not optimal when one dimension is removed

What do we have in our toolbox?

- Global algorithms
- Inductive bias
- L_∞-regularization
  - $||\mathbf{w}||_\infty \doteq \max(|w_1|, \ldots, |w_n|)$

Note: We *c*annot use:

- Domain identification
- *KL*-divergence in random subspaces (Satpal and Saragawi, 2007)
- Structural correspondence learning (Blitzer et al., 2007)

Robust optimization

In robust optimization (Ben-Tal and Nemirovski, 1998) we aim to find a solution $\mathbf{w}$ that minimizes a (parameterized) cost function $f(\mathbf{w}, \xi)$, where the true parameter ξ may differ from the observed $\hat{\xi}$. The task is to solve

$$\min_{\mathbf{w}} \max_{\hat{\xi} \in \Delta} f(\mathbf{w}, \hat{\xi}) \tag{1}$$

with Δ all possible realizations of ξ. An alternative to minimizing loss in the worst case is minimizing loss in the average case, or the sum of losses:

$$\min_{\mathbf{w}} \sum_{\hat{\xi} \in \Delta} f(\mathbf{w}, \hat{\xi}) \tag{2}$$

Robust learning in random subspaces

The learning algorithms considered in this paper aim to learn models $\mathbf{w}$ from finite samples (of size M) that minimize the expected loss on a distribution ρ (with, say, N dimensions):

$$\min_{\mathbf{w}} \mathbb{E}_{\langle y,\mathbf{x}\rangle\sim\rho} L(y, \text{sign}(\mathbf{w}\cdot\mathbf{x})) \tag{3}$$

OOV effects can be seen as introducing an extra parameter into this equation. Let ξ be a binary vector of length M selecting what dimensions are removed. In NLP we typically assume that $\xi = \langle 1, \ldots, 1\rangle$ and minimize the expected loss in the usual way:

$$\min_{\mathbf{w}} \mathbb{E}_{\langle y,\mathbf{x}\rangle\sim\rho} L(y, \text{sign}(\mathbf{w}\cdot\mathbf{x}\cdot\xi)) \tag{4}$$

but if we have a set Δ of possible instantiations of ξ such that ξ can be any binary vector, minimizing expected loss is likely to be suboptimal, as discussed in the introduction. In this paper we will instead minimize average expected loss *under random subspaces*:

$$\min_{\mathbf{w}} \sum_{\hat{\xi}\in\Delta} \mathbb{E}_{\langle y,\mathbf{x}\rangle\sim\rho} L(y, \text{sign}(\mathbf{w}\cdot\mathbf{x}\cdot\hat{\xi})) \tag{5}$$

We refer to this idea as robust learning in random subspaces (RLRS).

Robust learning in random subspaces

$X = \{\langle y_i, \mathbf{x}_i \rangle\}_{i=1}^{N}$
for $r \in R$ **do**
  $\mathbf{w}^0 = 0, \mathbf{v} = 0, i = 0$
  $\xi \leftarrow$ **random.bytes**(M)
  for $k \in K$ **do**
    for $n \in N$ **do**
      if $\text{sign}(\mathbf{w} \cdot \mathbf{x} \cdot \xi) \neq y_n$ **then**
        $\mathbf{w}^{i+1} \leftarrow$ **update**$(\mathbf{w}^i)$
        $i \leftarrow i + 1$
      end if
    end for
  end for
  $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{w}^N$
end for
return $\mathbf{w} = \mathbf{v}/(N \times K)$

Figure: Robust learning in random subspaces

Robust learning in random subspaces

	P	P-RLRS	err.red	p-value	SGD	SGD-RLRS	err.red	p-value
25	67.2	70.1	0.09	< 0.01	75.2	75.7	0.02	~ 0.17
50	63.8	66.2	0.07	< 0.01	68.6	70.9	0.07	~ 0.02
75	73.2	75.3	0.08	< 0.01	76.3	78.9	0.11	< 0.01
100	72.0	73.3	0.05	~ 0.06	73.6	77.1	0.15	< 0.01
150	72.3	76.2	0.14	< 0.01	74.6	79.2	0.18	< 0.01
250	70.4	72.6	0.07	~ 0.02	75.0	78.7	0.15	< 0.01

Figure: Results on 20 Newsgroups

Robust learning in random subspaces

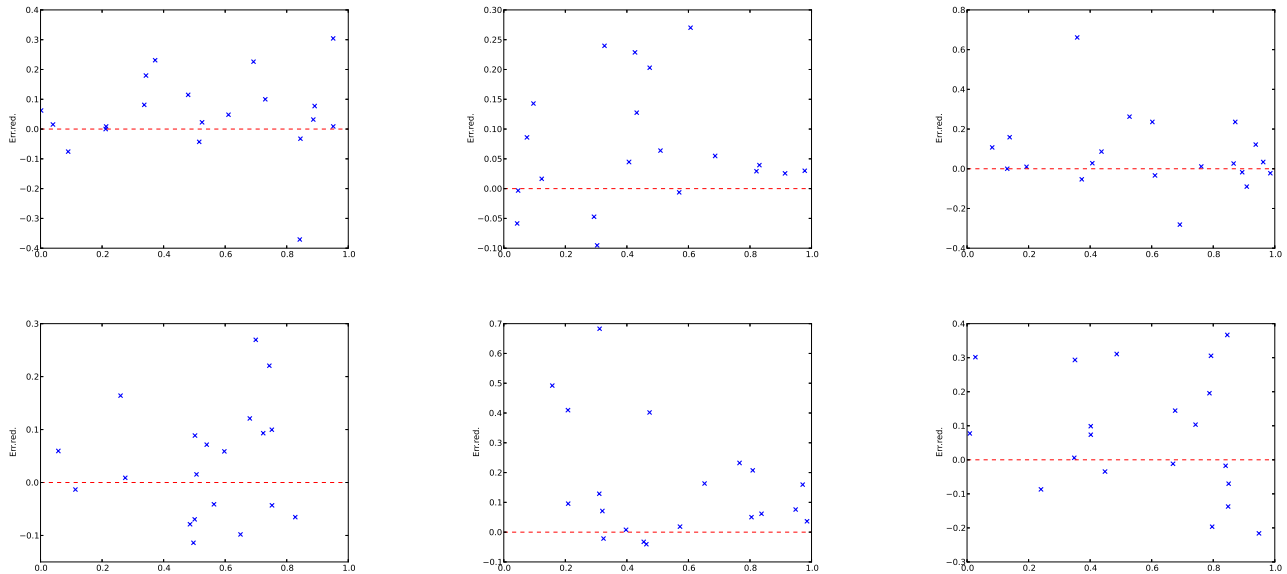

Figure: Plots of P-RLRS error reductions on 20 Newsgroups with $R = 25$ (upper left), $R = 50$ (upper right), $R = 75$ (lower left), $R = 100$ (lower mid), $R = 150$ (lower mid) and $R = 250$ (lower right).

Robust learning in random subspaces

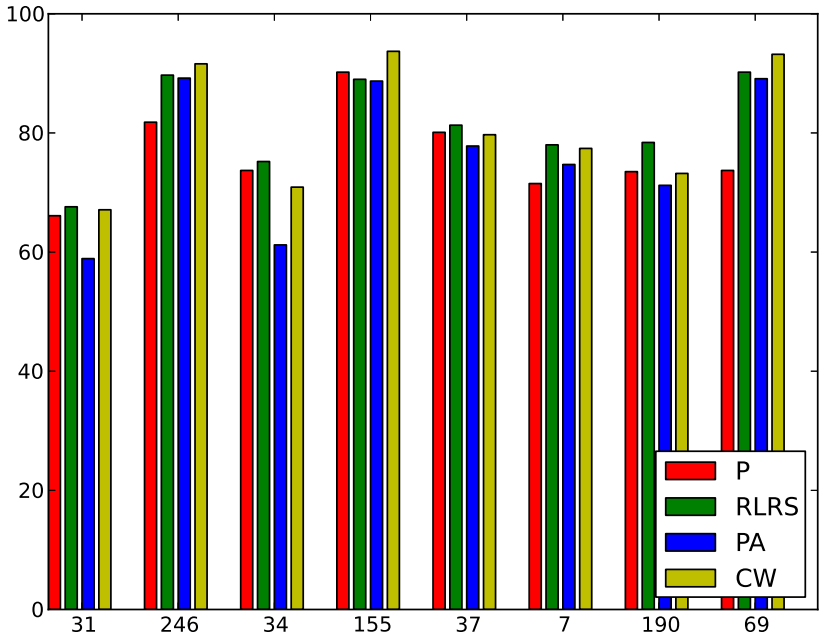

Figure: Classifier comparison on randomly sampled cross-domain problem instances in 20 Newsgroups

The results on the Web 2.0 Treebank were similar to those for 20 Newsgroups, and we observed consistent improvements with both robust perceptron and robust SGD learning. P-RLRS achieved an unlabeled attachment score of 68.8% on Twitter-dev (compared to 66.8%; err. red.: 0.06) and 76.7% on Sports-dev (compared to 73.6%; err.red.: 0.12).

Alternative approaches

- Confidence-weighted learning (CW) aggressively updates rare features and is often a strong baseline on biased data.
- The RLRS algorithm is essentially an ensemble learning algorithm, related to the **random subspaces method** (Ho, 1998), except averaging over multiple models rather than taking majority votes.
- Ensemble learning is known to lead to more robust models and therefore to performance gains in domain adaptation – so in a way our results are maybe not that surprising.
- However, our formulation of the RLRS objective enables direct optimization of robustness wrt. OOV effects. Secondly majority voting is not equivalent to averaging.
- **Feature bagging** (Sutton et al., 2006) combine an ensemble of subspace models in a product of experts to reduce weights under-training as an effect of indicative features swamping less indicative features, but use manually defined subsets of features rather than random subspaces.
- **Partitioned logistic regression** (Chang, 2008) learns logistic regression classifiers in non-random subspaces and combine them using a Naive Bayes classifier.

Alternative approaches

Typology from Caramanis et al. (2011):

Corrupted location	$P(X)$
Missing data	$P(X)$, $P(Y)$
Corrupted labels	$P(Y\|X)$

- **Robust learning by feature deletion**: Globerson and Roweis (2006) minimize the worst case loss with k deletions:

$$\max 1 - y_i \mathbf{w} \cdot \mathbf{x}_i \cdot (1 - \alpha_i)$$

  with $\alpha_1 \in \{0, 1\}$ and $\sum_j \alpha_{i,j} = k$
- Trafalis and Gilbert (2007) consider down-weighting of features in a similar framework.
- Caramanis and Mannor (2008) use robust optimization to learn under corrupted labels.

Alternative approaches

- Dekel and Shamir (2008) use robust learning by feature weighting and use L_∞-regularization instead of L_2-regularization (using linear programming).
- They report significantly better results than Globerson and Roweis (2006).
- Note you can randomly sample $(0, 1)$-reals to approximate their solution (using `numpy.random`).
- Terminology in Gamble (2007): The feature weight vector is known as the *pedigree* vector. Gamble (2007) devise a decision tree learning algorithm under pedigree, but assume perfect weighting.

The personal perceptron

- Key idea: Compute error prints for several models and find a relevant subset of models for each new instance.

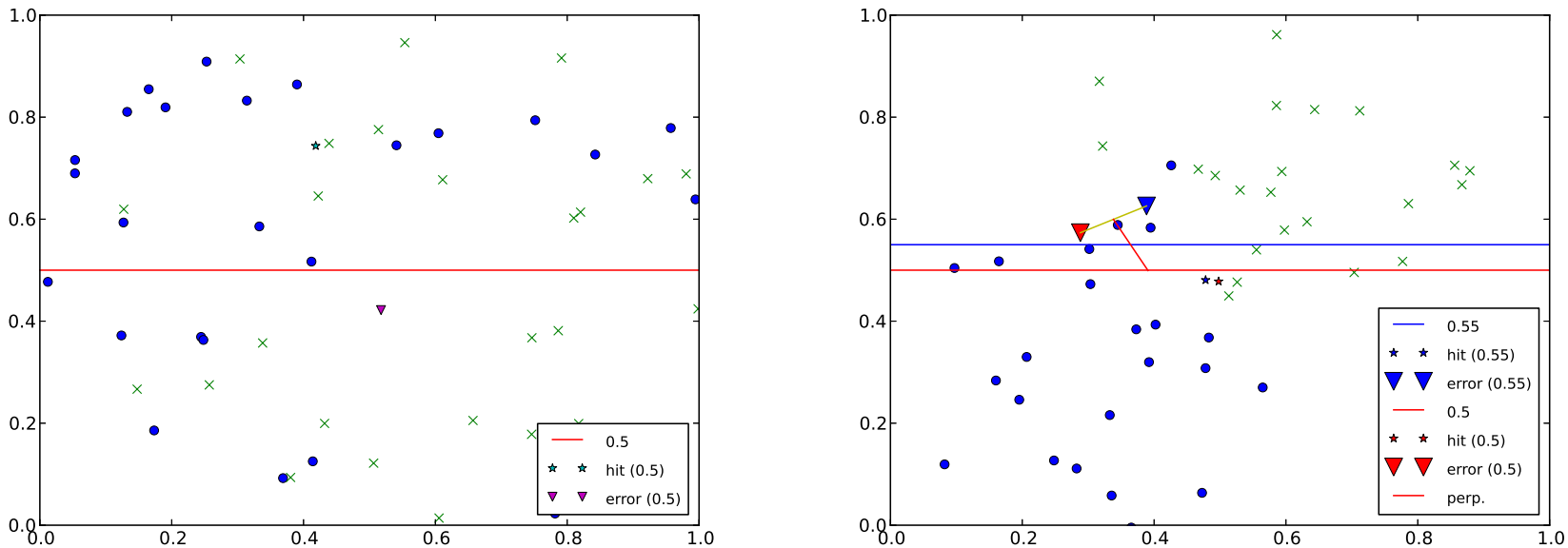

Figure: Plots of error and hit prints of decision boundaries (red and blue lines) in artificial data. Left: random. Right: mixture of Gaussians.

The personal perceptron

Input: $T = \{\langle \mathbf{x}_1, y_1 \rangle, \ldots, \langle \mathbf{x}_n, y_n \rangle\}$
Output: Predictions y_i for each test data point $\langle \mathbf{x}_i \rangle \in S$
$W = \emptyset$, $F = \emptyset$
$\mathbf{w} = \langle 0, \ldots, 0 \rangle$
for $1 \leq i \leq 100$ **do**
    for $\langle \mathbf{x}, y \rangle \in T$ **do**
        if $\text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq y$ **then**
            $W \leftarrow W \cup \{\mathbf{w} + \alpha(\text{sign}(\mathbf{w} \cdot \mathbf{x})x_i)\}$
            $F(\mathbf{w}) \leftarrow$ **average**$(\{\langle \mathbf{x}, y \rangle \in T \text{ s.t. } \text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq y\})$
        end if
    end for
end for
for $\langle \mathbf{x}, y \rangle \in S$ **do**
    $\mathbf{w} \leftarrow$ **average**$(\arg\max^k_{\mathbf{w}' \in W} M(F(\mathbf{w}'), \mathbf{x}))$
    return $\text{sign}(\mathbf{w} \cdot \mathbf{x})$
end for

Figure: The personal perceptron.

The personal perceptron

Theorem

For a personal perceptron with $p = k/|W|$ there is a k-nearest neighbor classifier with the same decision boundary with $2|W|$ labeled instances.

$\alpha = 0.1$	source	target	bl ($p = 1$)	$p = 0.8$	err.red.	LR	SVC
IES	Independent	Irish Times	85.42	89.58		**90.19**	80.39
	Independent	RTE	87.80	**90.24**		84.10	68.18
DEBATE2008	McCain	Obama	77.08	78.37		76.65	**80.70**
	Obama	McCain	83.08	**84.08**		83.00	82.51
Macro-AV			83.35	**85.57**	13.3	83.49	77.95
BL	$<$ 2003	$>$ 2003	89.00	**89.95**		74.53	78.77
	$>$ 2003	$<$ 2003	93.03	**93.53**		86.76	90.20
YOUTUBE	boys	girls	53.63	**54.19**		34.62	46.15
	boys	groups	45.05	46.06		36.10	**49.27**
Macro-AV			70.18	**70.93**	4.0	58.00	66.10
EPINION-60			65.41	68.31		**77.75**	**77.75**
EPINION-100			65.55	66.28		75.72	**76.01**
AMAZON	Apex	Apex	68.71	70.07		74.00	**74.67**
	Canon	Canon	73.26	73.26		**75.28**	**75.28**
Macro-AV			68.23	69.48	3.9	75.69	75.93

Results. All numbers in %. LR is L_1-regularized logistic regression, and SVC is L_1-regularized L_2-loss SVC (LibLinear).

The personal perceptron

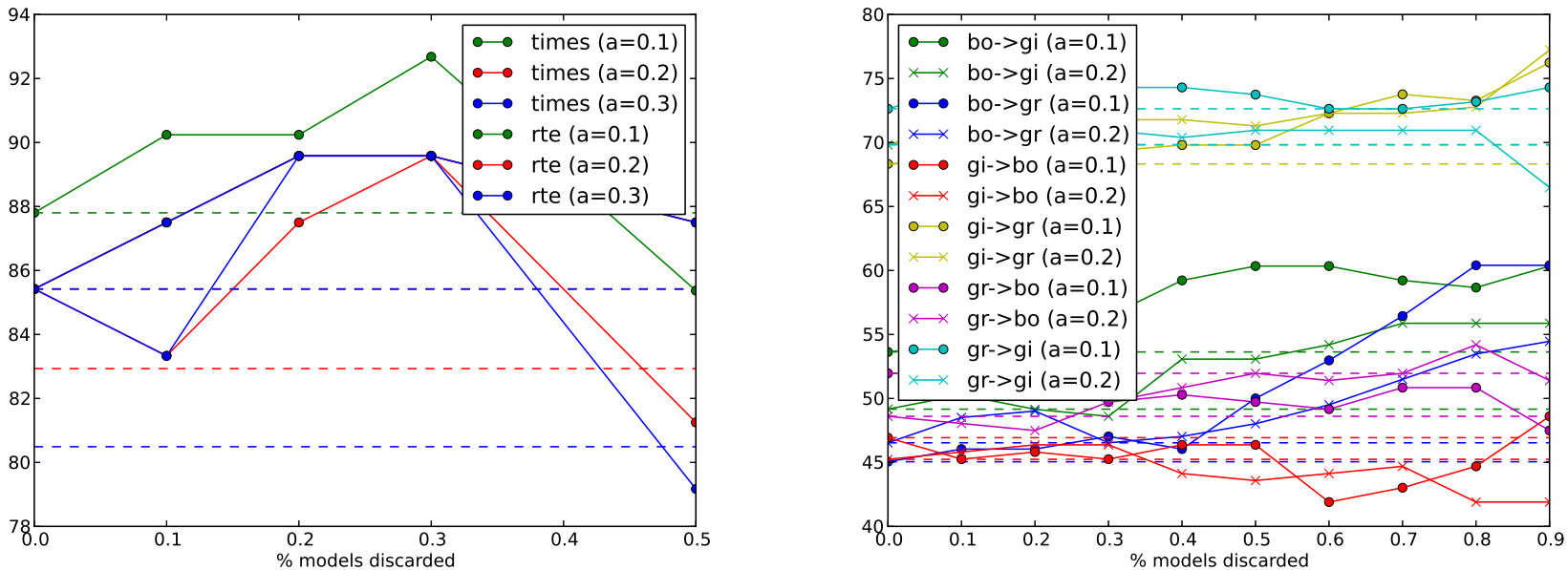

Figure: Results: IRISH ECON. SENTIMENT (left) and YOUTUBE-IDOL (right).

Alternative approaches

- Meta-learning with randomly corrupted examples.
- ?

Exercise: Any good ideas?

My to-do list:

Alternative approaches

- Meta-learning with randomly corrupted examples.
- ?

Exercise: Any good ideas?

My to-do list:

- Random feature weighting for robust learning.
- Combine importance weighting and robust optimization.
- Use hinge loss in the personal perceptron and make regularization depend on *KL*-divergence.

225 reasons why this line of research saves the planet (highlights)

1. It will bring language technology to the people, incl. Twitter users and Tagalog speakers.
73. Our algorithms will be adopted in bioinformatics, predictive analytics, etc.. Out goes financial crises, undiagnosed psychological diseases, etc.
223. It will keep the radical in me out of politics.
225. It will explain human cognition and enable us to build truly adaptive robots that can grow soy beans for us on Mars.

- You know what the best thing is?

It's fun, intuitive and easy. . .