# Online optimisation of log-linear weights in interactive machine translation

**Mara Chinea Rios, Germán Sanchis-Trilles, Daniel Ortiz-Martínez, Francisco Casacuberta**
Pattern Recognition and Human Language Technologies Group
Universitat Politècnica de València
Camino de Vera s/n
Valencia, Spain
mchinea@iti.upv.es, gsanchis@dsic.upv.es, dortiz@iti.upv.es, fcn@dsic.upv.es

## Abstract

Whenever the quality provided by a machine translation system is not enough, a human expert is required to correct the sentences provided by the machine translation system. In such a setup, it is crucial that the system is able to learn from the errors that have already been corrected. In this paper, we analyse the applicability of discriminative ridge regression for learning the log-linear weights of a state-of-the-art machine translation system underlying an interactive machine translation framework, with encouraging results.

**Keywords:** Interactive translation prediction, online learning, adaptive systems

## 1. Introduction

Adaptability is an important feature whenever statistical machine translation (SMT) systems are to be used within a computer assisted translation (CAT) framework. In such cases, the user expects the system to learn dynamically from its own errors, so that errors corrected once do not need to be corrected over and over again. Hence, the models need to be adapted *online*, i.e. without a complete retraining of the model parameters, since such retraining would be too costly. In this paper we will focus on adapting only the log-linear weights $\boldsymbol{\lambda}$ present in every state-of-the-art SMT system. Hence, the standard SMT equation (Brown et al., 1994) is complemented with a superindex denoting the current instant $t$:

$$\begin{aligned}\hat{\boldsymbol{y}} &= \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m^t h_m(\boldsymbol{x}, \boldsymbol{y}) \\ &= \underset{\boldsymbol{y}}{\operatorname{argmax}}\, \boldsymbol{\lambda}_t \cdot \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, g(\boldsymbol{x}, \boldsymbol{y}) \qquad (1)\end{aligned}$$

To simplify notation, we will omit subindex $t$ from input sentence $\boldsymbol{x}$ and output sentence $\hat{\boldsymbol{y}}$, although it is always assumed. $h_m(\boldsymbol{x}, \boldsymbol{y})$ represents an important feature for the translation of $\boldsymbol{x}$ into $\boldsymbol{y}$, $M$ is the number of models (or features) and $\lambda_m$ are the weights acting as scaling factors of the score functions. $g(\boldsymbol{x}, \boldsymbol{y})$ represents the score of a hypothesis $\boldsymbol{y}$ given an input sentence $\boldsymbol{x}$, and is not treated as a probability since the normalisation term has been omitted. Common feature functions $h_m(\boldsymbol{x}, \boldsymbol{y})$ include translation models, re-ordering models or the target language model. $\boldsymbol{h}$ and $\boldsymbol{\lambda}$ are estimated by means of training and development sets, respectively. However, the domain of such sets has an important impact on the final translation quality (Callison-Burch et al., 2011), and adaptation arises as an efficient way of alleviating this fact by using very limited amounts of in-domain data. We focus on two different CAT scenarios: standard post-edition (PE), and the more sophisticated interactive machine translation (IMT) (Barrachina et al., 2009) scenario. The main difference between PE and IMT is that in IMT the system provides improved completions after each user interaction, while in PE the system remains passive after providing the initial translation of the source sentence.

Similar work was performed in (Ortiz-Martínez et al., 2010), where an incremental version of the Expectation-Maximisation algorithm is used. However, such work focuses in the feature functions, while the present one focuses on the log-linear weights. Online adaptation of such weights is also studied in (Martínez-Gómez et al., 2012), although only applied to a conventional PE setup. As we will see later, the IMT setup yields challenges of its own.

## 2. Discriminative ridge regression

The main purpose of discriminative Ridge regression (Martínez-Gómez et al., 2012) (DRR) is that *good* hypothesis within a given $N$-best list score *higher*, and *bad* hypotheses score *lower*. It implements the estimation of $\boldsymbol{\lambda}$ as a regression problem between $g(\boldsymbol{x}, \boldsymbol{y})$, with $\boldsymbol{y} \in nbest(\boldsymbol{x})$ (i.e., the set of $n$ best hypotheses which can be derived from $\boldsymbol{x}$), and the translation quality of $\boldsymbol{y}$, $\mu(\boldsymbol{y})$. Let $\boldsymbol{y}^*$ be the hypothesis with the highest quality, but which might yield a lower score in Eq. 1[1]. Our purpose is to adapt the

[1] $\boldsymbol{y}^*$ does not necessarily match the reference translation $\boldsymbol{y}^\tau$ due to eventual coverage problems.

model parameters so that $\boldsymbol{y}^*$ is rewarded and achieves a higher score according to Eq. 1.

We define the *difference* in translation quality between the proposed hypothesis $\hat{\boldsymbol{y}}$ and the best hypothesis $\boldsymbol{y}^*$ in terms of a given quality measure $\mu(\cdot)$:

$$l(\hat{\boldsymbol{y}}) = |\mu(\hat{\boldsymbol{y}}) - \mu(\boldsymbol{y}^*)|, \qquad (2)$$

where the absolute value has been introduced in order to preserve generality. We also define the score difference between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}^*$ as:

$$\phi(\hat{\boldsymbol{y}}) = g(\boldsymbol{x}, \boldsymbol{y}^*) - g(\boldsymbol{x}, \hat{\boldsymbol{y}}). \qquad (3)$$

Ideally, we would like differences in $l(\cdot)$ to correspond to differences in $\phi(\cdot)$: if hypothesis $\boldsymbol{y}$ has a translation quality $\mu(\boldsymbol{y})$ that is very similar to the translation quality of $\mu(\boldsymbol{y}^*)$, we would like this to be reflected in translation score $g$, i.e., $g(\boldsymbol{x}, \boldsymbol{y})$ is very similar to $g(\boldsymbol{x}, \boldsymbol{y}^*)$.

For computing the new scaling factors $\boldsymbol{\lambda}_t$, the previously learnt $\boldsymbol{\lambda}_{t-1}$ is combined, for a certain learning rate $\alpha$, with an appropriate update step $\check{\boldsymbol{\lambda}}_t$, yielding (Martínez-Gómez et al., 2012):

$$\boldsymbol{\lambda}_t = (1 - \alpha)\boldsymbol{\lambda}_{t-1} + \alpha\check{\boldsymbol{\lambda}}_t. \qquad (4)$$

### 2.1. Discriminative ridge regression in PE

In a conventional post-editing scenario where the hypotheses are provided by a regular SMT system, the DRR algorithm requires an $N$-best list of hypotheses in decreasing order of score. Let $nbest(\boldsymbol{x})$ be such a list computed by our models for sentence $\boldsymbol{x}$. For adapting $\boldsymbol{\lambda}$, we define an $N \times M$ matrix $\mathrm{H}_{\boldsymbol{x}}$, where $M$ is the number of features in Eq. 1, containing the feature functions $\boldsymbol{h}$ of every hypothesis:

$$\mathrm{H}_{\boldsymbol{x}} = [\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}_1), \ldots, \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}_N)]' \,. \qquad (5)$$

Additionally, let $\mathrm{H}^*_{\boldsymbol{x}}$ be a matrix such that

$$\mathrm{H}^*_{\boldsymbol{x}} = [\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}^*), \ldots, \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}^*)]' \,, \qquad (6)$$

where all rows are identical and equal to the feature vector of the best hypothesis $\boldsymbol{y}^*$ within the $N$-best list. Then, $\mathrm{R}_{\boldsymbol{x}}$ is defined as

$$\mathrm{R}_{\boldsymbol{x}} = \mathrm{H}^*_{\boldsymbol{x}} - \mathrm{H}_{\boldsymbol{x}} \,. \qquad (7)$$

The key idea is to find a vector $\check{\boldsymbol{\lambda}}_t$ such that differences in scores are reflected as differences in the quality of the hypotheses. That is,

$$\mathrm{R}_{\boldsymbol{x}} \cdot \check{\boldsymbol{\lambda}}_t \propto \mathbf{l}_{\boldsymbol{x}} \,, \qquad (8)$$

where $\mathbf{l}_{\boldsymbol{x}}$ is a column vector of $N$ rows such that

$$\mathbf{l}_{\boldsymbol{x}} = [l(\boldsymbol{y}_1) \ldots l(\boldsymbol{y}_n) \ldots l(\boldsymbol{y}_N)]' \,, \ \forall \boldsymbol{y}_i \in nbest(\boldsymbol{x}). \qquad (9)$$

The objective is to find $\check{\boldsymbol{\lambda}}_t$ such that

$$\check{\boldsymbol{\lambda}}_t = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} |\mathrm{R}_{\boldsymbol{x}} \cdot \boldsymbol{\lambda} - \mathbf{l}_{\boldsymbol{x}}| \qquad (10)$$

$$= \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} ||\mathrm{R}_{\boldsymbol{x}} \cdot \boldsymbol{\lambda} - \mathbf{l}_{\boldsymbol{x}}||^2, \qquad (11)$$

where $||\cdot||^2$ is the Euclidean norm. Although Eqs. 10 and 11 are equivalent, Eq. 11 allows for a direct implementation thanks to the ridge regression[2], such that $\check{\boldsymbol{\lambda}}_t$ can be computed as the solution to the overdetermined system $\mathrm{R}_{\boldsymbol{x}} \cdot \check{\boldsymbol{\lambda}}_t = \mathbf{l}_{\boldsymbol{x}}$, given by

$$\check{\boldsymbol{\lambda}}_t = \left(\mathrm{R}'_{\boldsymbol{x}} \cdot \mathrm{R}_{\boldsymbol{x}} + \beta \mathrm{I}\right)^{-1} \mathrm{R}'_{\boldsymbol{x}} \cdot \mathbf{l}_{\boldsymbol{x}} \,, \qquad (12)$$

where a small $\beta$ is used as a regularisation term to stabilise $\mathrm{R}'_{\boldsymbol{x}} \cdot \mathrm{R}_{\boldsymbol{x}}$. $\beta = 0.01$ was used in the experiments described in this paper.

### 2.2. Discriminative ridge regression in IMT

In an IMT setting the quality metric to be used is no longer inherent to a single hypothesis, but to a complete wordgraph. In fact, it is quite common to measure the quality of a given IMT system by computing the amount of interactions required in order to modify the system's hypothesis so that it matches the reference. Once a single word has been introduced, the IMT system modifies the suffix, which implies that the number of interactions cannot be computed as a function of the hypothesis, but must be computed by first simulating the interaction procedure and is a function of a given wordgraph. Hence, since the metric to be optimised by online learning does not depend on a single-best hypothesis, the formulation of DRR needs to be reviewed.

At this stage, it would be reasonable to consider instead of a list of $N$-best hypotheses a list of $N$-best wordgraphs. However, the concept of $N$-best wordgraph is somewhat fuzzy. For this reason, instead of computing a true list of $N$-best wordgraphs we will obtain a set of $N$ scaling factors $\boldsymbol{\lambda}$, $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^n, \ldots, \boldsymbol{\lambda}^N\}$, and compute the wordgraph $W_{\boldsymbol{\lambda}^n}(\boldsymbol{x})$ associated to a given input sentence $\boldsymbol{x}$ and obtained for a certain set of scaling factors $\boldsymbol{\lambda}^n$. The resulting wordgraphs will not constitute a true list of $N$-best wordgraphs, but since the purpose of DRR is to reward those hypotheses (in this case wordgraphs) that score well, and penalise those that score worse, what is really important is to have wordgraphs with good quality, and others with worse. Hence, $\mathbf{l}_{\boldsymbol{y}}$ will be a column vector of $N$ rows such that

$$\mathbf{l}_{\boldsymbol{y}} = [l(W_{\boldsymbol{\lambda}^1}(\boldsymbol{x})) \ldots l(W_{\boldsymbol{\lambda}^n}(\boldsymbol{x})) \ldots l(W_{\boldsymbol{\lambda}^N}(\boldsymbol{x}))]' \qquad (13)$$

[2] Also known as Tikhonov regularisation.

where $l(W_{\boldsymbol{\lambda}^n}(\boldsymbol{x}))$ is the quality metric associated to wordgraph $W_{\boldsymbol{\lambda}^n}(\boldsymbol{x})$.

In addition, when considering DRR within an IMT setting matrix $\mathrm{H}_{\boldsymbol{x}}$ also needs to be redefined, since the features that need to be considered in this case no longer correspond to hypotheses in the $N$-best list, but to the wordgraphs generated with $\boldsymbol{\Lambda}$. Since a certain wordgraph $W_{\boldsymbol{\lambda}^n}(\boldsymbol{x})$ does not have a single set of features, but rather one feature vector for each one of the paths through the wordgraph, we will consider for building $\mathrm{H}_{\boldsymbol{x}}$ the feature vector $\boldsymbol{h}$ of the best path in $W_{\boldsymbol{\lambda}^n}(\boldsymbol{x})$, i.e., the feature vector of the best hypothesis in $W_{\boldsymbol{\lambda}^n}(\boldsymbol{x})$. Abusing notation and with the purpose of keeping notation unclogged, let $\boldsymbol{h}_{\boldsymbol{\lambda}^n}$ be such feature vector. Then, $\mathrm{H}_{\boldsymbol{x}}$ is defined for the IMT case as

$$\mathrm{H}_{\boldsymbol{x}} = \left[\boldsymbol{h}_{\boldsymbol{\lambda}^1}, \ldots, \boldsymbol{h}_{\boldsymbol{\lambda}^N}\right]'. \tag{14}$$

Equivalently, $\mathrm{H}^*_{\boldsymbol{x}}$ is defined in this case as

$$\mathrm{H}^*_{\boldsymbol{x}} = \left[\boldsymbol{h}_{\boldsymbol{\lambda}^*}, \ldots, \boldsymbol{h}_{\boldsymbol{\lambda}^*}\right], \tag{15}$$

with $\boldsymbol{h}_{\boldsymbol{\lambda}^*}$ being the feature vector of the best hypothesis of wordgraph $W_{\boldsymbol{\lambda}^*}(\boldsymbol{x})$, and $W_{\boldsymbol{\lambda}^*}(\boldsymbol{x})$ the wordgraph with the best performance from among those derived from $\boldsymbol{\Lambda}$.

In the present work, we explored two different strategies for computing $\boldsymbol{\Lambda}$:

1. Sampling $\boldsymbol{\Lambda}$ from a Gaussian distribution with the mean centred on the set of weights obtained in training time.
2. Running a Simplex optimisation procedure to compute the best set of weights for each sentence. Since the Simplex algorithm is an iterative algorithm, $\boldsymbol{\Lambda}$ will be composed of those weights that arise in each one of the iterations.

## 3. Experiments

The experiments conducted in this work were performed by using the English-Spanish data provided for the 2013 Workshop on Machine Translation [3]. The initial set of log-linear weights was adjusted by means of MERT (Och, 2003) on the development sets of 2008-2010. This initial system will be referred to as `baseline`. Since a true IMT experiment is too costly for experimentation purposes because it would require a human evaluation, we simulated this procedure by evaluating the system with the 2011 test set, and adapting the weights online after the system's performance was assessed for each bilingual sentence. System performance was evaluated by means of the Key Stroke Mouse Action (Barrachina et al., 2009) (KSMR) ratio, which measures the amount of key strokes and mouse actions that a user would need to perform in order to transform the original hypothesis provided by the system into the reference.

Table 1: Results in KSMR of the different online learning strategies studied

| Method | $\alpha$ | $N$ | KSMR |
|---|---|---|---|
| baseline | — | — | 40.6 |
| original | 0.01 | 5000 | 42.8 |
| gaussian | 0.001 | 201 | 40.9 |
| simplex | 0.0001 | 70 | 40.4 |

The open-source SMT toolkit Moses (Koehn et al., 2007) was used for building the initial SMT system, and the wordgraphs produced by the decoder were then used for producing the completion hypotheses by using an in-house IMT engine. Note that, by tuning the SMT system on the same domain as the test data, the improvements reported in this paper are not a product of a topic-adaptation process, but rather are inherent to the online process itself, seen as the ability of the system to adapt itself to the current test set being translated.

The results of the different online learning approaches can be seen in Table 1, together with the optimal learning rates for each one of them and the size of the $N$-best list used ($|\boldsymbol{\Lambda}|$ in the case of the Gaussian and simplex strategies). In this table, `baseline` displays the KSMR achieved without any online learning procedure enabled. `original` refers to the online learning of the log-linear weights as originally defined for PE (see Section 2.1.). `Gaussian` refers to the sampling strategy based on sampling from a Gaussian distribution (Section 2.2.). Finally, `simplex` refers to the Simplex strategy described in Section 2.2.. The learning rates were established previously in preliminary investigation. Note that the size of $N$ is different in each one of the methods. The reason for this is that, while generating a large amount of $N$-best translation hypotheses and assessing their quality by means of conventional SMT metrics is relatively cheap, building $N$-best word-graphs, as described in Section 2.2., and assessing their quality is fairly expensive, since it requires a full IMT simulation for finding out the amount of corrections required. Furthermore, the size of $\boldsymbol{\Lambda}$ in simplex is not a meta-parameter which can be fine-tuned, since the simplex algorithm was run until convergence. This means that the value of 70 was an average, i.e., not the actual amount of $\boldsymbol{\Lambda}$ considered for every sentence.

[3] http://www.statmt.org/wmt13

Concerning the results shown in Table 1, it can be seen that the Simplex strategy is the one that yields the best results and the only one that is able to improve over the initial baseline. Note that previous work (López-Salcedo et al., 2012) also reported improvements with the Gaussian strategy. However, such improvements were obtained by considering $|\mathbf{\Lambda}| = 500$. In the present work, similar improvements are obtained by considering only $|\mathbf{\Lambda}| = 70$.

## 4. Conclusions

In the present paper we have presented two possible adaptations of the discriminative Ridge regression algorithm for its application in an interactive machine translation framework. First, we have empirically shown that the original definition is not valid in such a framework, and then we have evaluated two possible alternatives. Among them, the one providing the best results and with the least amount of computational resources is a strategy that relies on the iterations performed by the simplex algorithm when computing the optimum set of weights for each individual sentence. The results obtained do not provide large gains, but prove that there is room for improvement in this direction.

## 5. Acknowledgments

Work supported by the European Union $7^{\text{th}}$ Framework Program (FP7/2007-2013) under the CasMaCat project (grant agreement nr. 287576), by Spanish MICINN under grant TIN2009-14205-C04-02 and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014).

## 6. References

S. Barrachina et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1994. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar F. Zaidan, editors. 2011. *Proceedings of the Sixth Workshop on SMT*. Association for Computational Linguistics, Edinburgh, Scotland, July.

Philipp Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL Demo and Poster Sessions, 2007*, pages 177–180, June 25–27.

Francisco-Javier López-Salcedo, Germán Sanchis-Trilles, and Francisco Casacuberta. 2012. Online learning of log-linear weights in interactive machine translation. *Advances in Speech and Language Technologies for Iberian Languages*, 328(2012):277–286.

P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta. 2012. Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, 45(9):3193–3203.

Franz J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proc. of the 41st annual conf. of the ACL*, pages 160–167, July 7–12.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *Proc. of NAACL*, pages 546–554, June 2–4.