Context Change and Versatile Models in Machine Learning

José Hernández-Orallo
Universitat Politècnica de València
jorallo@dsic.upv.es

ECML Workshop on Learning over Multiple Contexts
Nancy, 19 September 2014
Spot the difference
Outline

- Context change: occasional or systematic?
- Contexts: types and representation
- Adaptation procedures
- Versatile models
- Kinds of reframing
- Evaluation with context changes
- Related areas
- Conclusions
Context change: occasional or systematic?

- Contexts (domains, data, tasks, etc.) change.

- Has the model been prepared to be adapted to other contexts?
- Did we sufficiently generalise from context A?
- Is the adaptation process ad-hoc?
- Should we throw the model away and learn a new one?
Context change: occasional or systematic?

- Contexts change *repeatedly*...
Context change: occasional or systematic?

How can treat context change in a more systematic way?

1. Determine which *kinds* of contexts we will deal with.
2. Describe and *parameterise* the context space.
3. Use *versatile* models that are better prepared for changes.
4. Define appropriate *adaptation procedures* to deal with the changes.
5. Overhaul *evaluation tools* for a range of contexts.
Example of an area that does this: ROC analysis

1. The *kinds* of contexts dealt with are known as ‘operating conditions’.
2. Contexts are *parameterised* as skews (class and cost proportions).
3. Ranking models provide more *versatility* than crisp classifiers.
4. Models are *adapted* to contexts by changing the threshold.
5. ROC curves and other plots and metrics *evaluate* model behaviour for a range of contexts, assuming a given threshold choice method will be used.
Contexts: types and representation

- Data shift (covariate, prior probability, concept drift, ...).
  - Changes in $p(X)$, $p(Y)$, $p(X|Y)$, $p(Y|X)$, $p(X,Y)$
- Costs and utility functions.
  - Cost matrices, loss functions, reject costs, attribute costs, error tolerance...
- Uncertain, missing or noisy information
  - Noise or uncertainty degree, %missing values, missing attribute set, ...
- Representation change, constraints, background knowledge.
  - Granularity level, complex aggregates, attribute set, etc.
- Task change
  - Regression cut-offs, bins, number of classes or clusters, quantification, ...
**Contexts: types and representation**

- Is the context absolute or relative to the original context?
  - **Absolute:**
    - E.g. in context B positive class is three times more likely than negative class.
  - **Relative:**
    - E.g. positive class in context B is three times more likely than in the original context A.

- Is the context given or inferred?
  - **Given:**
    - E.g.: cost information, cut-off, attribute set, …
  - **Inferred (from the deployment data or a small labelled dataset):**
    - E.g.: p(X), % of missing data, class proportion, …

- Is the context changing once for each dataset or for each example?
  - **If the context changes for each example,**
    - a non-systematic approach becomes very problematic.
    - context inference is more difficult.
A context $\theta$ is a tuple of one or more values, discrete or numerical, that represent or summarise contextual information.

**Examples:**

- Contexts are cities and temperatures:
  - $\theta_A = \langle \text{Nancy, 20} \rangle$ is a context, while $\theta_B = \langle \text{Valencia, 30} \rangle$ is another context.

- Contexts are cost proportions.
  - $\theta = \langle c \rangle$ where $c$ is a cost proportion or a skew or a class prior.

- Contexts are attribute granularity.
  - $\theta = \langle \text{week, city, women, category} \rangle$ to specify granularities for dimensions time, store, customer and product, respectively.

- Contexts are error tolerance.
  - $\theta = \langle 20\% \rangle$ to specify that up to 20% of regression error is acceptable.
### Adaptation procedures

- **Retraining**: Train another model using the available (old and possibly new) data and the new context into account.

  - The original model is discarded (no knowledge reuse).
  - If there is plenty of new data, this is a reasonable approach.
  - Not very efficient if the context changes again and again (e.g., for each example).
  - The training data may have been lost or may not exist (the models may have been created or integrated by human experts).
  - May lead to context overfitting.
Adaptation procedures

- **Retraining with knowledge transfer**: Train another model using (or transferring) part of the knowledge from the original context.
  - Parts of the original model or other kinds of knowledge is still reused.
    - Instance-transfer (Pan & Yang 2010).
    - Feature-representation-transfer (Pan & Yang 2010).
    - Parameter-transfer (Pan & Yang 2010).
    - Relational-knowledge-transfer (Pan & Yang 2010).
    - Prediction-transfer: the original model is used to label examples (mimetism).
Adaptation procedures

- **Context-as-feature:** the parameters of the context are added as features to the training data.

  - Model is reused. Training data can be discarded.
  - Requires several contexts during training in order to generalise the feature.
  - Makes more sense when there is a different context per example.
  - The context works as a “second-order” feature, regulating how the other features should be used. Not many machine learning techniques are able to deal with this kind of pattern.
Adaptation procedures

- **Reframing**: process of applying an existing model to the new operating context by the proper transformation of inputs, outputs and/or patterns.
  
  - Model is reused. Training data can be discarded.
  - The reframing process is designed to be systematic (and automated), using $\theta$.
  - Only one original context is needed.
A versatile model is a model that captures more information than needed and/or generalises further than strictly necessary for the original context in order to be prepared to be reframed for a new context.

Examples:
- Generative models over discriminative models.
- Scoring classifiers over crisp classifiers.
- Models gathering statistics (means, co-variances, etc.) about the inputs/output.
- Unpruned trees over pruned trees.
- Models that take different kinds of features.
- Hierarchical clustering over clustering methods with a fixed no. of clusters.
Versatile model

How can we generate more versatile models?

- Redefine learning algorithms and models, so that they include more information.
  - E.g., keep some of the information used during learning (densities, clusters, alternative rules, etc.).

- Annotate models as a postprocess.
  - E.g., include statistics at each split of a decision tree.

- Enrich them using the training or a validation dataset.
  - E.g., calibration.

The knowledge is not gathered in a separate way from the model (as in knowledge transfer)

This knowledge is embedded in the model so that its adaptation can be automated.
Kinds of reframing

- Output reframing.
  - Outputs are reframed.

Examples and other names:
- Use of threshold choice methods with scoring classifiers (as in ROC analysis).
- Binarised regression problem (cutoff from regression to classification).
- Shifting the output to minimise expected cost in regression. By tuning (Bansal et al. 2008) or reframing (Hernandez-Orallo 2014).
Kinds of reframing

- Input reframing.
  - Inputs are reframed.

- Examples and other names:
  - Use of quantiles (El Jelali et al. 2013).
  - Feature shift (Ahmed et al. 2014)
Kinds of reframing

- Structural reframing.
  - The model is reframed.

Examples and other names:
  - Relabelling (e.g., using a small labelled dataset)
  - Postpruning (during deployment).
Evaluation with context changes

- The performance of a model $m$ on a data $D$ can be evaluated for a single context $\theta$ using a reframing procedure $R$.

\[ Q(R, m, D, \theta) \]

- If contexts change systematically, we want to see model performance using a reframing procedure for a range of operating contexts:
  - With a context plot: context on one or more axes and $Q$ on another axis.
    - Dominance regions can be visualised.
  - How can we summarise a curve?

\[ L(R, m, D, C, w) \triangleq \int_{\theta \in C} Q(R, m, D, \theta) w(\theta) d\theta \]

- A range of contexts is given by a set of contexts $C$ and a distribution $w$ over them.
Evaluation with context changes

- Example: classical cost curves are context plots.
  \[ Q(t; c) = 2\{c\pi_0(1 - F_0(t)) + (1 - c)\pi_1 F_1(t)\} \]
  \( c \) is the context

- Many other curves are possible if the reframing procedure is different.
- In this case, several threshold choice methods on the right.

\( F_0 \): TPR or sensitivity if threshold is set on \( t \)
\( 1 - F_1 \): TNR or specificity if threshold is set on \( t \)
Evaluation with context changes

- Example: regression asymmetric costs
  - For instance, using asymmetric absolute cost (Lin-Lin) for regression:
    \[
    \ell^A_\alpha(\hat{y}, y) \triangleq \begin{cases} 
    2\alpha(y - \hat{y}) \\
    2(1 - \alpha)(\hat{y} - y)
    \end{cases}
    \]
    \(\alpha\) is the context
  - Regression cost curves:
Example: REC curves (tolerance level)

\[ Acc = 1 \left( |y - \hat{y}| \leq tolerance \right) \]

*tolerance is the context*
Evaluation with context changes

- Example: attribute shift
  - One or more attributes have a constant shift (Ahmed et al. 2014):
    \[ x' \leftarrow x + \beta \]
    \( \beta \) is the context
  - In this context plot, we compare retraining with a reframing approach.
Evaluation with context changes

- Example: noise levels (Ferri et al. 2014)
  - Data may have different levels of noise.

\(\text{level of noise is the context}\)
Evaluation with context changes

- Example: misclassification cost (MC) vs attribute test cost (TC):

\[ JC_i \triangleq \alpha_i \cdot MC_i + (1 - \alpha_i) \cdot TC_i \]

\( \alpha \) is the context

- Different attribute subsets lead to different cost lines:
Evaluation with context changes

- Example: multidimensional (attributes are hierarchical dimensions)
  \[ \langle l_1, \ldots, l_d \rangle, \text{ with each } l_i \in h(X_i) \]

- To make the plot simpler, we use a Reduction Coefficient (RC), which expresses the level of aggregation of the data (from 0 to 1).

RC is a simplification of the context.
Evaluation with context changes

- Example: multilabel (Al-Otaibi 2014)
  - Costs per each label are introduced
  - Different colours represent different threshold choice methods.

Curves are for cases where the costs are equal for all labels. Clouds are for cases where cost are different for each label (but the average is on the x-axis).
Evaluation with context changes

- Example: clustering algorithms depending on number of clusters
  - Different clustering algorithms: The no. of clusters is the context
    - Kmeans is rerun (retrained) with different values for K.
    - Hierarchical methods are versatile models working for several contexts.

The “cost” can be any clustering performance metric, such as Davies-Bouldin index, the Dunn index or the Silhouette coefficient.
Related areas

- Data shift.
- Domain adaptation.
- Cost-sensitive learning.
- Learning with noisy data.
- Transfer learning.
- Multi-task learning.
- Transportability.
- Context-aware computing.
- Mimetic models.
- Theory revision.
- ROC analysis and cost plots.
Related areas

- A reframing perspective is distinctive:
  - Contexts are clearly identified and parameterised.
  - It’s not a one-to-one occasional transfer but a systematic application.
  - There can be several reframing methods for the same model and data, leading to different results.
  - Models are learnt in one context and task but kept for many contexts.
  - Performance is analysed in a range of contexts.
  - Models are reused.
Conclusions

- Disposing validated models again and again is not cost-efficient.
  - Reusing models seems more appealing.
- Versatile models should be as general as possible to cope with a range of contexts.
  - Validation has to take this range of contexts into account.
- Model deployment is crucial.
  - Models become good or bad for a context depending on the deployment procedure we are using.
- But don’t be blinded by reframing.
  - We should always consider the trade-off between retraining and reframing (and other possible options).