Laboratory Experience 3 - Mandatory

Task  The aim of this experience is to get familiar with the transfer learning techniques discussed in [2]. Specifically, you will have to replicate the results in Figure 2 (a,g) and figure 5, involving four different learning algorithms:

- the no-transfer Weighted Least-Squares SVM (WLS-SVM) baseline
- the Single Knowledge Transfer (KT) baseline
- the Average-KT baseline
- the Multi-KT algorithm

Data  The experience makes use of a subset of the Caltech 256 dataset [1], which can be downloaded from: [http://www.idiap.ch/ftp/courses/EE-700/material/experience3/KTcaltech256.tar.gz](http://www.idiap.ch/ftp/courses/EE-700/material/experience3/KTcaltech256.tar.gz)

The KTcaltech256/data directory contains:

- LBP, PHOG, RECOV, SIFT: every directory includes 18 mat files, each of them containing a $n \times m$ matrix, where $n$ is the number of samples of the object and $m$ is the dimensionality of the image descriptor.

- prior_models_4class.mat, prior_models_6class.mat, prior_models_10class.mat: these three files contains the the prior knowledges for three different class groups:
  1. four unrelated classes
  2. six related classes
  3. ten mixed classes

as described in the paper.
**Code**

All the code that is necessary to run the experience is included in the KTExperience.m Matlab class, downloadable from:
http://www.idiap.ch/ftp/courses/EE-700/material/experience3/KTExperience.m

You are asked to complete the code in some relevant points, remembering that:

1. Each line of code to be completed contains a specific suggestion that will be helpful
2. The WLS-SVM problem in eq. (8) of the paper can be easily solved by inverting the matrix

\[
G = \begin{bmatrix} K + \frac{1}{C}W & 1 \\ 1' & 0 \end{bmatrix}
\] (1)

3. The Single-KT optimization problem in eq. (10) can be solved using the matricial form:

\[
\begin{bmatrix} K + \frac{1}{C}W & 1 \\ 1' & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} y - \beta w \cdot \Phi(X) \\ 0 \end{bmatrix}
\] (2)

where \( \Phi(X) \) represents the matrix of training points in feature space

4. The solution of the Multi-KT optimization problem in eq. (11) can be found by solving the system:

\[
\begin{bmatrix} K + \frac{1}{C}W & 1 \\ 1' & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} y - \sum_j \beta_j w_j \cdot \Phi(X) \\ 0 \end{bmatrix}
\] (3)

5. the Multi-KT optimal solution \( w^* \) is of the form:

\[
w^* = \sum_i \alpha_i \phi(x_i) + \sum_j \beta_j w_j
\] (4)

**Experiments**

Use the developed code to run each experiment ten times, on ten different training/testing splits. Moreover, instead of plotting the results using only up to 6 samples of the target class for training, plot the results obtained with up to 30 target training samples / class. Finally, use the method:

`KTExperience.plotClassDistance(beta,classes)`

to plot, for each experiment, a class similarity map, similar to this:
Comment the results w.r.t. the computational costs and the benefits w.r.t. the baselines.

References
