APÉNDICE

Por último, se ha decidido finalizar este Proyecto Final de Carrera incluyendo el código que se ha utilizado para ejecutar las simulaciones. Debido a que ALIZE está implementado en un entorno Linux, se ha codificado un script en Shell para llevar a cabo todas las pruebas experimentales. Para facilitar las ejecuciones, se ha trabajado con archivos de tipo Makefile, lo cual nos ha permitido incluir todo el código en un único archivo y ejecutar aquellas partes que nos interese en cada momento haciendo uso de las propiedades del make en Linux. En concreto, se ha implementado dos archivos de tipo Makefile. Uno de ellos nos permitirá ejecutar todas las simulaciones necesarias en GMM para calcular los valores óptimos de cada una de las posibles condiciones: número de mezclas gaussianas, normalización puntuaciones, dimensión de los vectores cepstrales, etc. Una vez seleccionada las condiciones óptimas, con el segundo se procede a obtener los resultados de las restantes técnicas y de las posibles fusiones. El departamento de Voz e Imagen de la Universidad de Swansea me ha permitido hacer uso de parte del código elaborado en años anteriores por otros compañeros, con lo cual una parte del código de este proyecto se corresponde con modificaciones de antiguos scripts y adaptaciones de éstos a las condiciones del NIST para el año 2005 y a las especificaciones de este proyecto.

A continuación se procede a mostrar el código correspondiente al primer archivo de tipo Makefile que se ha utilizado en la primera parte de las simulaciones, en las que se llevaron a cabo diversas pruebas experimentales con el objetivo de obtener los parámetros y condiciones óptimas para GMM (las condiciones seleccionadas iban a ser las establecidas para el resto del proyecto).

```
################################################
#     NIST 2005 Eval with Alize GMM only
#     ******************************
#     Usage: make -j<nbProc> <target> options
#     Default options in opt/file_options
#     Benoit Fauve
#     David Bonomo Laynez
################################################

CD=ls
include opt/cond$(CD)

include opt/Makefile_options

FEAT=m33
include opt/FEAT_$(FEAT)
EXTMODE=
include opt/EXT$(EXTMODE)


FEATEXT=$(FEATT).norm.prm
GMMEXT=$(GMMT).gmm
FEATFAEXT=$(FAT).norm.prm
```

```
REST=$(GDR).$(TYPE)$(GMMT)
RESTNT=$(REST).t$(TNT)
M=sgmm
include opt/MOD_$(M)

TYPEALL=$(GDR).$(TYPE).t$(TNT).w$(WLDTYPE)


#######################
# Function definitions#
#######################

SCORING=$(BIN)Scoring --mode NIST --threshold $(THRESHOLD) --
segTypeTest $(TST) --trainTypeTest $(TRN) --adaptationMode n

TRAINTARGET=$(BIN)TrainTarget --config $(CFG)TrainTarget.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(GMMEXT) --saveMixtureFileExtension $(GMMEXT) --featureServerMask
$(MASK) --vectSize $(SIZE) --loadFeatureFileExtension $(FEATEXT) --
MAPRegFactorMean $(RF)

COMPUTETEST=$(BIN)ComputeTest --config $(CFG)target_seg.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE)  --
loadFeatureFileExtension $(FEATEXT) --gender $(GDR)



#########################
# Usage of the makefile #
#########################
example1:
      @echo "Display text for example 1"

example2: log/example2.log
log/example2.log:
      @echo "Display text for example 2"
      @echo "action from example2 are logged in log/example2.log"
      @echo "once run you have to remove log/example2.log to run it
again"
      echo example2> $@

example3: example1 example2

all:
      @echo "*****************************************************"
      @echo "******      Launching a NIST 2005 - 06 Evaluation
      ******"
      @echo "*****************************************************"
      @echo "Usage: make <target> options"
      @echo "-------------------------------------------------------
------------------"
      @echo "There is a lot of targets, please read the makefile to
know them, here is some examples:"
      @echo "feature_extract, feature_norm, world, models, tests,
tnorm_tests, ztnorm, decision, ....."
      @echo ""
      @echo "Options:"
      @echo "make -j <number of targets at a time> -> useful for
multiprocessor machines"
```

```
	@echo "make -n -> prevent execution, just print the steps it
would do"
	@echo "*****************************************************"


#######################################################################
# Make variables that include lst files depending on their type       #
#######################################################################

LST_TAR= $(wildcard lst/target_$(GDR).$(TRNTYPE).lst)
LST_WLD= $(wildcard lst/world_$(GDR)_$(WLDTYPE).lst)
LST_COH= $(wildcard lst/world_$(GDR)_$(COHORTTYPE).lst)
LST_IMP= $(wildcard lst/imp_$(GDR)_$(TNT).lst)
LST_SEG= $(wildcard lst/seg_$(GDR).$(TYPE).lst)
LST_ANY= $(wildcard lst/$(LISTNAME))
LST_Z= $(wildcard lst/znorm_$(GDR).$(ZNT).lst)

LSTS_TAR= $(wildcard target_$(GDR).$(TRNTYPE))
LSTS_WLD= $(wildcard world_$(GDR)_$(WLDTYPE))
LSTS_IMP= $(wildcard imp_$(GDR)_$(TNT))
LSTS_SEG= $(wildcard seg_$(GDR).$(TYPE))
LSTS_ANY= $(wildcard $(LISTNAME))

LSTS_ALL=all.$(TYPEALL)
LST_ALL=lst/all.$(TYPEALL).lst


############################
#### create full list  ####
############################

builtlist:
	cat $(LST_TAR) > $(LST_ALL)
	cat $(LST_SEG) >> $(LST_ALL)
	cat $(LST_COH) >> $(LST_ALL)
	cat $(LST_IMP) >> $(LST_ALL)

doall: generatefeatall world2test

generatefeatall: generatefeatwldimp generatefeattest
generatefeatwldimp: generatefeatwld generatefeatimp
generatefeatwld: extractwld energywld featnormwld
generatefeatimp: extractimp energyimp featnormimp
generatefeatz: extractz energyz featnormz
generatefeatcoh: extractcoh energycoh featnormcoh
generatefeattest: generatefeatseg generatefeattarget
generatefeatseg: extractseg energyseg featnormseg
generatefeattarget: extractclient energyclient featnormclient
generatefeatlst: extractlst energylst featnormlst

extractall: extractwld extractclient extractseg extractimp

world2test: world target imp tests
train2test: target imp tests
tests: target_seg post_nonorm imp_seg post_tnorm
world2testnonorm: world target target_seg post_nonorm
```

```
######################
# Feature Extraction #
######################

LOGP_TAR= $(LST_TAR:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_WLD= $(LST_WLD:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_COH= $(LST_COH:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_IMP= $(LST_IMP:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_Z= $(LST_Z:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_SEG= $(LST_SEG:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_ANY= $(LST_ANY:lst/%=log/FeatureExtraction.%$(FEATTAG).log)

extractclient: $(LOGP_TAR)
extractwld: $(LOGP_WLD)
extractcoh: $(LOGP_COH)
extractimp: $(LOGP_IMP)
extractz: $(LOGP_Z)
extractseg: $(LOGP_SEG)
extractlst: $(LOGP_ANY)

log/FeatureExtraction.%$(FEATTAG).log: lst/%
      @echo `hostname` > $@
      perl bin/pl/feature_extract.pl $< $(FEATTAG)>> $@
      @echo `date` >> $@


###################
# Silence Removal #
###################

LOGE_TAR= $(LST_TAR:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_SEG= $(LST_SEG:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_WLD= $(LST_WLD:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_COH= $(LST_COH:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_IMP= $(LST_IMP:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_Z= $(LST_Z:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_ANY= $(LST_ANY:lst/%.lst=log/EnergyDetector.%$(FEATT).log)

# Computing labelFiles
energy: energyclient energyworld energyimp energyseg

energyclient:$(LOGE_TAR)
energywld:$(LOGE_WLD)
energycoh:$(LOGE_COH)
energyimp:$(LOGE_IMP)
energyseg:$(LOGE_SEG)
energylst:$(LOGE_ANY)
energyz:$(LOGE_Z)

log/EnergyDetector.%$(FEATTAG).n$(GAUSSNRG)m$(ALPHA).log: lst/%.lst
      @echo `hostname` > $@
      cat $< | xargs -i rm -f "$(LBLFP){}.lbl" >> $@
      $(BIN)EnergyDetector --config $(CFG)EnergyDetector.cfg --
featureServerMask $(ENERGYPOS) --loadFeatureFileExtension
$(FEATTAG).tmp.prm --thresholdMode meanStd --mixtureDistribCount
$(GAUSSNRG) --alpha=$(ALPHA) --labelFilesPath $(LBLFP) --
inputFeatureFilename $< >> $@
      @echo `date` >> $@

log/EnergyDetector.%$(FEATTAG).z$(GAUSSNRG)w$(ALPHA).log: lst/%.lst
      @echo `hostname` > $@
```

```
	cat $< | xargs -i rm -f "$(LBLFP){}.lbl" >> $@
	$(BIN)NormFeat --config $(CFG)NormFeat_energy.cfg --
loadFeatureFileVectSize $(ORGSIZE) --loadFeatureFileExtension
$(FEATTAG).tmp.prm --saveFeatureFileExtension $(FEATTAG).enr.tmp.prm -
-featureServerMask $(ENERGYPOS) --inputFeatureFilename $< >> $@
	$(BIN)EnergyDetector --config $(CFG)EnergyDetector.cfg --
alpha=$(ALPHA) --featureServerMask 0 --loadFeatureFileExtension
$(FEATTAG).enr.tmp.prm --thresholdMode weight --mixtureDistribCount
$(GAUSSNRG) --labelFilesPath $(LBLFP) --inputFeatureFilename $< >> $@
	@echo `date` >> $@


#########################
# Feature Normalisation #
#########################

LOGN_TAR= $(LST_TAR:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_WLD= $(LST_WLD:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_COH= $(LST_COH:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_IMP= $(LST_IMP:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_Z= $(LST_Z:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_SEG= $(LST_SEG:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_ANY= $(LST_ANY:lst/%.lst=log/NormFeat.%$(FEATT).log)

featnorm: featnormclient featnormwld featnormimp featnormseg

featnormclient: $(LOGN_TAR)
featnormwld: $(LOGN_WLD)
featnormcoh: $(LOGN_COH)
featnormimp: $(LOGN_IMP)
featnormseg: $(LOGN_SEG)
featnormlst: $(LOGN_ANY)
featnormz: $(LOGN_Z)
log/NormFeat.%$(FEATT).log: lst/%.lst
	@echo `hostname` > $@
	$(BIN)NormFeat --config $(CFG)NormFeat.cfg --vectSize $(ORGSIZE)
--loadFeatureFileExtension $(FEATTAG).tmp.prm --
saveFeatureFileExtension $(FEATEXT) --labelFilesPath $(LBLFP) --
writeAllFeatures false --inputFeatureFilename $< > $@
	@echo `date` >> $@



########################
# World Model Training #
########################

world: ./data/ubm/world_$(GDR)$(GMMEXT)
./data/ubm/world_$(GDR)$(GMMEXT):
	@echo `hostname` > log/WorldModelTraining.$(GDR)$(GMMEXT).log
	$(BIN)TrainWorld --config $(CFG)TrainWorld.cfg --
mixtureDistribCount $(NBGAUSS) --inputFeatureFilename
./lst/world_$(GDR)_$(WLDTYPE).lst --outputWorldFilename
ubm/world_$(GDR) --loadFeatureFileExtension $(FEATEXT) --
loadMixtureFileExtension $(GMMEXT) --saveMixtureFileExtension
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE) >>
log/WorldModelTraining.$(GDR)$(GMMEXT).log
	@echo `date` >> log/WorldModelTraining.$(GDR)$(GMMEXT).log

worldbg: ./data/ubm/world_$(GDR)bg$(GMMEXT)
./data/ubm/world_$(GDR)bg$(GMMEXT):
```

```
	@echo `hostname` > log/WorldModelTrainingbg.$(GDR)$(GMMEXT).log
	$(BIN)TrainWorld.exe --config $(CFG)TrainWorldbg.cfg --
mixtureDistribCount $(NBGAUSS) --inputFeatureFilename
./lst/world_$(GDR)_$(WLDTYPE).lst --outputWorldFilename
ubm/world_$(GDR) --loadFeatureFileExtension $(FEATEXT) --
loadMixtureFileExtension $(GMMEXT) --saveMixtureFileExtension
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE) --
inputWorldFilename xxx >> log/WorldModelTrainingbg.$(GDR)$(GMMEXT).log
	@echo `date` >> log/WorldModelTrainingbg.$(GDR)$(GMMEXT).log


######################################################
# Target Models Training and  TNorm Model Training #
######################################################

target: log/TrainGMMtarget.$(TRNTYPE).$(GDR)$(GMMEXT).log
log/TrainGMMtarget.$(TRNTYPE).$(GDR)$(GMMEXT).log:
	@echo `hostname` > $@
	$(PLDIR)linkfiles.pl $(TRLST) $(GMMEXT) >> $@
	$(TRAINTARGET) $(TRNF) >> $@
	@echo `date` >> $@

imp: log/TrainGMMimp.t$(TNT).$(GDR)$(GMMEXT).log
log/TrainGMMimp.t$(TNT).$(GDR)$(GMMEXT).log:
	@echo `hostname` >> $@
	$(TRAINTARGET) $(TNTRNF) >> $@
	@echo `date` >> $@


###########
# Testing #
###########

# Main Ndx - Target Model / Segment Test
target_seg: log/TestGMMtarget_seg.$(REST).log
log/TestGMMtarget_seg.$(REST).log:
	@echo `hostname` > $@
	$(COMPUTETEST) --ndxFilename ./lst/ndx/$(GDR).$(TYPE).ndx --
outputFilename ./res/target_seg.$(REST).res >> $@
	@echo `date` >> $@

post_nonorm: log/nonormEER$(DET).GMM.$(REST).log
log/nonormEER$(DET).GMM.$(REST).log:
	@echo `hostname` > $@
	$(SCORING) --inputFile ./res/target_seg.$(REST).res --outputFile
./res/GMM.$(REST).nist >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(REST).nist $(YEAR) $(TYPE)
$(DET) >> $@
	@echo `date` >> $@

imp_seg: log/TestGMMimp_seg.$(RESTNT).log
log/TestGMMimp_seg.$(RESTNT).log:
	@echo `hostname` > $@
	$(COMPUTETEST) $(TNNDXF) --outputFilename
./res/imp_seg.$(RESTNT).res >> $@
	@echo `date` >> $@

post_tnorm: log/tnormEER$(DET).GMM.$(RESTNT).log
log/tnormEER$(DET).GMM.$(RESTNT).log:
	@echo `hostname` > $@
```

```
	$(BIN)ComputeNorm --config $(CFG)ComputeNorm.cfg --tnormNistFile
./res/imp_seg.$(RESTNT).res --testNistFile
./res/target_seg.$(REST).res --normType tnorm --outputFileBaseName
./res/$(RESTNT).res --verbose false >> $@
	$(SCORING) --inputFile ./res/$(RESTNT).res.tnorm --outputFile
./res/GMM.$(RESTNT).nist >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(RESTNT).nist $(YEAR) $(TYPE)
$(DET) >> $@
	@echo `date` >> $@
plottn:
	$(PLDIR)plotDET.pl ./res/GMM.$(RESTNT).nist $(YEAR) $(TYPE)
$(DET)


gmmztnorm: gmm_znorm gmm_ztnorm post_ztnorm

gmm_znorm: log/TestGMMznorm.$(RESTNT).z$(ZNT).log
log/TestGMMznorm.$(RESTNT).z$(ZNT).log:
	@echo `hostname` > $@
	$(COMPUTETEST) --ndxFilename
./lst/ndx/Znorm.$(GDR).$(TRNTYPE).z$(ZNT).ndx --outputFilename
./res/znormGMM.$(RESTNT).z$(ZNT).res >> $@
	@echo `date` >> $@

gmm_ztnorm: log/TestGMMztnorm.$(RESTNT).z$(ZNT).log
log/TestGMMztnorm.$(RESTNT).z$(ZNT).log:
	@echo `hostname` > $@
	$(COMPUTETEST) --ndxFilename
./lst/ndx/ZTnorm.$(GDR).t$(TNT).z$(ZNT).ndx --outputFilename
./res/ztnormGMM.$(RESTNT).z$(ZNT).res >> $@
	@echo `date` >> $@

post_ztnorm: log/ztnormEER.GMM$(DET).$(RESTNT).z$(ZNT).log
log/ztnormEER.GMM$(DET).$(RESTNT).z$(ZNT).log:
	$(BIN)ComputeNorm --config $(CFG)ComputeNorm.cfg --znormNistFile
./res/znormGMM.$(RESTNT).z$(ZNT).res --ztnormNistFile
./res/ztnormGMM.$(RESTNT).z$(ZNT).res --tnormNistFile
./res/imp_seg.$(RESTNT).res --testNistFile
./res/target_seg.$(REST).res --normType ztnorm --outputFileBaseName
./res/$(RESTNT).z$(ZNT).res --verbose false >> $@
	$(BIN)ComputeNorm --config $(CFG)ComputeNorm.cfg --znormNistFile
./res/znormGMM.$(RESTNT).z$(ZNT).res --ztnormNistFile
./res/ztnormGMM.$(RESTNT).z$(ZNT).res --tnormNistFile
./res/imp_seg.$(RESTNT).res --testNistFile
./res/target_seg.$(REST).res --normType tznorm --outputFileBaseName
./res/$(RESTNT).z$(ZNT).res --verbose false >> $@
	$(SCORING) --inputFile ./res/$(RESTNT).z$(ZNT).res.ztnorm --
outputFile ./res/GMM.$(RESTNT).z$(ZNT).zt.nist >> $@
	$(SCORING) --inputFile ./res/$(RESTNT).z$(ZNT).res.tznorm --
outputFile ./res/GMM.$(RESTNT).z$(ZNT).tz.nist >> $@
	$(SCORING) --inputFile ./res/$(RESTNT).z$(ZNT).res.znorm --
outputFile ./res/GMM.$(REST).z$(ZNT).nist >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(RESTNT).z$(ZNT).zt.nist $(YEAR)
$(TYPE) $(DET) >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(RESTNT).z$(ZNT).tz.nist $(YEAR)
$(TYPE) $(DET) >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(REST).z$(ZNT).nist $(YEAR)
$(TYPE) $(DET) >> $@
	@echo `date` >> $"@
```

A continuación se mostrará el archivo Makefile que se ha utilizado en las tres restantes técnicas. Como se podrá comprobar, dicho código vuelve a contener parte del código que se ha utilizado en el anterior script para las simulaciones de GMM. Esto no debe resultar llamativo puesto que ya se ha comentado en varias ocasiones que todas las técnicas empleadas en este proyecto fin de carrera parten de GMM-UBM.

```
##########################################################################
#     NIST 2005-2006 Eval with Alize + GMMs + any SVMs + NAP + Factor
#     Analysis
#     ******************************
#     Usage: make -j<nbProc> <target> options
#     Default options in opt/file_options
#     Nicolas Scheffer
#     Benoit Fauve
#     David Bonomo Laynez
##########################################################################

#Default options
include opt/Makefile_options
CD=1s
FEAT=m50
M=sgmm

## other condition ? CD = 1s or 10s or...
include opt/cond$(CD)

## other type of feature? m50
include opt/FEAT_$(FEAT)

## SVM on other stuff?
include opt/SVMMODE_$(M)

## Octave or Matlab?
OM=matlab



########################
# Function definitions #
########################

SCORING=$(BIN)Scoring --mode NIST --threshold $(THRESHOLD) --
segTypeTest $(TST) --trainTypeTest $(TRN) --adaptationMode n

TRAINTARGET=$(BIN)TrainTarget --config $(CFG)TrainTarget.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(GMMEXT) --saveMixtureFileExtension $(GMMEXT) --featureServerMask
$(MASK) --vectSize $(SIZE) --loadFeatureFileExtension $(FEATEXT) --
MAPRegFactorMean $(RF)

MODELTOSV=$(BIN)/modelToSv --config $(CFG)modelToSV.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(MODEXT) --vectorFilesExtension $(SVEXT) --vectorFilesPath
$(DATAPATH) --mixtureFilesPath $(DATAPATH)

COMPUTETEST=$(BIN)ComputeTest --config $(CFG)target_seg.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
```

```
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE) --
loadFeatureFileExtension $(FEATEXT) --gender $(GDR)

SVMTRAIN=$(BIN)Svm --config cfg/SvmTrain.cfg --inputBckList
./lst/world_$(GDR)_$(COHORTTYPE).lst --vectorFilesExtension $(SVEXT) -
-vsize $(SVSIZE) --modelFilesExtension $(SVMEXT) --C $(PARAMC) --
bckFilesPath $(DATAPATH) --vectorFilesPath $(DATAPATH) --
modelFilesPath $(DATAPATH)
SVMTEST=$(BIN)Svm --config $(CFG)SvmTest.cfg --vectorFilesExtension
$(SVEXT) --modelFilesExtension $(SVMEXT) --gender $(GDR) --vsize
$(SVSIZE) --bckFilesPath $(DATAPATH) --vectorFilesPath $(DATAPATH) --
modelFilesPath $(DATAPATH)

COMPUTESTATS=$(BIN)ComputeJFAStats --config $(CFG)ComputeJFAStats.cfg
--inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE) --
loadFeatureFileExtension $(FEATEXT) --saveMatrixFilesExtension
.$(GDR)$(FAT1).mat

FATRAINTARGET=$(BIN)TrainTarget --config $(CFG)TrainTargetFA.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(GMMEXT) --saveMixtureFileExtension $(FAT).gmm --featureServerMask
$(MASK) --vectSize $(SIZE) --loadFeatureFileExtension $(FEATEXT) --
eigenChannelMatrix EC.$(GDR)$(FAT) --eigenChannelNumber $(KFA) --
numThread $(NT)

FACOMPUTETEST=$(BIN)ComputeTest --config $(CFG)target_segFA.cfg --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(FAT).gmm --featureServerMask $(MASK) --vectSize $(SIZE) --
loadFeatureFileExtension $(FEATEXT) --gender $(GDR) --
eigenChannelMatrix EC.$(GDR)$(FAT) --eigenChannelNumber $(KFA) --
numThread $(NT)



#########################
# Usage of the makefile #
#########################

example1:
      @echo "Display text for example 1"

example2: log/example2.log
log/example2.log:
      @echo "Display text for example 2"
      @echo "action from example2 are logged in log/example2.log"
      @echo "once run you have to remove log/example2.log to log it
again"
      echo example2> $@
all:
      @echo "*****************************************************"
      @echo "******      Launching a NIST 2005 - 06 Evaluation
      ******"
      @echo "*****************************************************"
      @echo "Usage: make <target> options"
      @echo "-------------------------------------------------------
------------------"
      @echo "There is a lot of targets, please read the makefile to
know them, here is some examples:"
      @echo "feature_extract, feature_norm, world, models, tests,
tnorm_tests, ztnorm, decision, ....."
```

```
	@echo ""
	@echo "Options:"
	@echo "make -j <number of targets at a time> -> useful for
multiprocessor machines"
	@echo "make -n -> prevent execution, just print the steps it
would do"
	@echo "*****************************************************"

doall: generatefeatall world2test

generatefeatallg: generatefeatall polyexpall

generatefeatall: generatefeatwldimp generatefeattest
generatefeatwldimp: generatefeatwld generatefeatimp
generatefeatwld: extractwld energywld featnormwld
generatefeatimp: extractimp energyimp featnormimp
generatefeatz: extractz energyz featnormz
generatefeatcoh: extractcoh energycoh featnormcoh
generatefeattest: generatefeatseg generatefeattarget
generatefeatseg: extractseg energyseg featnormseg
generatefeattarget: extractclient energyclient featnormclient
generatefeatlst: extractlst energylst featnormlst

extractall: extractwld extractclient extractseg extractimp

world2test: world target imp tests
train2test: target imp tests
tests: targetseg postnonorm impseg posttnorm
world2testnonorm: world target targetseg postnonorm

featsgmm: generatefeatall generatefeatcoh
sgmm: world target imp trainsgmm mod2svall svm
trainsgmm: traincohort traintest
svm: svmtarget svmimp testssvm
testssvm: svmtargetseg svmpostnonorm svmimpseg svmposttnorm
napall: generatefeatlst trainlst naptraining naptrans

fa: eigenchannel targetfa impfa testsfa
world2testfa: targetfa impfa testsfa
testsfa: targetsegfa postnonormfa impsegfa posttnormfa
fapost: postnonormfa posttnormfa fapostztnorm
faall: fa fatraincohort fatraintest
fan: eigenchannel targetfa impfa targetsegfa postnonormfa

total: doall sgmm generatefeatlst faall napall


jfa: jfamat targetjfa impjfa testsjfa
testsjfa: targetsegjfa postnonormjfa impsegjfa posttnormjfa

iv: tvstats tvmat targetiv impiv testsiv
testsiv: targetsegiv postnonormiv impsegiv posttnormiv



#####################################################################
# Make variables that include lst files depedending on their type #
#####################################################################

LST_TAR= $(wildcard lst/target_$(GDR).$(TRNTYPE).lst)
LST_WLD= $(wildcard lst/world_$(GDR)_$(WLDTYPE).lst)
```

```
LST_COH= $(wildcard lst/world_$(GDR)_$(COHORTTYPE).lst)
LST_IMP= $(wildcard lst/imp_$(GDR)_$(TNT).lst)
LST_SEG= $(wildcard lst/seg_$(GDR).$(TYPE).lst)
LST_ANY= $(wildcard lst/$(LISTNAME))
LST_Z= $(wildcard lst/znorm_$(GDR).$(ZNT).lst)

LSTS_TAR= $(wildcard target_$(GDR).$(TRNTYPE))
LSTS_WLD= $(wildcard world_$(GDR)_$(WLDTYPE))
LSTS_IMP= $(wildcard imp_$(GDR)_$(TNT))
LSTS_SEG= $(wildcard seg_$(GDR).$(TYPE))
LSTS_ANY= $(wildcard $(LISTNAME))


######################
# Feature Extraction #
######################

LOGP_TAR= $(LST_TAR:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_WLD= $(LST_WLD:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_COH= $(LST_COH:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_IMP= $(LST_IMP:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_Z= $(LST_Z:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_SEG= $(LST_SEG:lst/%=log/FeatureExtraction.%$(FEATTAG).log)
LOGP_ANY= $(LST_ANY:lst/%=log/FeatureExtraction.%$(FEATTAG).log)

extractclient: $(LOGP_TAR)
extractwld: $(LOGP_WLD)
extractcoh: $(LOGP_COH)
extractimp: $(LOGP_IMP)
extractz: $(LOGP_Z)
extractseg: $(LOGP_SEG)
extractlst: $(LOGP_ANY)

log/FeatureExtraction.%$(FEATTAG).log: lst/%
      @echo `hostname` at `date` > $@
      perl bin/pl/feature_extract.pl $< $(FEATTAG)>> $@
      @echo `date` >> $@


###################
# Silence Removal #
###################

LOGE_TAR= $(LST_TAR:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_SEG= $(LST_SEG:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_WLD= $(LST_WLD:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_COH= $(LST_COH:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_IMP= $(LST_IMP:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_Z  = $(LST_Z:lst/%.lst=log/EnergyDetector.%$(FEATT).log)
LOGE_ANY= $(LST_ANY:lst/%.lst=log/EnergyDetector.%$(FEATT).log)

energy: energyclient energyworld energyimp energyseg

energyclient:$(LOGE_TAR)
energywld:$(LOGE_WLD)
energycoh:$(LOGE_COH)
energyimp:$(LOGE_IMP)
energyseg:$(LOGE_SEG)
energylst:$(LOGE_ANY)
energyz:$(LOGE_Z)
```

```
## mean base frame selection works directly on the energy component
log/EnergyDetector.%$(FEATTAG).n$(GAUSSNRG)m$(ALPHA).log: lst/%.lst
	@echo `hostname` at `date` > $@
	cat $< | xargs -i rm -f "$(LBLFP){}.lbl" >> $@
	$(BIN)EnergyDetector --config $(CFG)EnergyDetector.cfg --
featureServerMask $(ENERGYPOS) --loadFeatureFileExtension
$(FEATTAG).tmp.prm --thresholdMode meanStd --mixtureDistribCount
$(GAUSSNRG) --alpha=$(ALPHA) --labelFilesPath $(LBLFP) --
inputFeatureFilename $< >> $@
	@echo `date` >> $@

## weight base frame selction works  on the normalised energy
component (NormFeat first)
log/EnergyDetector.%$(FEATTAG).z$(GAUSSNRG)w$(ALPHA).log: lst/%.lst
	@echo `hostname` at `date` > $@
	cat $< | xargs -i rm -f "$(LBLFP){}.lbl" >> $@
	$(BIN)NormFeat --config $(CFG)NormFeat_energy.cfg --
loadFeatureFileVectSize $(ORGSIZE) --loadFeatureFileExtension
$(FEATTAG).tmp.prm --saveFeatureFileExtension $(FEATTAG).enr.tmp.prm -
-featureServerMask $(ENERGYPOS) --inputFeatureFilename $< >> $@
	$(BIN)EnergyDetector --config $(CFG)EnergyDetector.cfg --
alpha=$(ALPHA) --featureServerMask 0 --loadFeatureFileExtension
$(FEATTAG).enr.tmp.prm --thresholdMode weight --mixtureDistribCount
$(GAUSSNRG) --labelFilesPath $(LBLFP) --inputFeatureFilename $< >> $@
	@echo `date` >> $@



#########################
# Feature Normalisation #
#########################

LOGN_TAR= $(LST_TAR:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_WLD= $(LST_WLD:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_COH= $(LST_COH:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_IMP= $(LST_IMP:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_Z= $(LST_Z:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_SEG= $(LST_SEG:lst/%.lst=log/NormFeat.%$(FEATT).log)
LOGN_ANY= $(LST_ANY:lst/%.lst=log/NormFeat.%$(FEATT).log)

featnorm: featnormclient featnormwld featnormimp featnormseg

featnormclient: $(LOGN_TAR)
featnormwld: $(LOGN_WLD)
featnormcoh: $(LOGN_COH)
featnormimp: $(LOGN_IMP)
featnormseg: $(LOGN_SEG)
featnormlst: $(LOGN_ANY)
featnormz: $(LOGN_Z)
log/NormFeat.%$(FEATT).log: lst/%.lst
	@echo `hostname` at `date` > $@
	$(BIN)NormFeat --config $(CFG)NormFeat.cfg --vectSize $(ORGSIZE)
--loadFeatureFileExtension $(FEATTAG).tmp.prm --
saveFeatureFileExtension $(FEATEXT) --labelFilesPath $(LBLFP) --
writeAllFeatures false --inputFeatureFilename $< > $@
	@echo `date` >> $@
```

```
########################
# World Model Training #
########################

world: ./data/ubm/world_$(GDR)$(GMMEXT)
./data/ubm/world_$(GDR)$(GMMEXT):
	@echo `hostname` at `date` >
log/WorldModelTraining.$(GDR)$(GMMEXT).log

	$(BIN)TrainWorld --config $(CFG)TrainWorld.cfg --
mixtureDistribCount $(NBGAUSS) --inputFeatureFilename
./lst/world_$(GDR)_$(WLDTYPE).lst --outputWorldFilename
ubm/world_$(GDR) --loadFeatureFileExtension $(FEATEXT) --
loadMixtureFileExtension $(GMMEXT) --saveMixtureFileExtension
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE) >>
log/WorldModelTraining.$(GDR)$(GMMEXT).log
	@echo `date` >> log/WorldModelTraining.$(GDR)$(GMMEXT).log


#####################################################
# Target Models Training and TNorm Models Training #
#####################################################

# Target Models Training
target: log/TrainGMMtarget.$(TRNTYPE).$(GDR)$(GMMEXT).log
log/TrainGMMtarget.$(TRNTYPE).$(GDR)$(GMMEXT).log:
	@echo `hostname` at `date` > $@
	$(PLDIR)linkfiles.pl $(TRLST) $(GMMEXT) >> $@
	$(TRAINTARGET) $(TRNF) >> $@
	@echo `date` >> $@

#TNorm Models Training
imp: log/TrainGMMimp.t$(TNT).$(GDR)$(GMMEXT).log
log/TrainGMMimp.t$(TNT).$(GDR)$(GMMEXT).log:
	@echo `hostname` at `date` >> $@
	$(TRAINTARGET) $(TNTRNF) >> $@
	@echo `date` >> $@


###########
# Testing #
###########

# Main Ndx - Target Model / Segment Test
targetseg: log/TestGMMtarget_seg.$(REST).log
log/TestGMMtarget_seg.$(REST).log:
	@echo `hostname` at `date` > $@
	$(COMPUTETEST) --ndxFilename ./lst/ndx/$(GDR).$(TYPE).ndx --
outputFilename ./res/target_seg.$(REST).res >> $@
	@echo `date` >> $@

### scoring (.res to .nist) and then tidy and split
postnonorm: log/nonormEER$(DET).GMM.$(REST).log
log/nonormEER$(DET).GMM.$(REST).log:
	@echo `hostname` at `date` > $@
	$(SCORING) --inputFile ./res/target_seg.$(REST).res --outputFile
./res/GMM.$(REST).nist >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(REST).nist $(YEAR) $(TYPE)
$(DET) >> $@
	@echo `date` >> $@
```

```
impseg: log/TestGMMimp_seg.$(RESTNT).log
log/TestGMMimp_seg.$(RESTNT).log:
	@echo `hostname` at `date` > $@
	$(COMPUTETEST) $(TNNDXF) --outputFilename
./res/imp_seg.$(RESTNT).res >> $@
	@echo `date` >> $@

posttnorm: log/tnormEER$(DET).GMM.$(RESTNT).log
log/tnormEER$(DET).GMM.$(RESTNT).log:
	@echo `hostname` at `date` > $@
	$(BIN)ComputeNorm --config $(CFG)ComputeNorm.cfg --tnormNistFile
./res/imp_seg.$(RESTNT).res --testNistFile
./res/target_seg.$(REST).res --normType tnorm --outputFileBaseName
./res/$(RESTNT).res --verbose false >> $@
	$(SCORING) --inputFile ./res/$(RESTNT).res.tnorm --outputFile
./res/GMM.$(RESTNT).nist >> $@
	$(PLDIR)postnist.pl ./res/GMM.$(RESTNT).nist $(YEAR) $(TYPE)
$(DET) >> $@
	@echo `date` >> $@



##############################################
# SuperGMM get extra models                  #
# train GMM to use with the SVM              #
# the segment files must be train (traintest)#
# trainlst (for NAP)                         #
##############################################


traintest: log/TrainGMMseg.$(TYPE).$(GDR)$(GMMEXT).log
log/TrainGMMseg.$(TYPE).$(GDR)$(GMMEXT).log:
	@echo `hostname` at `date` > $@
	$(TRAINTARGET) --targetIdList
./lst/trn/train_$(GDR).b$(TYPE).trn >> $@
	@echo `date` >> $@

traincohort: log/TrainGMMcohort.c$(COHORTTYPE).$(GDR)$(GMMEXT).log
log/TrainGMMcohort.c$(COHORTTYPE).$(GDR)$(GMMEXT).log:
	@echo `hostname` at `date` > $@
	$(TRAINTARGET) --targetIdList
./lst/trn/train_world_$(GDR)_$(COHORTTYPE).trn >> $@
	@echo `date` >> $@

trainznorm: log/TrainGMMznorm.z$(ZNT).$(GDR)$(GMMEXT).log
log/TrainGMMznorm.z$(ZNT).$(GDR)$(GMMEXT).log:
	@echo `hostname` at `date` > $@
	$(TRAINTARGET) --targetIdList
./lst/trn/train_znorm.$(GDR).$(ZNT).trn >> $@
	@echo `date` >> $@

trainlst: log/TrainGMMlist.$(LISTTRNNAME).$(GDR)$(GMMEXT).log
log/TrainGMMlist.$(LISTTRNNAME).$(GDR)$(GMMEXT).log:
	@echo `hostname` at `date` > $@
	$(TRAINTARGET) --targetIdList lst/$(LISTTRNNAME) >> $@
	@echo `date` >> $@
```

```
# model to SV - cohort target imp seg
# .gmm to .sv : extra the mean vectors

mod2sv: log/ModelToSV.$(GDR)$(MODEXT).log
log/ModelToSV.$(GDR)$(MODEXT).log:
	$(MODELTOSV) --inputFilename $(TRLST) >> $@
	$(MODELTOSV) --inputFilename $(TNLST) >> $@
	$(MODELTOSV) --inputFilename ./lst/trn/train_$(GDR).b$(TYPE).trn
>> $@
	$(MODELTOSV) --inputFilename
./lst/trn/train_world_$(GDR)_$(COHORTTYPE).trn >> $@
	$(MODELTOSV) --inputFilename
./lst/trn/train_znorm.$(GDR).$(ZNT).trn >> $@

mod2svtarget: log/ModelToSV.$(MODT).$(GDR).$(TRN)$(YEAR)$(MODEXT).log
log/ModelToSV.$(MODT).$(GDR).$(TRN)$(YEAR)$(MODEXT).log:
	$(MODELTOSV) --inputFilename $(TRLST)  >> $@
mod2svimp: log/ModelToSV.$(MODT).$(GDR).t$(TNT)$(MODEXT).log
log/ModelToSV.$(MODT).$(GDR).t$(TNT)$(MODEXT).log:
	$(MODELTOSV) --inputFilename $(TNLST)  >> $@
mod2svseg:log/ModelToSV.$(MODT).$(GDR).$(TYPE)$(MODEXT).log
log/ModelToSV.$(MODT).$(GDR).$(TYPE)$(MODEXT).log:
	$(MODELTOSV) --inputFilename ./lst/trn/train_$(GDR).b$(TYPE).trn
>> $@
mod2svcoh:log/ModelToSV.$(MODT).$(GDR).c$(COHORTTYPE)$(MODEXT).log
log/ModelToSV.$(MODT).$(GDR).c$(COHORTTYPE)$(MODEXT).log:
	$(MODELTOSV) --inputFilename
./lst/trn/train_world_$(GDR)_$(COHORTTYPE).trn  >> $@

mod2svz:log/ModelToSV.$(MODT).$(GDR).z$(ZNT)$(MODEXT).log
log/ModelToSV.$(MODT).$(GDR).z$(ZNT)$(MODEXT).log:
	$(MODELTOSV) --inputFilename
./lst/trn/train_znorm.$(GDR).$(ZNT).trn  >> $@

mod2svall: cpworld mod2svtarget mod2svimp mod2svseg mod2svcoh

cpworld: $(DATAPATH)/ubm/world_$(GDR)$(MODEXT)
$(DATAPATH)/ubm/world_$(GDR)$(MODEXT):
	cp $(DATAPATH)/ubm/world_$(GDR)$(GMMEXT)
$(DATAPATH)/ubm/world_$(GDR)$(MODEXT)



##################
#     SVM        #
#SVM train target#
##################

svmtarget: log/Train$(MODT)target.$(TRNTYPE).$(GDR)$(SVMEXT).log
log/Train$(MODT)target.$(TRNTYPE).$(GDR)$(SVMEXT).log:
	@echo `hostname` at `date` > $@
# need to change to train with more files
	$(PLDIR)linkfiles.pl $(TRLST) $(SVMEXT) >> $@
	$(SVMTRAIN) --inputFilename $(TRLST) >> $@
	@echo `date` >> $@
```

```
##################
# SVM train imp  #
##################

svmimp: log/Train$(MODT)imp$(MODT).t$(TNT).$(GDR)$(SVMEXT).log
log/Train$(MODT)imp$(MODT).t$(TNT).$(GDR)$(SVMEXT).log:
	@echo `hostname` at `date` > $@
	$(SVMTRAIN) --inputFilename $(TNLST) >> $@
	@echo `date` >> $@


############
# SVM test #
############

svmtargetseg: log/Testtarget_seg.$(SVMREST).log
log/Testtarget_seg.$(SVMREST).log:
	@echo `hostname` at `date` > $@
	$(SVMTEST) --inputFilename ./lst/ndx/$(GDR).b$(TYPE).ndx --
outputFilename ./res/target_seg.$(SVMREST).res >> $@
	@echo `date` >> $@

svmpostnonorm: log/nonormEER$(DET).$(SVMREST).log
log/nonormEER$(DET).$(SVMREST).log:
	@echo `hostname` at `date` > $@
	$(SCORING) --inputFile ./res/target_seg.$(SVMREST).res --
outputFile ./res/$(SVMREST).nist >> $@
	$(PLDIR)postnist.pl ./res/$(SVMREST).nist $(YEAR) $(TYPE) $(DET)
>> $@
	@echo `date` >> $@

svmimpseg: log/Testimp_seg.$(SVMREST).t$(TNT).log
log/Testimp_seg.$(SVMREST).t$(TNT).log:
	@echo `hostname` at `date` > $@
	$(SVMTEST) --outputFilename ./res/imp_seg.$(SVMREST).t$(TNT).res
--inputFilename ./lst/ndx/imp_rev_$(TNT)_$(GDR).$(TYPE).ndx>> $@
	@echo `date` >> $@

svmposttnorm: log/tnormEER$(DET).$(SVMREST).t$(TNT).log
log/tnormEER$(DET).$(SVMREST).t$(TNT).log:
	@echo `hostname` at `date` > $@
	$(BIN)ComputeNorm --config $(CFG)ComputeNorm.cfg --tnormNistFile
./res/imp_seg.$(SVMREST).t$(TNT).res --testNistFile
./res/target_seg.$(SVMREST).res --normType tnorm --outputFileBaseName
./res/$(SVMREST).t$(TNT).res --verbose false >> $@
	$(SCORING) --inputFile ./res/$(SVMREST).t$(TNT).res.tnorm --
outputFile ./res/$(SVMREST).t$(TNT).nist >> $@
	$(PLDIR)postnist.pl ./res/$(SVMREST).t$(TNT).nist $(YEAR)
$(TYPE) $(DET) >> $@
	@echo `date` >> $@


#################
# NAP training  #
#################

### nap matrix is built in data/NAPmat/
naptraining: log/NAPtraining.$(GDR)$(NAPT).log
log/NAPtraining.$(GDR)$(NAPT).log:
	@echo `hostname` at `date` > $@
```

```
	echo
"extention='$(GMMEXT)';gdr='$(GDR)';naplst='$(NAPLST)';k=$(K);lstname=
'./lst/$(GDR).$(NAPLST).ndx';" > $@.m
	cat bin/scripts_m/naptraining.m >> $@.m
	nohup $(OM) -nodesktop -nosplash -nodisplay < $@.m >> $@
	@echo `date` >> $@


######################
# NAP transformation #
######################

LOGNP_TAR= $(LST_TAR:lst/%.lst=log/NAPtransform.%$(NAPT).log)
LOGNP_WLD= $(LST_WLD:lst/%.lst=log/NAPtransform.%$(NAPT).log)
LOGNP_COH= $(LST_COH:lst/%.lst=log/NAPtransform.%$(NAPT).log)
LOGNP_IMP= $(LST_IMP:lst/%.lst=log/NAPtransform.%$(NAPT).log)
LOGNP_Z = $(LST_Z:lst/%.lst=log/NAPtransform.%$(NAPT).log)
LOGNP_SEG= $(LST_SEG:lst/%.lst=log/NAPtransform.%$(NAPT).log)
LOGNP_ANY= $(LST_ANY:lst/%.lst=log/NAPtransform.%$(NAPT).log)

#naptrans:linknap naptrans_coh naptrans_imp naptrans_seg
naptrans_client naptrans_z
naptrans:linknap naptrans_coh naptrans_imp naptrans_seg
naptrans_client naptrans_world

naptrans_wld: $(LOGNP_WLD)
naptrans_coh: $(LOGNP_COH)
naptrans_imp: $(LOGNP_IMP)
naptrans_z: $(LOGNP_Z)
naptrans_seg: $(LOGNP_SEG)
naptrans_client:$(LOGNP_TAR)
naptrans_any: $(LOGNP_ANY)

#could remove
linknap:
log/LinkNap.$(GDR).$(TRN)$(YEAR).k$(K).n$(NAPLST)$(GMMEXT).log
log/LinkNap.$(GDR).$(TRN)$(YEAR).k$(K).n$(NAPLST)$(GMMEXT).log:
	$(PLDIR)linkfiles.pl $(TRLST) .k$(K).n$(NAPLST)$(GMMEXT) > $@

log/NAPtransform.%$(NAPT).log: lst/%.lst
	@echo `hostname` at `date` > $@
	echo
"extention='$(GMMEXT)';gdr='$(GDR)';naplst='$(NAPLST)';k=$(K);lstname=
'$<';" > $@.m
	cat bin/scripts_m/naptransform.m >> $@.m
	nohup $(OM) -nodesktop -nosplash -nodisplay < $@.m >> $@
	@echo `date` >> $@

naptrans_world:log/NAPtransform.$(WLDTYPE)_world.$(GDR)$(NAPT).log
log/NAPtransform.$(WLDTYPE)_world.$(GDR)$(NAPT).log:
	@echo `hostname` at `date` > $@
	echo
"extention='$(GMMEXT)';gdr='$(GDR)';naplst='$(NAPLST)';k=$(K);lstname=
'lst/world_$(GDR).lst';" > $@.m
	cat bin/scripts_m/naptransform.m >> $@.m
	nohup $(OM) -nodesktop -nosplash -nodisplay < $@.m >> $@
	@echo `date` >> $@
```

```
######
# FA #
######

## calculate FA statistics
fastats: datafa/FAmat/FA_N.$(GDR)$(FAT1).mat
datafa/FAmat/FA_N.$(GDR)$(FAT1).mat:
	@echo `hostname` at `date` >> log/FAstats.$(GDR)$(FAT1).log
	@echo building $@ >> log/FAstats.$(GDR)$(FAT1).log
	$(COMPUTESTATS) --matrixFilesPath datafa/FAmat/FA_ --ndxFilename
lst/$(GDR).$(FALST).ndx >> log/FAstats.$(GDR)$(FAT1).log
	@echo `date` >> log/FAstats.$(GDR)$(FAT1).log

### cat stats
catfastats: datafa/FAmat/FA_N.$(GDR).l$(FALST3)$(GMMT).mat
datafa/FAmat/FA_N.$(GDR).l$(FALST3)$(GMMT).mat:
	@echo `hostname` at `date` >
log/FAcat.$(FALST1).$(FALST2).$(FALST3).$(GDR)$(GMMT).log
	echo
"ext1='.$(GDR).l$(FALST1)$(GMMT).mat';ext2='.$(GDR).l$(FALST2)$(GMMT).
mat';ext3='.$(GDR).l$(FALST3)$(GMMT).mat';pref='datafa/FAmat/FA_';" >
$@.m
	cat bin/scripts_m/catstats.m >> $@.m
	nohup $(OM) -nodesktop -nosplash -nodisplay < $@.m >>
log/FAcat.$(FALST1).$(FALST2).$(FALST3).$(GDR)$(GMMT).log
	@echo `date` >>
log/FAcat.$(FALST1).$(FALST2).$(FALST3).$(GDR)$(GMMT).log

### Train a channel matrix (check that fastats is done)
eigenchannel: fastats datafa/FAmat/EC.$(GDR)$(FAT).mat
datafa/FAmat/EC.$(GDR)$(FAT).mat:
	@echo `hostname` at `date` > log/EC.$(GDR)$(FAT).log
	@echo building $@ >> log/EC.$(GDR)$(FAT).log
	$(BIN)EigenChannel --config $(CFG)EigenChannel.cfg --
eigenChannelNumber $(KFA) --ndxFilename lst/$(GDR).$(FALST).ndx --
inputWorldFilename ubm/world_$(GDR) --loadMixtureFileExtension
$(GMMEXT) --featureServerMask $(MASK) --vectSize $(SIZE) --
loadFeatureFileExtension $(FEATEXT) --loadMatrixFilesExtension
.$(GDR)$(FAT1).mat --saveMatrixFilesExtension .$(GDR)$(FAT).mat --
numThread $(NT) >> log/EC.$(GDR)$(FAT).log
	@echo `date` >> log/EC.$(GDR)$(FAT).log



################################
# FA TRAINING: Target and TNorm #
################################

targetfa: log/TrainFAtarget.$(TRNTYPE).$(GDR)$(FAT).log
log/TrainFAtarget.$(TRNTYPE).$(GDR)$(FAT).log:
	@echo `hostname` at `date` > $@
	$(PLDIR)linkfiles.pl $(TRLST) $(FAT).gmm ./datafa/ >> $@
	$(FATRAINTARGET) $(TRNF)>> $@
	@echo `date` >> $@

impfa: log/TrainFAimp.t$(TNT).$(GDR)$(FAT).gmm.log
log/TrainFAimp.t$(TNT).$(GDR)$(FAT).gmm.log:
	@echo `hostname` at `date` >> $@
	$(FATRAINTARGET) $(TNTRNF) >> $@
	@echo `date` >> $@
```

```
##############
# FA Testing #
##############

# Main Ndx - Target Model / Segment Test
targetsegfa: log/TestFAtarget_seg$(FAREST).log
log/TestFAtarget_seg$(FAREST).log:
	@echo `hostname` at `date` > $@
	cp data/ubm/world_$(GDR)$(GMMEXT)
data/ubm/world_$(GDR)$(FAT).gmm
	$(FACOMPUTETEST) $(NDXF) --outputFilename $(FARESFILE) >> $@
	@echo `date` >> $@

postnonormfa: log/nonormEER$(DET).FA$(FAREST).log
log/nonormEER$(DET).FA$(FAREST).log:
	@echo `hostname` at `date` > $@
	$(SCORING) --inputFile $(FARESFILE) --outputFile
./res/FA$(FAREST).nist >> $@
	$(PLDIR)postnist.pl ./res/FA$(FAREST).nist $(YEAR) $(TYPE)
$(DET) >> $@
	@echo `date` >> $@

impsegfa: log/TestFAimp_seg$(FARESTNT).log
log/TestFAimp_seg$(FARESTNT).log:
	@echo `hostname` at `date` > $@
	cp data/ubm/world_$(GDR)$(GMMEXT)
data/ubm/world_$(GDR)$(FAT).gmm
	$(FACOMPUTETEST) $(TNNDXF) --outputFilename $(FATRESFILE) >> $@
	@echo `date` >> $@

posttnormfa: log/tnormEER$(DET).FA$(FARESTNT).log
log/tnormEER$(DET).FA$(FARESTNT).log:
	@echo `hostname` at `date` > $@
	$(BIN)ComputeNorm --config $(CFG)ComputeNorm.cfg --tnormNistFile
$(FATRESFILE) --testNistFile $(FARESFILE) --normType tnorm --
outputFileBaseName ./res/FA$(FARESTNT).res >> $@
	$(SCORING) --inputFile ./res/FA$(FARESTNT).res.tnorm --
outputFile ./res/FA$(FARESTNT).nist >> $@
	$(PLDIR)postnist.pl ./res/FA$(FARESTNT).nist $(YEAR) $(TYPE)
$(DET) >> $@
	@echo `date` >> $@



###############################
# FA SuperGMM get extra models #
###############################

fatraintest: log/TrainFAseg.$(TYPE).$(GDR)$(FAT).log
log/TrainFAseg.$(TYPE).$(GDR)$(FAT).log:
	@echo `hostname` at `date` > $@
	$(FATRAINTARGET) --targetIdList
./lst/trn/train_$(GDR).b$(TYPE).trn >> $@
	$(MODELTOSVFA) --inputFilename
./lst/trn/train_$(GDR).b$(TYPE).trn >> $@
	@echo `date` >> $@

fatraincohort: log/TrainFAcohort.c$(COHORTTYPE).$(GDR)$(FAT).log
log/TrainFAcohort.c$(COHORTTYPE).$(GDR)$(FAT).log:
	@echo `hostname` at `date` > $@
```

```
	$(FATRAINTARGET) --targetIdList
./lst/trn/train_world_$(GDR)_$(COHORTTYPE).trn >> $@
	$(MODELTOSVFA) --inputFilename
./lst/trn/train_world_$(GDR)_$(COHORTTYPE).trn >> $@
	@echo `date` >> $@

#########################################################################
```