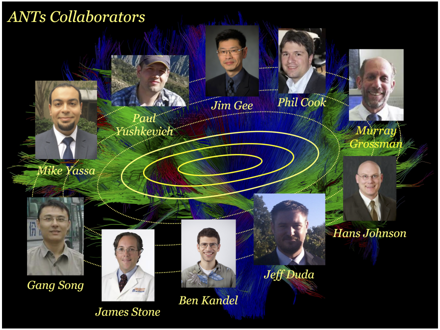
ASHS revisited — can we do better?

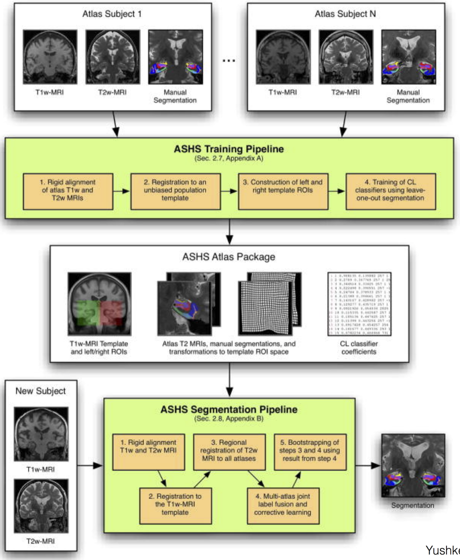
Nick Tustison and Mike Yassa

UCI



Motivation

ASHS overview



Training steps

1. atlas registration
2. joint label fusion
3. AdaBoost training

Segmentation steps

1. atlas registration
2. joint label fusion
3. corrective learning
4. repeat 1*, 2, 3
5. heuristics

Yushkevich et al., Human Brain Mapping, 36:258-287, 2015.

Joint Label Fusion

Modifications

- Registration
  - `ANTS` → `antsRegistration`
  - B-spline SyN ("`-t BSplineSyN[...]`")
  - `WarpImageMultiTransform` → `antsApplyTransforms`
  - generic label interpolation ("`-n GenericLabel[Linear]`")
- `jointfusion` → `antsJointFusion`
  - non-negative least squares option (vs. SVD)
  - multi-threaded
  - memory issues
  - joint intensity fusion

Registration — beyond original SyN

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 28 April 2014
doi: 10.3389/fninf.2014.00044

The Insight ToolKit image registration framework

Brian B. Avants[1]*, Nicholas J. Tustison[2], Michael Stauffer[1], Gang Song[1], Baohua Wu[1] and James C. Gee[1]

[1] *Penn Image Computing and Science Laboratory, Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA*
[2] *Department of Radiology and Medical Imaging, University of Virginia, Charlottesville, VA, USA*

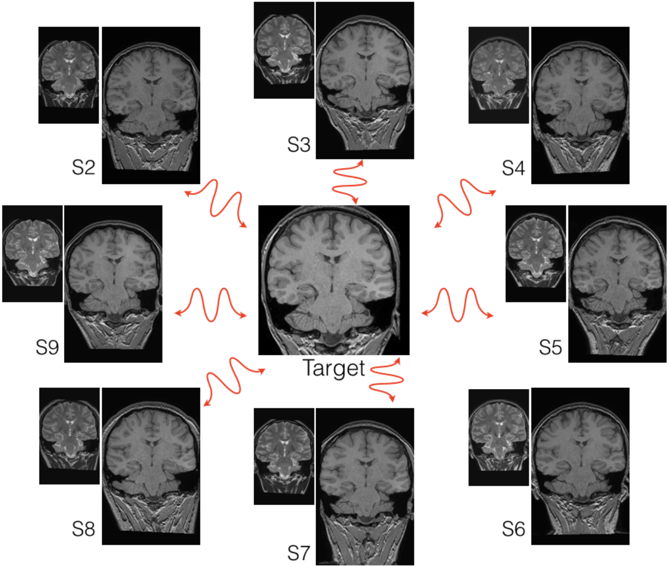
frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 23 December 2013
doi: 10.3389/fninf.2013.00039

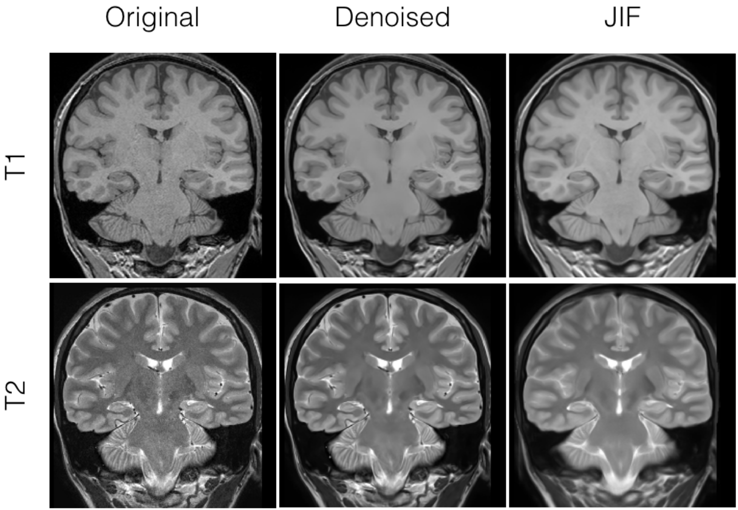
Explicit B-spline regularization in diffeomorphic image registration

Nicholas J. Tustison[1]* and Brian B. Avants[2]

T2 joint intensity fusion



T2 joint intensity fusion sample results



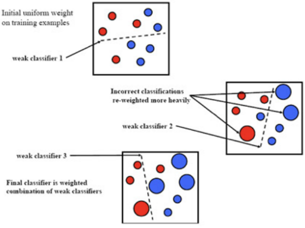
Corrective learning

Modifications

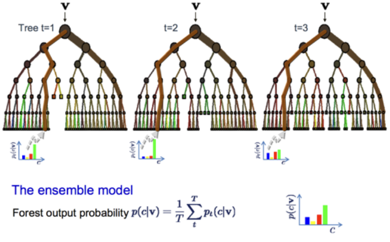
- machine learning technique
  - AdaBoost (original ASHS)
  - random forests
  - extreme gradient boosting
- ANTsR implementation
  - open-source
  - easy to change machine learning techniques
- prior knowledge
  - two classes (original ASHS)
  - four classes

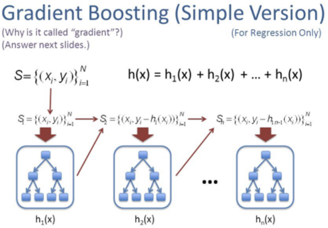
Machine learning techniques

- AdaBoost
- Random forests
- Extreme gradient boosting
- Support vector machines
- etc.



$$H(x) = sign(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$$



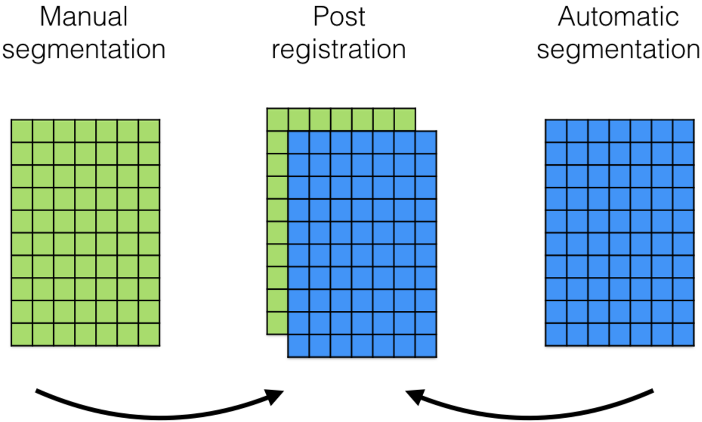


ANTsR facilitates technique substitution

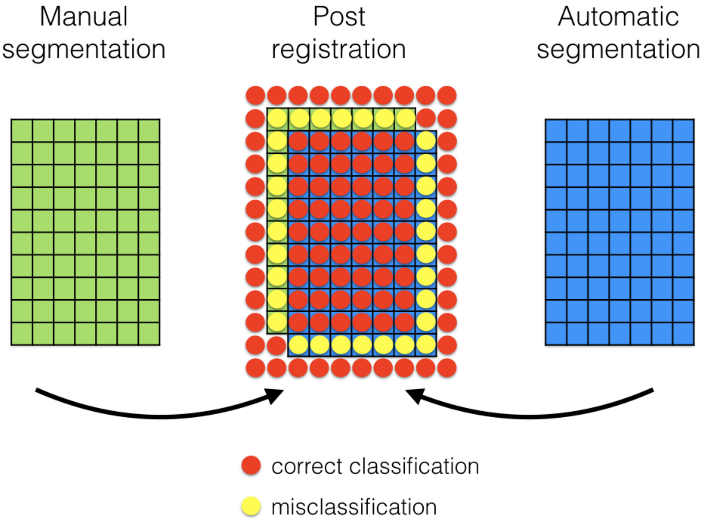
```r
#   ** xgboost modeling **

  modelData <- modelDataPerLabel
  modelData$Labels <- NULL
  modelData <- as.matrix( modelData )
  modelLabels <- as.character( modelDataPerLabel$Labels )

  modelDataPerLabelXgb <- xgb.DMatrix( modelData, label = modelLabels )

#      * xgboost tuning using cross validation
#
#  http://www.slideshare.net/odsc/owen-zhangopen-sourcetoolsanddscompetitions1 (slide 23)
#
#  xgb.cv.history <- xgb.cv( data = modelDataPerLabelXgb, nround = 500, nthread = 2,
#                            nfold = 5, metrics = list ( "merror" ), max.delspth = 3,
#                            eta = 0.3, objective = "multi:softprob", num_class = 4 )
#
  paramXgb <- list( max.depth = 6, eta = 0.3, silent = 0, objective = "multi:softprob", num_class = length( binaryLabelSet ) )
  modelXgb <- xgboost::xgb.train( paramXgb, modelDataPerLabelXgb, nrounds = 2, nthread = 2, verbose = 0 )

  labelModels[[l]] <- modelXgb
```

```r
#   ** randomForest modeling **
#
#      * randomForest tuning
#
#   capture.output( modelForestTuneRF <- randomForest::tuneRF(
#     modelDataPerLabel[, !( colnames( modelDataPerLabel ) == 'Labels' )], modelDataPerLabel$Labels,
#     plot = FALSE
#     ) )
#   minMtry <- modelForestTuneRF[which( modelForestTuneRF[,2] == min( modelForestTuneRF[,2] ) ), 1]
#   numberOfPredictors <- ncol( modelDataPerLabel[, !( colnames( modelDataPerLabel ) == 'Labels' )] )
#   message( "  mtry min = ", minMtry, " (number of total predictors = ", numberOfPredictors, ")\n", sep = "" )
#
#   modelFormula <- as.formula( "Labels ~ . " )
#   modelForest <- randomForest::randomForest( modelFormula, modelDataPerLabel,
#     ntree = 500, type = "classification", importance = TRUE, na.action = na.omit )
#
#   labelModels[[l]] <- modelForest
```

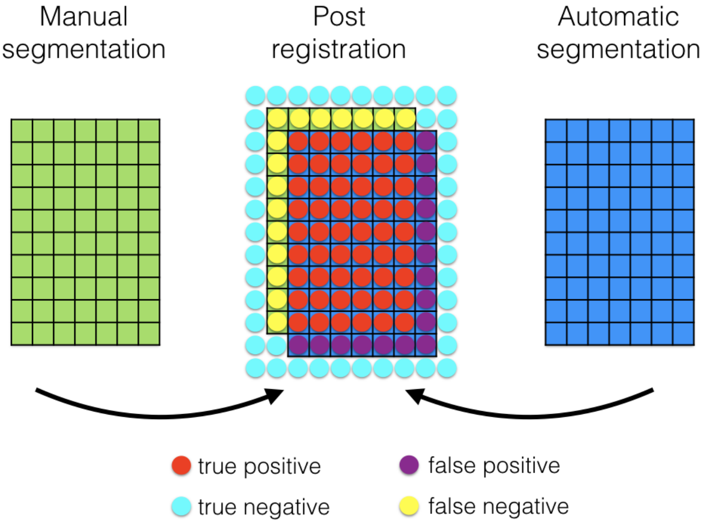
Incorporate additional prior knowledge



Two-class AdaBoost



Four-class random forest or extreme gradient boosting



Results

Summary

Paul's original ASHS (Table II, page 267)

ASHS Stage	Mean dice (left)	Mean dice (right)
Single atlas (average)	0.562	0.564

Penn Data

ASHS Stage	Mean dice (left)	Mean dice (right)
Single atlas (average)	0.587	0.570
JLF (SVD)	0.746	0.738
JLF (NNLS)	0.751	0.744
AdaBoost(NNLS) distanceDilate	0.775	0.763
RFmin (NNLS)	0.767	0.757
Xgbmin (NNLS)	0.777	0.765

UCI Data

ASHS Stage	Mean dice (left)	Mean dice (right)
Single atlas (average)	0.667	0.680
JLF (SVD)	0.783	0.780
JLF (NNLS)	0.776	0.779
AdaBoost(NNLS) distanceDilate	0.793	0.785
RFmin (NNLS)	0.795	0.786
Xgbmin (NNLS)	0.799	0.788

Data overview

- Penn data
  - Yushkevich et al., Hum Brain Mapp. 2015 Jan; 36(1): 258–287.
  - 29 subjects
  - 10 labels per hemisphere (2 are discarded prior to analysis)
  - hippocampal subfields and extrahippocampal cortical structures (ERC/PRC/PHC)
- UCI Data (“Stark Training Set”)
  - 19 subjects
  - 3 labels per hemisphere
  - ?

Penn Data

```
> aovLeft <- aov( Dice ~ Method + Label, allDataFrameLeft )
> summary( aovLeft )

              Df Sum Sq Mean Sq F value Pr(>F)
Method         3  0.357  0.1190   20.29 8.87e-13 ***
Label          8 12.809  1.6011  273.04  < 2e-16 ***
Residuals   1032  6.052  0.0059
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD( aovLeft, "Method" )

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Dice ~ Method + Label, data = allDataFrameLeft)

$Method
                      diff         lwr        upr     p adj
SVD-NNLS      -0.006890317 -0.02413941 0.01035878 0.7331132
AdaBoost-NNLS  0.030424709  0.01317561 0.04767380 0.0000374 <-
Xgb-NNLS       0.035627356  0.01837826 0.05287645 0.0000008 <-
AdaBoost-SVD   0.037315026  0.02006593 0.05456412 0.0000002 <-
Xgb-SVD        0.042517673  0.02526858 0.05976677 0.0000000 <-
Xgb-AdaBoost   0.005202648 -0.01204645 0.02245174 0.8652726
```

```
> aovRight <- aov( Dice ~ Method + Label, allDataFrameRight )
> summary( aovRight )

              Df Sum Sq Mean Sq F value Pr(>F)
Method         3  0.335  0.1117   17.93 2.37e-11 ***
Label          8  8.427  1.0533  169.10  < 2e-16 ***
Residuals   1032  6.428  0.0062
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD( aovRight, "Method" )

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Dice ~ Method + Label, data = allDataFrameRight)

$Method
                      diff         lwr        upr     p adj
SVD-NNLS      -0.006271697 -0.02404952 0.01150612 0.8007052
AdaBoost-NNLS  0.029442103  0.01166428 0.04721992 0.0001301 <-
Xgb-NNLS       0.034971774  0.01719395 0.05274959 0.0000029 <-
AdaBoost-SVD   0.035713801  0.01793598 0.05349162 0.0000017 <-
Xgb-SVD        0.041243471  0.02346565 0.05902129 0.0000000 <-
Xgb-AdaBoost   0.005529670 -0.01224815 0.02330749 0.8542476
```

UCI Data

```
> aovLeft <- aov( Dice ~ Method + Label, allDataFrameLeft )
> summary( aovLeft )

             Df Sum Sq Mean Sq F value Pr(>F)
Method        4 0.0142 0.00354   3.175 0.0139 *
Label         3 0.1113 0.03711  33.265 <2e-16 ***
Residuals   372 0.4150 0.00112
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD( aovLeft, "Method" )

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Dice ~ Method + Label, data = allDataFrameLeft)

$Method
                      diff           lwr        upr     p adj
SVD-NNLS      0.0001321447 -1.472049e-02 0.01498478 0.9999999
AdaBoost-NNLS 0.0102401711 -4.612460e-03 0.02509280 0.3245300
Xgb-NNLS      0.0149043158  5.168453e-05 0.02975695 0.0487253 <-
Rf-NNLS       0.0112067368 -3.645894e-03 0.02605937 0.2361193
AdaBoost-SVD  0.0101080263 -4.744605e-03 0.02496066 0.3378764
Xgb-SVD       0.0147721711 -8.046021e-05 0.02962480 0.0520404 <-
Rf-SVD        0.0110745921 -3.778039e-03 0.02592722 0.2472042
Xgb-AdaBoost  0.0046641447 -1.018849e-02 0.01951678 0.9109412
Rf-AdaBoost   0.0009665658 -1.388607e-02 0.01581920 0.9997728
Rf-Xgb       -0.0036975789 -1.855021e-02 0.01115505 0.9601853
```

```
> aovRight <- aov( Dice ~ Method + Label, allDataFrameRight )
> summary( aovRight )

             Df  Sum Sq Mean Sq F value Pr(>F)
Method        4 0.00458 0.00115   1.772  0.134
Label         3 0.14848 0.04949  76.595 <2e-16 ***
Residuals   372 0.24037 0.00065
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD( aovRight, "Method" )

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Dice ~ Method + Label, data = allDataFrameRight)

$Method
                      diff          lwr         upr     p adj
SVD-NNLS      0.0005491711 -0.010754534 0.011852877 0.9999289
AdaBoost-NNLS 0.0054772368 -0.005826469 0.016780942 0.6737533
Xgb-NNLS      0.0091173289 -0.002186377 0.020421034 0.1778540
Rf-NNLS       0.0060720658 -0.005231640 0.017375771 0.5810319
AdaBoost-SVD  0.0049280658 -0.006375640 0.016231771 0.7542446
Xgb-SVD       0.0085681579 -0.002735548 0.019871863 0.2318887
Rf-SVD        0.0055228947 -0.005780811 0.016826600 0.6667792
Xgb-AdaBoost  0.0036400921 -0.007663613 0.014943798 0.9031737
Rf-AdaBoost   0.0005948289 -0.010708877 0.011898534 0.9999023
Rf-Xgb       -0.0030452632 -0.014348969 0.008258442 0.9473355
```