

Walkthrough: Effect of TP on word learning

Dan Mirman

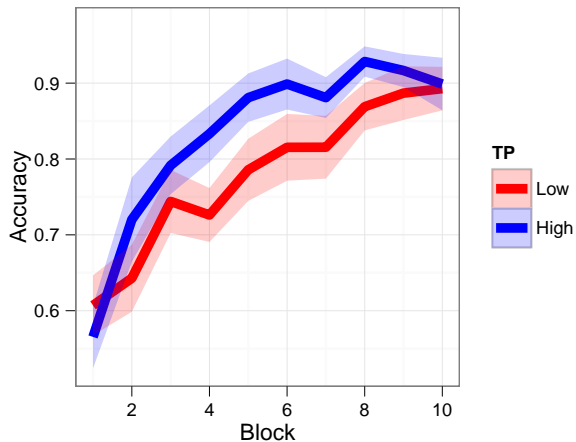
June 28, 2012

Load the data

```
> load("Examples.Rdata")  
> summary(WordLearnEx)
```

Subject	TP	Block	Accuracy
244 : 10	Low :280	Min. : 1.0	Min. :0.000
253 : 10	High:280	1st Qu.: 3.0	1st Qu.:0.667
302 : 10		Median : 5.5	Median :0.833
303 : 10		Mean : 5.5	Mean :0.805
305 : 10		3rd Qu.: 8.0	3rd Qu.:1.000
306 : 10		Max. :10.0	Max. :1.000
(Other):500			

Plot the data



Orthogonal polynomial time

Make a second order (quadratic) orthogonal polynomial in the range of Block

```
> t <- poly(unique(WordLearnEx$Block), 2)
```

Add it to data frame, aligned by Block value

```
> WordLearnEx[,paste("ot", 1:2, sep="")] <-  
+   t[WordLearnEx$Block, 1:2]
```

Fit the model

```
> library(lme4)  
> m.full <- lmer(Accuracy ~ (ot1+ot2)*TP + (ot1+ot2 | Subject),  
+   data=WordLearnEx, REML=F)
```

Examine parameter estimates

```
> print(m.full, corr=F)
```

```
Linear mixed model fit by maximum likelihood
```

```
Formula: Accuracy ~ (ot1 + ot2) * TP + (ot1 + ot2 | Subject)
```

```
Data: WordLearnEx
```

```
      AIC      BIC logLik deviance REMLdev  
-332.6 -276.4  179.3   -358.6   -327.4
```

```
Random effects:
```

Groups	Name	Variance	Std.Dev.	Corr
Subject	(Intercept)	0.0107622	0.103741	
	ot1	0.0154228	0.124188	-0.327
	ot2	0.0062798	0.079245	-0.280 -0.816
Residual		0.0245601	0.156717	

```
Number of obs: 560, groups: Subject, 56
```

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	0.778525	0.021727	35.83
ot1	0.286315	0.037788	7.58
ot2	-0.050849	0.033188	-1.53
TPHigh	0.052961	0.030727	1.72
ot1:TPHigh	0.001075	0.053441	0.02
ot2:TPHigh	-0.116455	0.046935	-2.48

p-values?!

Option 1: Model comparisons

```
> m.base <- lmer(Accuracy ~ (ot1+ot2) + (ot1+ot2 | Subject),
+               data=WordLearnEx, REML=F)
> m.0 <- lmer(Accuracy ~ (ot1+ot2) + TP + (ot1+ot2 | Subject),
+            data=WordLearnEx, REML=F)
> m.1 <- lmer(Accuracy ~ (ot1+ot2) + TP + ot1:TP + (ot1+ot2 | Subject),
+            data=WordLearnEx, REML=F)
```

p-values?!

Option 1: Model comparisons

```
> m.base <- lmer(Accuracy ~ (ot1+ot2) + (ot1+ot2 | Subject),
+               data=WordLearnEx, REML=F)
> m.0 <- lmer(Accuracy ~ (ot1+ot2) + TP + (ot1+ot2 | Subject),
+            data=WordLearnEx, REML=F)
> m.1 <- lmer(Accuracy ~ (ot1+ot2) + TP + ot1:TP + (ot1+ot2 | Subject),
+            data=WordLearnEx, REML=F)
> anova(m.base, m.0, m.1, m.full)
```

Data: WordLearnEx

Models:

```
m.base: Accuracy ~ (ot1 + ot2) + (ot1 + ot2 | Subject)
m.0: Accuracy ~ (ot1 + ot2) + TP + (ot1 + ot2 | Subject)
m.1: Accuracy ~ (ot1 + ot2) + TP + ot1:TP + (ot1 + ot2 | Subject)
m.full: Accuracy ~ (ot1 + ot2) * TP + (ot1 + ot2 | Subject)
```

	Df	AIC	BIC	logLik	Chisq	Chi	Df	Pr(>Chisq)
m.base	10	-330.77	-287.50	175.39				
m.0	11	-330.32	-282.72	176.16	1.5506		1	0.21304
m.1	12	-328.68	-276.75	176.34	0.3576		1	0.54984
m.full	13	-332.63	-276.37	179.32	5.9506		1	0.01471 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

p-values?!

Option 2: Normal approximation

```
> coefs <- data.frame(summary(m.full)@coefs)
> coefs$p <- 2*(1-pnorm(abs(coefs$t.value)))
> coefs[grep("*TP*",rownames(coefs),value=T),]
```

	Estimate	Std..Error	t.value	p
TPHigh	0.052960714	0.03072720	1.72357751	0.08478414
ot1:TPHigh	0.001075406	0.05344079	0.02012331	0.98394500
ot2:TPHigh	-0.116454843	0.04693532	-2.48117728	0.01309492

Plot the model fit

