

Identification, Inference, and Prediction

Adam M. Lauretig

12/18/2019

Welcome

- ▶ Talk: Parameter identification, what it means for inference and prediction
- ▶ Latent Variable Models: Factorization Machines, Interactive Fixed Effects
- ▶ Stan code + simulations!

Who I Am

- ▶ Data Scientist
- ▶ Previously, Ph.D. in political science (at THE Ohio State University)
- ▶ Word Embeddings, Causal Inference, Bayesian Statistics

Why We Model

- ▶ Descriptive Inference: is there an association between \mathbf{X} and \mathbf{y} ?
- ▶ Prediction: with \mathbf{X} and \mathbf{y} , what is \mathbf{y} for a new \mathbf{X} ?
- ▶ Causal Inference: does changing \mathbf{X} , change \mathbf{y} ?
- ▶ Shameless plug: Book Chapter **[insert link](#)**

Parameter Identification

- ▶ Unique solution to the model
- ▶ Necessary for causal inference
- ▶ Allows for uncertainty and interpretability

Latent Variable Models

- ▶ Learning parameters to reconstruct observed data
- ▶ Ex: Principal Components Analysis, Factor Analysis, Word2vec, etc
- ▶ Data $\mathbf{X}_{N \times J}$ is decomposed into two low rank matrices: $\boldsymbol{\gamma}_{N \times K}$ and $\boldsymbol{\delta}_{K \times J}$
- ▶ Various assumptions about the structure of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$

Factorization Machines/Interactive Fixed Effects

- ▶ Combine regression with a latent variable model on the residuals

But Why?

Factorization Machines/Interactive Fixed Effects

- ▶ Regression Model, for one observation
- ▶ Categorical predictors $\mathbf{x}_{n \in N}$, $\mathbf{x}_{j \in J}$
- ▶ Outcome \mathbf{y}
- ▶ Parameters β

$$y_{nj} = \mathbf{x}_n \beta_1 + \mathbf{x}_j \beta_2 + \varepsilon_{nj}$$

```
m1 <- lm(y ~ factor(group_1) + factor(group_2))
```

- ▶ With Interactions

$$y_{nj} = \mathbf{x}_n \beta_1 + \mathbf{x}_j \beta_2 + \mathbf{x}_n \times \mathbf{x}_j \beta_3 + \varepsilon_{nj}$$

```
m1 <- lm(y ~ factor(group_1) * factor(group_2))
```

Factorization Machines/Interactive Fixed Effects

Problems! - We can only estimate β_3 for observed interactions - As N and J grow, β_3 increases $N * J$

Factorization Machines/Interactive Fixed Effects

Solution!

- ▶ Replace β_3 with the dot product of low-rank latent factors:
- ▶ $\gamma_{N \times K}$
- ▶ $\delta_{J \times K}$
- ▶ β_3 is now $\gamma_n \cdot \delta_j^\top$

Factorization Machines/Interactive Fixed Effects

- ▶ Interaction model:

$$y_{nj} = \mathbf{x}_n \beta_1 + \mathbf{x}_j \beta_2 + \gamma_n \cdot \delta_j^\top + \varepsilon_{nj}$$

- ▶ Depending on our assumptions about $\gamma_n \cdot \delta_j^\top$, we can now create FMs or IFEs

Factorization Machines

- ▶ Each element of δ_j and γ_n is Normally distributed
- ▶ Automatic Relevance Determination (ARD) prior to shrink the matrix rank

$$y_{nj} \sim N(\mathbf{x}_n \beta_1 + \mathbf{x}_j \beta_2 + \gamma_n \cdot \delta_j^\top, 1)$$

$$\beta \sim N(0, \sigma^2)$$

$$\gamma_{n,k} \sim N(0, \psi_k)$$

$$\delta_{j,k} \sim N(0, 1)$$

$$\psi_k \sim \text{Gam}(a, b)$$

$$a \sim \text{Gam}(1, 1)$$

$$b \sim \text{Gam}(1, 1)$$

Simulate a Factorization Machine:

First, the regression component:

```
seed_to_use = 123
N = 1
J = 1
K = 1
set.seed(seed_to_use)
# number of levels for first covariate
N <- N
group_1 <- paste0("i", 1:N)
# number of levels for second covariate
J <- J
group_2 <- paste0("j", 1:J)
# number of latent dimensions
K <- K

# observed data ----
predictors <- expand.grid(group_1 = group_1, group_2 = group_2)
X_mat <- sparse.model.matrix(~ factor(group_1) + factor(group_2) - 1, data = predictors)

# for sparsity, since here, we're assuming we have only dummies
# creating numeric values for each individual FE
predictors_as_numeric <- cbind(
  as.numeric(factor(predictors[, 1])), as.numeric(factor(predictors[, 2])))

# the regression part of the equation
betas <- matrix(rnorm(n = ncol(X_mat), 0, 2))
linear_predictor <- X_mat %*% betas
```

Simulate a Factorization Machine:

Next, the latent factors:

```
# FM factors ----
# group_1 factors are gammas
# gamma_sd <- sort(rgamma(K, .1, .1), decreasing = TRUE)
a <- rgamma(1, shape = 2, rate = 2)
b <- rgamma(1, shape = 2, rate = 2)

gamma_sd <- sort(rgamma(n = K, shape = a, rate = b), decreasing = FALSE)
gammas <- mvrnorm(
  n = N, mu = rep(0, K), Sigma = gamma_sd * diag(K))

# group 2 factors are deltas
deltas <- mvrnorm(
  n = J, mu = rep(0, K), Sigma = diag(K))

factor_terms <- matrix(NA, nrow = nrow(linear_predictor), ncol = 1)

# multiply factors for each observation
for(i in 1:nrow(predictors)){
  g1 <- as.character(predictors[i, 1])
  g1 <- as.numeric(substr(g1, 2, nchar(g1)))

  g2 <- as.character(predictors[i, 2])
  g2 <- as.numeric(substr(g2, 2, nchar(g2)))

  factor_terms[i, ] <- matrix(
    gammas[g1, ], nrow = 1) %*%
    matrix(deltas[g2, ], ncol = 1)
}

y <- linear_predictor + factor_terms + rnorm(
  n = nrow(linear_predictor), 0, 1)
```