ALLOCATING FUNDS TO THE RIGHT PEOPLE

{ ADEM DUGALIC AND TRAM NGUYEN } {ADUGALIC, TRAM}@STANFORD.EDU CS229

PROBLEM

In many developing countries, government programs targeting the poor face the problem of identifying who they really are because income is usually unobserved for the very poor. A common approach is obtaining a list of poor households from the local administrative office, but the officers might have incentives to misreport the poorest households to some extent to benefit their own relatives and friends. If the government wishes to allocate funds to H poor households in the society, who should receive the funds?

We propose two machine learning algorithms to deal with the issue. Our better performing algorithm makes a significant improvements in identifying poor households in the population.

DATA

US household data extracted from the 2015 American Community Survey. We draw a sample of $50,000$ households from the data that are representative of the US population.

We treat 51 US states as different "villages". We construct the variable $z^{(i,k)}$ that indicates whether household i in village k lives below the income poverty threshold $\bar{y}$. The U.S. Census Bureau sets the household income poverty threshold in 2016 at $24,036$, so we choose $\bar{y} = 24,036$.

We simulate the dataset 200 times and assess the performance of our algorithms over the simulated datasets.

FEATURES

Household characteristics are: number of household members, the head's age and education, dwelling characteristics such as stove, fridge, television, etc., mortgage payment, food stamp eligibility and spending on gas, water and fuel.
We construct $z^{(i,k)}$ in the following way:

- First let $z^{(i,k)}$ equal to 1 if household's i in village k income is below the poverty line, and 0 otherwise.
- Then endow each village k with the probability of misreporting $\tau_k = \frac{\beta_k}{2}$, where β_k is randomly drawn from Beta(2,4).
- Finally, for each village k we randomly switch the values of $z^{(i,k)}$ at rate τ_k independently across households.

METHOD

Although we do not observe whether a household is poor, $s^{(i)} \equiv 1(y^{(i)} \leq \bar{y})$, we do observe imperfect signal of the variable of interest, $z^{(i)}$, which distinguishes our setup from standard unsupervised learning problems.

We apply two types of EM algorithms: discriminative EM and generative EM.

REFERENCES

[1] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. Wiley-Interscience, 2nd ed., 2002.

[2] L. N. R. D. Dempster, A.P., "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

MODELS

Discriminative EM

Assume that $p(s = 1) = \frac{1}{1+e^{-\theta' x}}$ for some parameter vector $\theta \in R^n$.
Given conditional distributional assumptions on s and z, the log likelihood is

$$l(\theta, \tau) = \sum_{k=1}^{K}\sum_{i=1}^{M_k} \log p\left(z^{(i,k)} | x^{(i,k)}; \theta, \tau_k\right)$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{M_k} \log \sum_{s=0}^{1} p\left(z^{(i,k)}, s | x^{(i,k)}; \theta, \tau_k\right)$$

The EM algorithm consists of initializing vector of parameters (θ, τ) and repeatedly carrying out the following two steps until convergence:

- (E-step) For each $k = 1, ..., K$ and $i = 1, ..., M_k$, set

$$Q_{i,k}(s) = p(s | z^{(i,k)}, x^{(i,k)}; \theta, \tau)$$

- (M-step) Set

$$(\theta, \tau) := \arg\max_{(\theta,\tau)} \sum_{k=1}^{K}\sum_{i=1}^{M_k}\sum_{s=0}^{1} Q_{i,k}(s) \log \frac{p\left(z^{(i,k)}, s | x^{(i,k)}; \theta, \tau_k\right)}{Q_{i,k}(s)}$$

which simplifies to

$$\tau_k = \frac{1}{M_k}\sum_{i=1}^{M_k}\left(Q_{i,k}(0) z^{(i,k)} + Q_{i,k}(1)(1 - z^{(i,k)})\right)$$

for $k = 1, ..., K$

$$\theta := \arg\max_{\theta} \sum_{k=1}^{K}\sum_{i=1}^{M_k} \log\left(\frac{e^{-\theta x^{(i,k)}} Q_{i,k}(0)}{1 + e^{-\theta' x^{(i,k)}}}\right)$$

where

$$p(s | z, x; \theta, \tau) = \frac{(1-\tau_k)^{1-|z-s|}\tau_k^{|z-s|} e^{-\theta' x(1-s)}}{(1-\tau_k)^{1-z}\tau_k^{z} e^{-\theta' x} + (1-\tau_k)^{z}\tau_k^{1-z}}$$

$$p(z, s | x; \theta, \tau_k) = (1-\tau_k)^{1-|z-s|}\tau_k^{|z-s|}\frac{e^{-\theta' x(1-s)}}{1 + e^{-\theta' x}}$$

Generative EM

Because some of our features are continuous and some discrete, we find it more appropriate to make Gaussian distributional assumptions on its principal components. We use the first four principal components because they capture almost all of the variation in the data and ease numerical computation. We refer to the transformed feature vector for individual i as $x^{(i)}$. We assume that $x^i | s \sim \mathcal{N}(\mu_s, \Sigma_s)$, and that $y \sim$ Bernoulli(ρ). The log likelihood is

$$l(\mu, \Sigma, \rho, \tau) = \sum_{k=1}^{K}\sum_{i=1}^{M_k} \log p\left(z^{(i,k)}, x^{(i,k)}; \mu, \Sigma, \rho, \tau_k\right)$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{M_k} \log \sum_{s=0}^{1} p\left(z^{(i,k)}, s, x^{(i,k)}; \mu, \Sigma, \rho, \tau_k\right)$$

The EM algorithm consists of initializing vector of parameters $(\mu, \Sigma, \rho, \tau)$ and repeatedly carrying out the following two steps until convergence:

- (E-step) For each $k = 1, ..., K$ and $i = 1, ..., M_k$, set

$$Q_{i,k}(s) = p(s | z^{(i,k)}, x^{(i,k)}; \mu, \Sigma, \rho, \tau)$$

- (M-step) Set

$$(\mu, \Sigma, \rho, \tau) := \arg\max_{(\mu,\Sigma,\rho,\tau)} \sum_{k=1}^{K}\sum_{i=1}^{M_k}\sum_{s=0}^{1} Q_{i,k}(s) \log \frac{p\left(z^{(i,k)}, s, x^{(i,k)}; \mu_s, \Sigma_s, \rho, \tau_k\right)}{Q_{i,k}(s)}$$

which simplifies to

$$\tau_k = \frac{1}{M_k}\sum_{i=1}^{M_k}\left(Q_{i,k}(0) z^{(i,k)} + Q_{i,k}(1)(1 - z^{(i,k)})\right)$$

$$\mu_s = \frac{\sum_{k=1}^{K}\sum_{i=1}^{M_k} Q_{i,k}(s) x^{(i,k)}}{\sum_{k=1}^{K}\sum_{i=1}^{M_k} Q_{i,k}(s)}$$

$$\Sigma_s = \frac{\sum_{k=1}^{K}\sum_{i=1}^{M_k} Q_{i,k}(s)(x^{(i,k)} - \mu_s)(x^{(i,k)} - \mu_s)^T}{\sum_{k=1}^{K}\sum_{i=1}^{M_k} Q_{i,k}(s)}$$

$$\rho = \frac{1}{M}\sum_{k=1}^{K}\sum_{i=1}^{M_k} Q_{i,k}(1) \quad \text{where}$$

$$p(s | z, x; \mu, \Sigma, \rho, \tau) = \frac{(1-\tau_k)^{1-|z-s|}\tau_k^{|z-s|}\phi\left(\Sigma_s^{-\frac{1}{2}}(x-\mu_s)\right)(1-\rho)^{1-s}\rho^{s}}{(1-\tau_k)^{1-z}\tau_k^{z}\phi\left(\Sigma_0^{-\frac{1}{2}}(x-\mu_0)\right)(1-\rho) + (1-\tau_k)^{z}\tau_k^{1-z}\phi\left(\Sigma_1^{-\frac{1}{2}}(x-\mu_1)\right)\rho}$$

$$p(z, s, x; \mu_s, \Sigma_s, \rho, \tau) = (1-\tau_k)^{1-|z-s|}\tau_k^{|z-s|}\phi\left(\Sigma_s^{-\frac{1}{2}}(x-\mu_s)\right)(1-\rho)^{1-s}\rho^{s}$$

Having obtained estimates β for each method, we compute $\hat{p}^{(i,k)} \equiv p(s^{(i,k)} = 1 | z^{(i,k)}, x^{(i,k)}; \beta)$ where $\beta = (\theta, \tau)$ for the discriminative EM and $\beta = (\mu, \Sigma, \rho, \tau)$ for the generative EM. We then for both methods find the households with the highest $H = 3000$ order statistics of the set $\{\hat{p}^{(i,k)} : 1 \leq i \leq M_k, 1 \leq k \leq K\}$, as well as identify households who are classified as poor (i.e. $\hat{p}^{(i,k)} > 0.5$).

FUTURE RESEARCH

The allocation may be improved even further by using a generative EM with different distributional assumptions over the features. The method used in this paper has a potential to improve classification in a wide variety of situations in which labels are imperfectly observed.

RESULTS

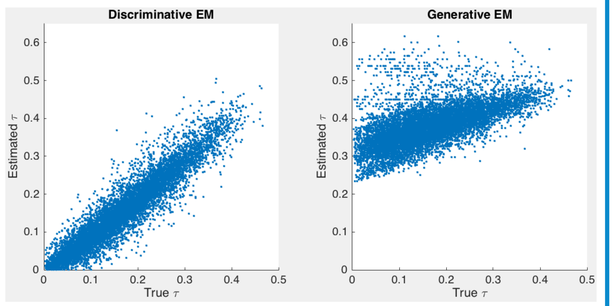

Figure 1: Estimated versus true misreporting rates

	MSE	Correlation
Discriminative EM	0.00022062	0.94214
Generative EM	0.010148	0.60304

Table 1: Mean square errors and correlation coefficients between estimated and true misreporting rates

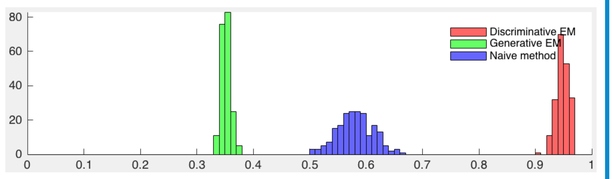

Figure 2: Fraction of poor households among selected 3000 households (200 simulations)

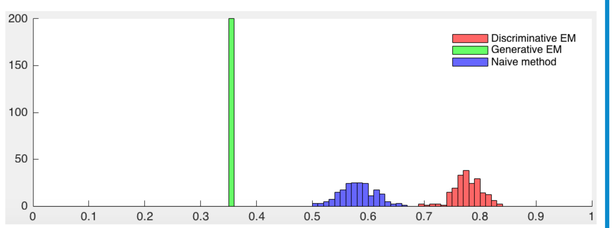

Figure 3: Fraction of poor households among those predicted to be poor (200 simulations)

DISCUSSION

- The naive method that ignores corruption allocates 58% of funds on average to poor households (Figure 2). The discriminative EM improves the allocation to about 95%. Thus, additional $37\% \times 3000 = 1110$ households who desperately need aid receive it if the discriminative EM is used to allocate aid.
- The generative EM does very poorly, substantially worsening the allocation of the naive method because the main underlying assumption that the features are normally distributed is no longer true for our data.