Modelling Weekly Sales Patterns

Pete Bunch

18th August 2015

Me

A bit about me

pictures of me doing things

A bit about me

pictures of some projects

Explain what I'm doing here.

Tesco Sales Modelling

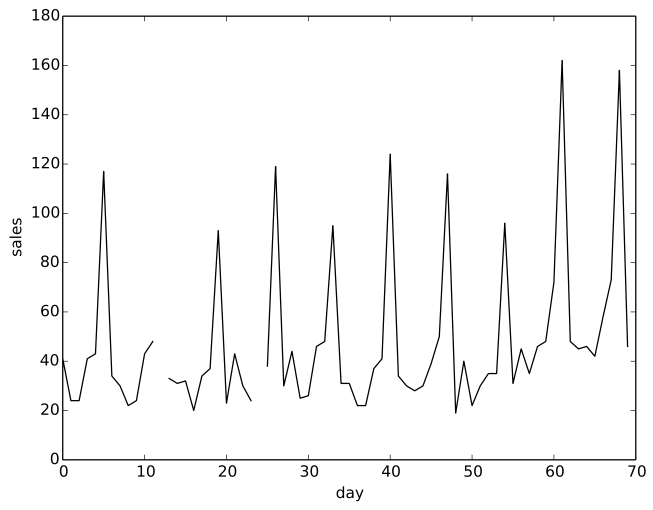
Tesco

- Largest supermarket chain in the UK (28% market share)
- Over 3,000 stores, operating in all regions
- Average of 10,000 products in each store
- Prediction for each product in each store, out to three weeks in advance
- Weather, promotions, events, reductions, seasonal patterns
- My focus: weekly patterns

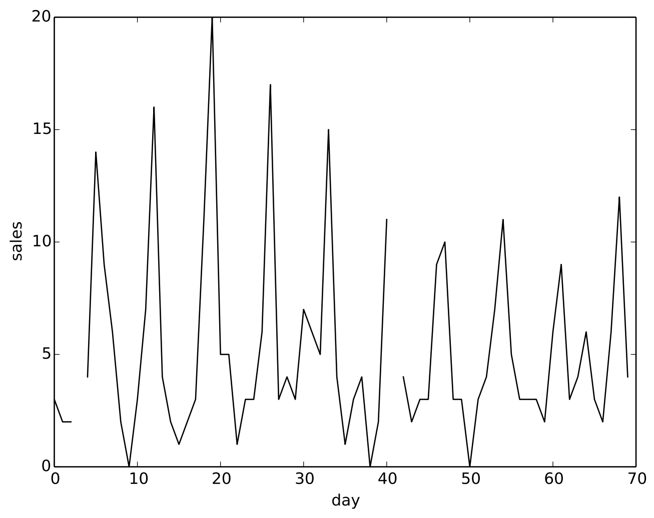
Why does it matter?

- Waste
- Stockholding
- Supplier interactions

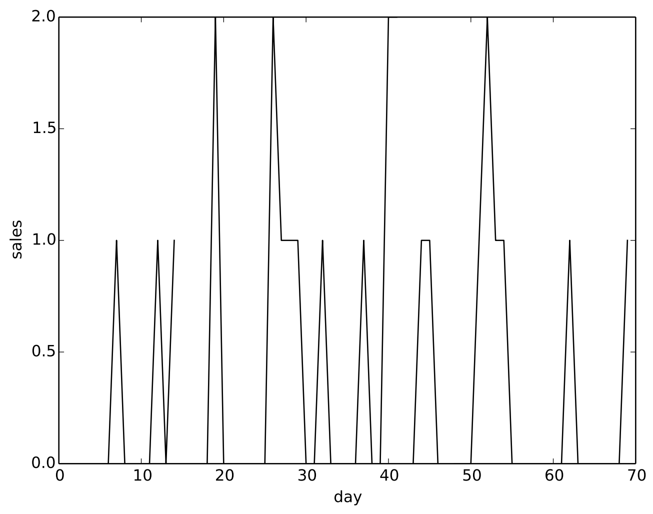
Sales data: a "fast" seller



Sales data: a “medium” seller



Sales data: a “slow” seller



Sales model: single-product, single-store

$$x_{ij} \underset{\text{i.i.d.}}{\sim} \mathcal{P}(u_i \cdot v_j) \tag{1}$$

x_{ij} sales on day i of week j

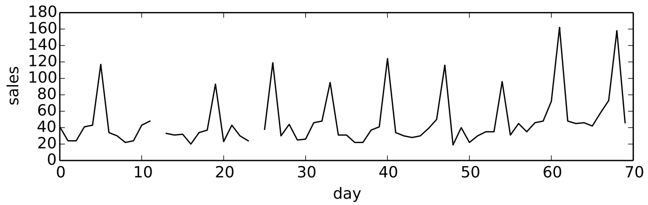
v_j expected sales in week j

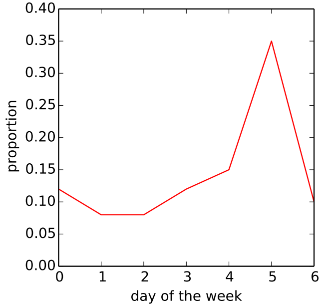
u_i proportion of sales on day i

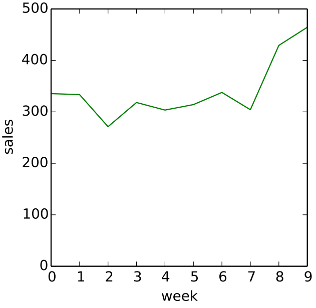
$$\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix} = \begin{bmatrix} \vdots \\ U \\ \vdots \end{bmatrix} \times \begin{bmatrix} \dots & V & \dots \end{bmatrix} \tag{2}$$

$$\sum_i u_i = 1 \tag{3}$$

Sales model: single-product, single-store







Fitting: maximum likelihood

Mask variables: $m_{ij} \in \{0, 1\}$

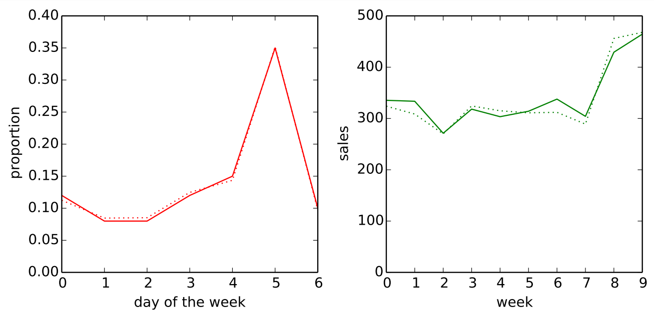
Log-likelihood:

$$\mathcal{L}(U, V) = \sum_{i,j} m_{i,j} \left[-u_i v_j + x_{ij} \log(u_i v_j) - \log(x_{ij}!) \right] \qquad (4)$$

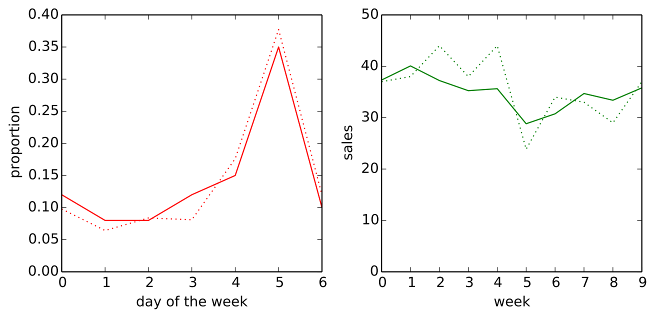
Maximize iteratively:

$$u_i \leftarrow \frac{\sum_j x_{ij} m_{ij}}{\sum_j v_j m_{ij}} \qquad v_j \leftarrow \frac{\sum_i x_{ij} m_{ij}}{\sum_j u_i m_{ij}} \qquad (5)$$

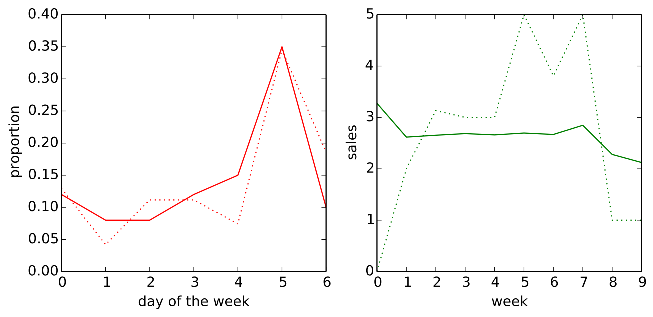
Fast seller training



Medium seller training



Slow seller training



Sales model: single-store, multiple products

$$x_{ij}^{(n)} \underset{\text{i.i.d.}}{\sim} \mathcal{P}\left(u_i^{(z_n)} \cdot v_j^{(n,z_n)}\right) \tag{6}$$

$x_{ij}^{(n)}$	sales of product n on day i of week j
$u_i^{(k)}$	proportion of sales on day i for products in cluster k
$v_j^{(n,k)}$	expected sales of product n in week j, estimated as if it were in cluster k
z_n	cluster indicator for product n

Fitting: maximum likelihood

Log-likelihood:

$$\mathcal{L}(U, V, Z) = \sum_{i,j,n} m_{i,j}^{(n)} \Big[- u_i^{(z_n)} v_j^{(n,z_n)} + x_{ij}^{(n)} \log(u_i^{(z_n)} v_j^{(n,z_n)}) - \log(x_{ij}^{(n)}!) \Big] \tag{7}$$

- V update unchanged
- U update uses all products in the cluster
- Z update by assigning each product to the cluster which maximises the likelihood

Tendency to get stuck in local maxima. Sensitive to starting conditions.

Better fitting: expectation maximisation

Marginal log-likelihood:

$$\mathcal{L}(U, V) = \log\left(\sum_Z p(X|U, V, Z)p(Z)\right) \tag{8}$$

- Lower bound the marginal log-likelihood
- Bound depends on $p(Z|X, U, V) = \prod_n p(z_n|X, U, V)$
- Soft clustering

Further details

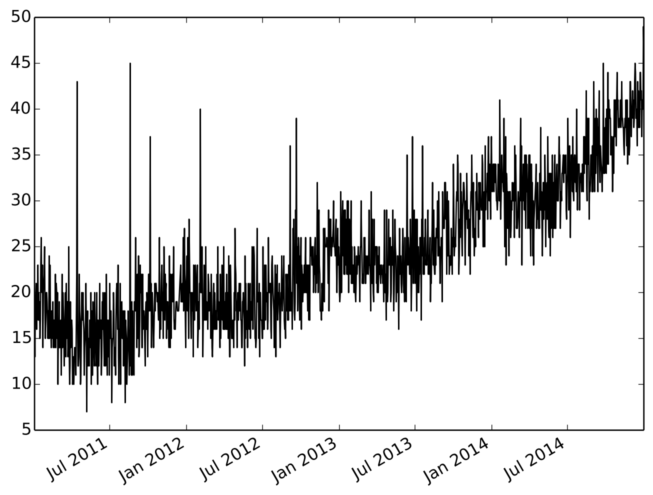
- Common clustering over multiple stores
- Probabilistic outlier detection
- Train on a rolling window every few weeks
- Implemented in SQL (with MATLAB)
- Sales are not Poisson distributed. Model over-dispersion.

Does it work?

- Yes
- 1–2% improvement in forecast accuracy metrics
- Estimated £(some millions) in fresh food waste per year

BlackLocus Sales Modelling Challege

Forecast sales for 2015



Another Poisson model

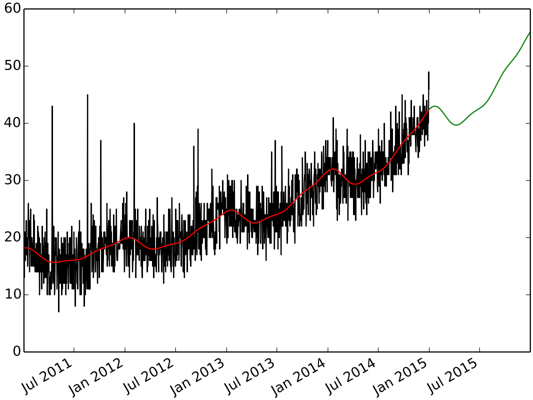
$$x_t \underset{\text{i.i.d.}}{\sim} \mathcal{P}(f_t \cdot g_t) \tag{9}$$

f_t Linear combination of polynomial basis functions

g_t Linear combination of Fourier series basis functions (yearly period)

- Maximise likelihood by gradient ascent (Newton's method)
- Choose model orders by cross-validation (train on 2011-2013, assess on 2014)
- Linear trend extrapolation

Forecast sales for 2015



If I had more time...

- Learn distribution over parameters and marginalise (MCMC)
- Model and remove outliers
- Check Poisson assumption