Clustering and Naive Bayes

Natural Language Processing

Sergey Nikolenko

Harbour Space University, Barcelona, Spain

January 9, 2018

GENERALIZING NAIVE BAYES

- Naive Bayes makes two additional simplifying assumptions:
    - we know class labels for all documents;
    - each document belongs to only one class (has only one topic).
- It turns out we can remove both of these constraints.
- First, what do we do if we don't know class labels, i.e., the dataset is unlabeled?
- What kind of a problem is it and how can we solve it?
- It's clustering, but we need to start with the EM algorithm...

PROBLEM SETTING

- Often the data has *latent* (missing) variables.
- We have the result of sampling a distribution, but some of the parameters are not known.
- We can treat latent variables as random values and look for the maximal likelihood hypothesis h, i.e., maximize

$$\mathrm{E}[p(D|h)] = \mathrm{E}[\int p(D, z|h)\mathrm{d}z]$$

for latent variables z.

SPECIAL CASE

- Example: consider a random variable x sampled from a mixture of two Gaussians with the same variance σ^2 and different means μ_1, μ_2.
- Two-stage sampling, but we don't know the first stage results.
- One point is a triple $\langle x_i, z_{i1}, z_{i2} \rangle$, where $z_{ij} = 1$ iff x_i was generated from distribution j, and we don't know z_{ij}.

EM ALGORITHM

- EM algorithm idea:
    - generate a hypothesis $h = (\mu_1, \mu_2)$;
    - while we have not reached local maximum:
        - compute the expectation $E(z_{ij})$ given the current hypothesis (E–step);
        - compute the new hypothesis $h' = (\mu'_1, \mu'_2)$ assuming that z_{ij} take values $E(z_{ij})$ computed before (M–step).

- For the Gaussians:

$$E(z_{ij}) = \frac{p(x = x_i|\mu = \mu_j)}{p(x = x_i|\mu = \mu_1) + p(x = x_i|\mu = \mu_2)} = \\ = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{e^{-\frac{1}{2\sigma^2}(x_i - \mu_1)^2} + e^{-\frac{1}{2\sigma^2}(x_i - \mu_2)^2}}.$$

- We compute the expectations and then tune the hypothesis:

$$\mu_j \leftarrow \frac{1}{m} \sum_{i=1}^{m} E(z_{ij}) x_i.$$

EM FORMALLY

- Formally, we are maximizing the likelihood with data $\mathcal{X} = \{x_1, \dots, x_N\}$.

$$L(\theta \mid \mathcal{X}) = p(\mathcal{X} \mid \theta) = \prod p(x_i \mid \theta)$$

  or, which is the same, maximizing $\ell(\theta \mid \mathcal{X}) = \log L(\theta \mid \mathcal{X})$.
- EM can help if this maximum is hard to find, but easy once we know something else...

EM FORMALLY

- Suppose that the data has *latent variables* such that the problem would be easy if we knew them.
- They don't necessarily have to correspond to anything interesting, maybe they are there just for convenience.
- In any case, we get a dataset $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ with joint density

$$p(z \mid \theta) = p(x, y \mid \theta) = p(y \mid x, \theta)p(x \mid \theta).$$

- Full likelihood $L(\theta \mid \mathcal{Z}) = p(\mathcal{X}, \mathcal{Y} \mid \theta)$ is a random variable since we don't know $\mathcal{Y}$.

EM FORMALLY

- Note that the real likelihood is $L(\theta) = E_Y\left[p(\mathcal{X}, \mathcal{Y} \mid \theta) \mid \mathcal{X}, \theta\right]$.
- E-step computes the conditional expectation of the (log) full likelihood given $\mathcal{X}$ and current estimates for parameters θ_n:

$$Q(\theta, \theta_n) = E\left[\log p(\mathcal{X}, \mathcal{Y} \mid \theta) \mid \mathcal{X}, \theta_n\right].$$

- Here θ_n are current estimates, θ are unknown values (which we want to get at the end); i.e., $Q(\theta, \theta_n)$ is a function of θ.

EM FORMALLY

- E-step computes the conditional expectation of the (log) full likelihood given $\mathcal{X}$ and current estimates for parameters θ:

$$Q(\theta, \theta_n) = E\left[\log p(\mathcal{X}, \mathcal{Y} \mid \theta) \mid \mathcal{X}, \theta_n\right].$$

- Conditional expectation:

$$E\left[\log p(\mathcal{X}, \mathcal{Y} \mid \theta) \mid \mathcal{X}, \theta_n\right] = \int_y \log p(\mathcal{X}, y \mid \theta) p(y \mid \mathcal{X}, \theta_n) \mathrm{d}y,$$

  where $p(y \mid \mathcal{X}, \theta_n)$ is the marginal distribution of latent variables.
- EM works best when it's easy to compute, maybe even analytically.
- Instead of $p(y \mid \mathcal{X}, \theta_n)$ we can substitute $p(y, \mathcal{X} \mid \theta_n) = p(y \mid \mathcal{X}, \theta_n) p(\mathcal{X} \mid \theta_n)$, it won't change anything.

EM FORMALLY

- As a result, after the E-step of the EM algorithm we get the function $Q(\theta, \theta_n)$.
- On the M-step, we maximize

$$\theta_{n+1} = \arg\max_\theta Q(\theta, \theta_n).$$

- And repeat until convergence.
- Actually, it suffices to find θ_{n+1} such that $Q(\theta_{n+1}, \theta_n) > Q(\theta_n, \theta_n)$ – Generalized EM.
- It remains to see what $Q(\theta, \theta_n)$ means and why it all works.

- We wanted to pass from θ_n to θ such that $\ell(\theta) > \ell(\theta_n)$.

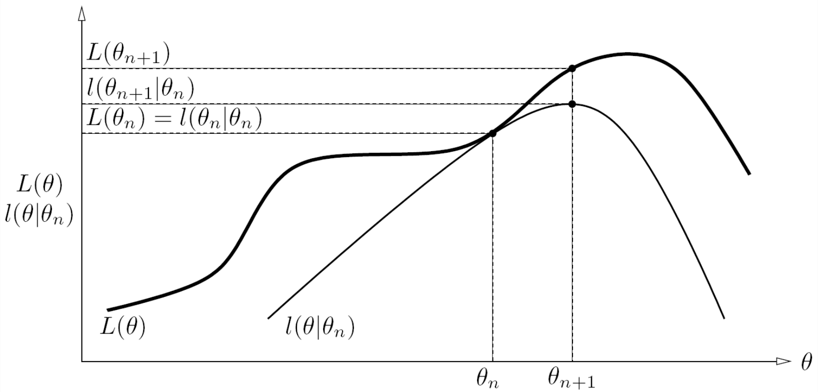
$$
\begin{aligned}
&\ell(\theta) - \ell(\theta_n) = \\
&\qquad = \log\left(\int_y p(\mathcal{X} \mid y, \theta) p(y \mid \theta) \mathrm{d}y\right) - \log p(\mathcal{X} \mid \theta_n) = \\
&= \log\left(\int_y p(y \mid \mathcal{X}, \theta_n) \frac{p(\mathcal{X} \mid y, \theta) p(y \mid \theta)}{p(y \mid \mathcal{X}, \theta_n)} \mathrm{d}y\right) - \log p(\mathcal{X} \mid \theta_n) \geq \\
&\geq \int_y p(y \mid \mathcal{X}, \theta_n) \log\left(\frac{p(\mathcal{X} \mid y, \theta) p(y \mid \theta)}{p(y \mid \mathcal{X}, \theta_n)}\right) \mathrm{d}y - \log p(\mathcal{X} \mid \theta_n) = \\
&\qquad = \int_y p(y \mid \mathcal{X}, \theta_n) \log\left(\frac{p(\mathcal{X} \mid y, \theta) p(y \mid \theta)}{p(\mathcal{X} \mid \theta_n) p(y \mid \mathcal{X}, \theta_n)}\right) \mathrm{d}y.
\end{aligned}
$$

- Thus, we get

$$\ell(\theta) \geq l(\theta, \theta_n) =$$
$$= \ell(\theta_n) + \int_y p(y \mid \mathcal{X}, \theta_n) \log \left(\frac{p(\mathcal{X} \mid y, \theta) p(y \mid \theta)}{p(\mathcal{X} \mid \theta_n) p(y \mid \mathcal{X}, \theta_n)} \right) \mathrm{d}y.$$

Упражнение. Prove that $l(\theta_n, \theta_n) = \ell(\theta_n)$.

- In other words, we have found a lower bound on $\ell(\theta)$ everywhere that touches it at point θ_n.
- I.e., we have found a lower bound for the likelihood and move to a point that maximizes it (or at least improves).
- This is called minorization-maximization (*MM*).


$L(\theta_{n+1})$
$l(\theta_{n+1}|\theta_n)$
$L(\theta_n) = l(\theta_n|\theta_n)$
$L(\theta)$
$l(\theta|\theta_n)$
$L(\theta)$
$l(\theta|\theta_n)$
θ
θ_n
θ_{n+1}

Justification of EM

- It remains to see that we can maximize Q.

$$\begin{aligned}\theta_{n+1} = \arg\max_\theta l(\theta, \theta_n) = \arg\max_\theta \{\ell(\theta_n) + \\ + \int_y f(y \mid \mathcal{X}, \theta_n) \log\left(\frac{p(\mathcal{X} \mid y, \theta) f(y \mid \theta)}{p(\mathcal{X} \mid \theta_n) f(y \mid \mathcal{X}, \theta_n)}\right) \mathrm{d}y\right\} = \\ = \arg\max_\theta \left\{\int_y p(y \mid \mathcal{X}, \theta_n) \log\left(p(\mathcal{X} \mid y, \theta) p(y \mid \theta)\right) \mathrm{d}y\right\} = \\ = \arg\max_\theta \left\{\int_y p(y \mid \mathcal{X}, \theta_n) \log p(\mathcal{X}, y \mid \theta) \mathrm{d}y\right\} = \\ = \arg\max_\theta \{Q(\theta, \theta_n)\},\end{aligned}$$

and the rest does not depend on θ.

EM AND CLUSTERING

IDEAS?

- How can we apply EM to clustering?

- Hypothesis: test examples are drawn independently from a mixture of cluster distributions

$$p(x) = \sum_{c \in C} w_c p_c(x), \quad \sum_{c \in C} w_c = 1,$$

where w_c is the probability to get a point from cluster c, p_c is the density of cluster c.

- What would be the form of p_c?

- What would be the form of p_c?
- Let's try... mmm... well, Gaussians. :)
- *Hypothesis* 2: each cluster c is a d–dimensional Gaussian distribution with mean $\mu_c = \{\mu_{c1}, \ldots, \mu_{cd}\}$ and diagonal matrix of covariances $\Sigma_c = \mathrm{diag}(\sigma^2_{c1}, \ldots, \sigma^2_{c2})$ (i.e., separate variance for every independent coordinate).

- Thus, we have formalized clustering as learning a mixture of distributions. That's where EM comes into play.
- Each test example looks like $(f_1(x), \dots, f_n(x))$.
- Latent variables in this case are probabilities g_{ic} of x_i to belong to cluster $c \in C$.

- E–step: by Bayes theorem, we compute latent variables g_{ic}:

- E–step: by Bayes theorem, we compute latent variables g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

IDEA

- E–step: by Bayes theorem, we compute latent variables g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–step: with g_{ic} we refine cluster parameters w, μ, σ:

- E–step: by Bayes theorem, we compute latent variables g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–step: with g_{ic} we refine cluster parameters w, μ, σ:

$$w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic},$$

IDEA

- E–step: by Bayes theorem, we compute latent variables g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–step: with g_{ic} we refine cluster parameters w, μ, σ:

$$w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic}, \quad \mu_{cj} = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} f_j(x_i),$$

- E–step: by Bayes theorem, we compute latent variables g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–step: with g_{ic} we refine cluster parameters w, μ, σ:

$$w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic}, \quad \mu_{cj} = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} f_j(x_i),$$

$$\sigma_{cj}^2 = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} \left(f_j(x_i) - \mu_{cj}\right)^2.$$

`EMCluster`$(X, |C|)$:

- Initialize $|C|$ clusters; initial approximation: $w_c := 1/|C|$, $\mu_c :=$ random x_i, $\sigma^2_{cj} := \frac{1}{n|C|}\sum_{i=1}^n \left(f_j(x_i) - \mu_{cj}\right)^2$.
- While cluster composition changes:
  - E–step: $g_{ic} := \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}$.
  - M–step: $w_c = \frac{1}{n}\sum_{i=1}^n g_{ic}$, $\mu_{cj} = \frac{1}{nw_c}\sum_{i=1}^n g_{ic} f_j(x_i)$,

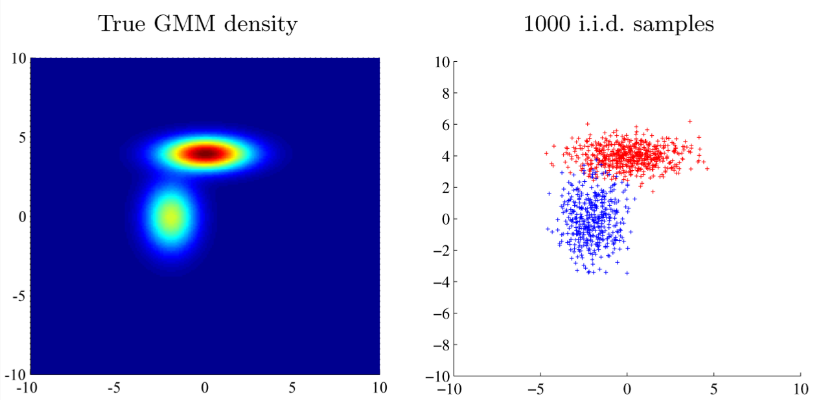
$$\sigma^2_{cj} = \frac{1}{nw_c}\sum_{i=1}^n g_{ic}\left(f_j(x_i) - \mu_{cj}\right)^2 .$$

  - Find which cluster x_i falls into:

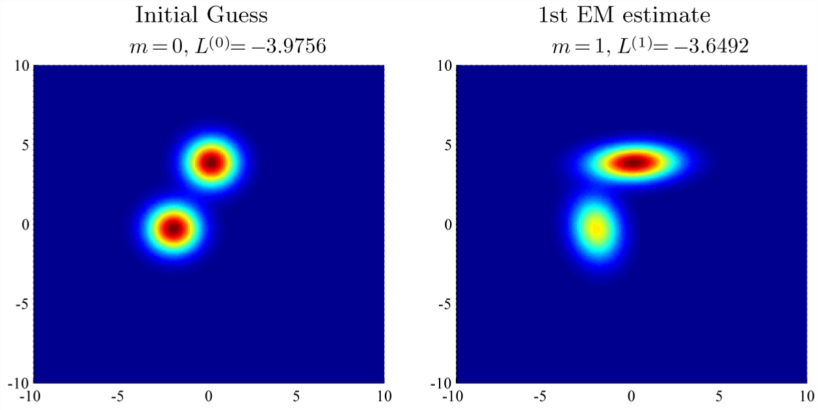
$$\mathrm{clust}_i := \arg\max_{c \in C} g_{ic}.$$

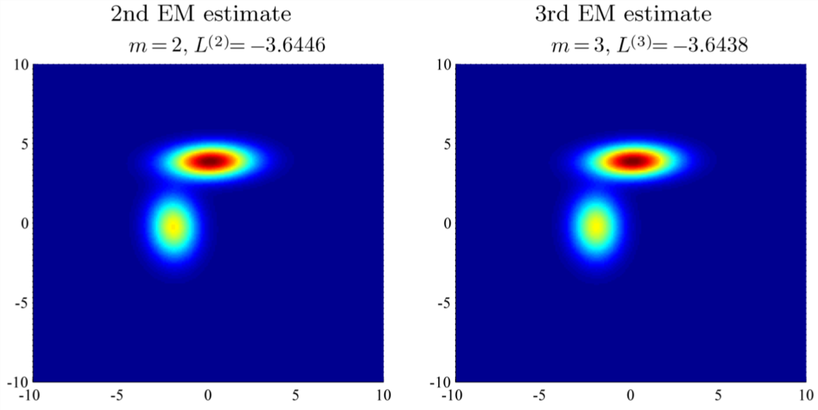
Упражнение. Prove that E-step and M-step indeed look like this.

EXAMPLE



True GMM density
10
5
0
-5
-10
-10
-5
0
5
10
1000 i.i.d. samples
10
8
6
4
2
0
−2
−4
−6
−8
−10
−10
−5
0
5
10


Initial Guess
$m = 0$, $L^{(0)} = -3.9756$
10
5
0
-5
-10
-10
-5
0
5
10
1st EM estimate
$m = 1$, $L^{(1)} = -3.6492$
10
5
0
-5
-10
-10
-5
0
5
10


2nd EM estimate
$m = 2$, $L^{(2)} = -3.6446$
3rd EM estimate
$m = 3$, $L^{(3)} = -3.6438$
10
5
0
-5
-10
-10
-5
0
5
10

PROBLEM

- We still need to specify the number of clusters.
- Possible solution: BIC.
- Other possible solution: non-parametric methods (out of our scope for now).

- k-means is a simplification of EM.
- Instead of computing probabilities of clusters, we use hard clustering.
- Besides, we cannot change the form of clusters in k–means (and that's not so bad).

OBJECTIVE

- Formally, k–means minimizes the error

$$E(X, C) = \sum_{i=1}^{n} ||x_i - \mu_i||^2,$$

  where μ_i is the cluster centroid nearest to x_i.
- I.e., we move centers and automatically relate points to nearest clusters.

NAIVE BAYES CLUSTERING

- So now we can come back to clustering with the naive Bayes assumptions.
- It is discrete, and it's best to solve directly with the EM algorithm (Expectation–Maximization):
  - on the E-step we compute expectations of which document belongs to which topic/class;
  - on the M-step, recompute probabilities $p(w \mid t)$ for fixed labels with naive Bayes.
- This is a simple generalization.
- Harder problem – generalize to multiple topics per document.

THANK YOU!

Thank you for your attention!