Advanced Machine Learning

Cristian Perez Jensen

November 4, 2024

Contents

List of symbols

$\doteq$	Equality by definition
$\approx$	Approximate equality
$\propto$	Proportional to
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$i : j$	Set of natural numbers between i and j. *I.e.*, $\{i, i+1, \dots, j\}$
$f : A \to B$	Function f that maps elements of set A to elements of set B
$\mathbb{1}\{\text{predicate}\}$	Indicator function (1 if predicate is true, otherwise 0)
$\boldsymbol{v} \in \mathbb{R}^n$	n-dimensional vector
$\boldsymbol{M} \in \mathbb{R}^{m \times n}$	$m \times n$ matrix
$\boldsymbol{M}^\top$	Transpose of matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	Inverse of matrix $\boldsymbol{M}$
$\det(\boldsymbol{M})$	Determinant of $\boldsymbol{M}$
$\frac{\mathrm{d}}{\mathrm{d}x} f(x)$	Ordinary derivative of $f(x)$ w.r.t. x at point $x \in \mathbb{R}$
$\frac{\partial}{\partial x} f(\boldsymbol{x})$	Partial derivative of $f(\boldsymbol{x})$ w.r.t. x at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^n$	Gradient of $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}^2_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^{n \times n}$	Hessian of $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$

1 Paradigms of data science

Let $\{x_1, \ldots, x_n\}$ be i.i.d. samples, generated by an unknown distribution P. Assume that this distribution is in a distribution family,

$$\mathcal{H} = \{p(\cdot \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}.$$

The goal is to learn the parameters $\boldsymbol{\theta}$ that fit the data $\{x_1, \ldots, x_n\}$ best.

Frequentism. In frequentism, the maximum likelihood estimator (MLE) parameters maximize the following,

$$\begin{aligned}\boldsymbol{\theta}^\star &\in \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \log p(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \mid \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta}).\end{aligned}$$

Bayesianism. Bayesianism assumes that there is a prior over distributions. The maximum a posteriori (MAP) parameters maximize the following,

$$\begin{aligned}\boldsymbol{\theta}^\star &\in \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \log p(\boldsymbol{\theta} \mid \boldsymbol{X}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \log p(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}) && \text{Bayes' rule.} \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \log p(\boldsymbol{\theta}) + \sum_{i=1}^{n} \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta}).\end{aligned}$$

In practice, the prior acts as a regularization term.

Statistical learning. Now, assume that we have labeled samples $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$, where y is the target variable. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. For a predictor function $f : \mathcal{X} \to \mathcal{Y}$, we define its risk as the expected loss,

$$\mathcal{R}(f) \doteq \mathbb{E}_{X,Y}[\ell(y, f(\boldsymbol{x}))].$$

In statistical learning, we want to find a function that minimizes the risk. However, since the distribution over X, Y is unknown, we cannot compute $\mathcal{R}(f)$ directly. Instead, we use the empirical risk as a surrogate,

$$\hat{\mathcal{R}}(f) \doteq \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)).$$

The goal is to obtain the empirical risk minimizer,

$$f^\star \in \operatorname*{argmin}_{f\in\mathcal{F}} \hat{\mathcal{R}}(f),$$

where $\mathcal{F}$ is a family of functions that we assume f belongs to.

2 *Anomaly detection*

In anomaly detection, we are given a sample of objects $\mathcal{X} \subseteq \mathbb{R}^d$ with a normal class $\mathcal{N} \in \mathcal{X}$—the data points that we wish to keep. We wish to construct a function $\phi : \mathcal{X} \to \{0, 1\}$, such that

$$\phi(\boldsymbol{x}) = 1 \iff \boldsymbol{x} \notin \mathcal{N}.$$

Formally, an anomaly is an unlikely event. Hence, the strategy is to fit a model of a parametric family of distributions to the data $\mathcal{X}$,

$$\mathcal{H} = \{p(\cdot \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}.$$

Then, we define the anomaly score of $\boldsymbol{x}$ as a low probability $p(\boldsymbol{x} \mid \boldsymbol{\theta}^\star)$ according to the optimal model in this hypothesis class.

Anomaly detection in a high-dimensional space is hard, because the normal class can be very complex. The idea is to project $\mathcal{X}$ down to a lower dimensionality and perform anomaly detection there—hopefully the projected version of the normal class $\Pi(\mathcal{N})$ is less complex. In order to find the optimal linear projection, we will use principal component analysis (PCA).

Furthermore, it has been observed that linear projections of high-dimensional distributions onto low-dimensional spaces resemble Gaussian distributions. Hence, after performing PCA, we will fit a Gaussian mixture model (GMM) to the projected data.

Principal component analysis. The goal of PCA is to linearly project $\mathbb{R}^d$ to $\mathbb{R}^{d^-}$ such that the maximum amount of variance of the data is preserved.[1] Consider the base case $d^- = 1$. Let $\boldsymbol{u} \in \mathbb{R}^d$ with $\|\boldsymbol{u}\| = 1$, we project onto $\boldsymbol{u}$ by inner product,

[1] Components with larger variance are more informative.

$$\boldsymbol{x} \mapsto \boldsymbol{u}^\top \boldsymbol{x}.$$

The sample mean of the reduced dataset is computed by

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{u}^\top \boldsymbol{x}_i = \boldsymbol{u}^\top \bar{\boldsymbol{x}},$$

where $\boldsymbol{x}$ is the sample mean of the original dataset. Further, the sample variance of the reduced dataset is

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{u}^\top \boldsymbol{x}_i - \boldsymbol{u}^\top \bar{\boldsymbol{x}}\right)^2 &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{u}^\top (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \boldsymbol{u} \\ &= \boldsymbol{u}^\top \left(\frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^\top\right) \boldsymbol{u} \\ &= \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u},\end{aligned}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the dataset. Since we want the projection that preserves the maximum variance, we have the following objective,

$$\boldsymbol{u}^\star \in \operatorname*{argmax}_{\|\boldsymbol{u}\|=1} \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u}.$$

The Lagrangian of this problem is

$$\mathcal{L}(\boldsymbol{u};\lambda) = \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} + \lambda\left(1 - \|\boldsymbol{u}\|^2\right)$$

with gradient

$$\frac{\partial \mathcal{L}(\boldsymbol{u};\lambda)}{\partial \boldsymbol{u}} = 2\boldsymbol{\Sigma}\boldsymbol{u} - 2\lambda\boldsymbol{u} \overset{!}{=} 0.$$

So, $\boldsymbol{u}$ must satisfy $\boldsymbol{\Sigma}\boldsymbol{u} = \lambda\boldsymbol{u}$—$\boldsymbol{u}$ is an eigenvector of $\boldsymbol{\Sigma}$. It is easy to see that this must be the principal eigenvector by rewriting the objective,

$$\begin{aligned}\boldsymbol{u}^\star &\in \operatorname*{argmax}_{\|\boldsymbol{u}\|=1} \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \\ &= \operatorname*{argmax}_{\substack{\|\boldsymbol{u}\|=1 \\ (\boldsymbol{u},\lambda)\in\operatorname{eig}(\boldsymbol{\Sigma})}} \lambda\|\boldsymbol{u}\|^2 \\ &= \operatorname*{argmax}_{\substack{\boldsymbol{u}\in\mathbb{R}^d \\ (\boldsymbol{u},\lambda)\in\operatorname{eig}(\boldsymbol{\Sigma})}} \lambda \\ &= \boldsymbol{u}_1.\end{aligned}$$

For $d^- > 1$, the remaining principal components can be computed with a similar idea. Iteratively, we factor out the previous principal components and do as above on the transformed dataset. For example, to get the second principal component, we first factor out the first principal component,

$$\mathcal{X}_1 \doteq \{\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{u}_1}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\} = \{\boldsymbol{x} - \boldsymbol{u}_1^\top \boldsymbol{x} \cdot \boldsymbol{u}_1 \mid \boldsymbol{x} \in \mathcal{X}\}.$$

Then, we do the same as above.

Gaussian mixture model. The probability density function (PDF) of a Gaussian mixture model with k components is formalized as a convex combination of Gaussians,

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{j=1}^{k} \pi_j \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j).$$

The parameters of this model are

$$\boldsymbol{\theta} = \{\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j \mid j \in [k]\},$$

where $\sum_{j=1}^{k} \pi_j = 1$ and $\{\boldsymbol{\Sigma}_j \mid j \in [k]\}$ are positive definite. We fit the parameters of this model by maximizing the log-likelihood,

$$\begin{aligned}\boldsymbol{\theta}^\star &\in \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \log p(\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\};\boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} \log p(\boldsymbol{x}_i;\boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j).\end{aligned}$$

```
1: Initialize $\boldsymbol{\theta}_0$
2: for $t \in [T]$ do
3:     $q^\star \in \operatorname{argmin}_q E(q, \boldsymbol{\theta}_{t-1})$
4:     $\boldsymbol{\theta}_t \in \operatorname{argmax}_{\boldsymbol{\theta}} M(q^\star, \boldsymbol{\theta})$
5: end for
6: return $\boldsymbol{\theta}_T$
```

Algorithm 1. The expectation-maximization algorithm, where

$$M(q, \boldsymbol{\theta}) \doteq \mathbb{E}_{z\sim q}\left[\log \frac{p(\boldsymbol{X}, z; \boldsymbol{\theta})}{q(z)}\right]$$
$$E(q, \boldsymbol{\theta}) \doteq \mathbb{E}_{z\sim q}\left[\log \frac{q(z)}{p(z \mid \boldsymbol{X}; \boldsymbol{\theta})}\right].$$

Note that the above is intractable, so we would like to decompose it into tractable terms that can be computed. Let's assume that we know from which latent variable each data point was generated, then we can compute the MLE of the extended dataset $\{(\boldsymbol{x}_i, z_i) \mid i \in [n]\}$ as

$$\begin{aligned}
\log p(\boldsymbol{X}, \boldsymbol{z}; \boldsymbol{\theta}) &= \sum_{i=1}^{n} \log p(\boldsymbol{x}_i, z_i) \\
&= \sum_{i=1}^{n} \log(\pi_{z_i} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})) \\
&= \sum_{i=1}^{n} \log \pi_{z_i} + \sum_{i=1}^{n} \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}),
\end{aligned}$$

which is tractable to maximize. Let q be a distribution over $[k]$, then we can rewrite the log-likelihood into tractable terms,

$$\begin{aligned}
\log p(\boldsymbol{X}; \boldsymbol{\theta}) &= \mathbb{E}_{z\sim q}[\log p(\boldsymbol{X}; \boldsymbol{\theta})] \\
&= \mathbb{E}_{z\sim q}\left[\log\left(\frac{p(\boldsymbol{X}, z; \boldsymbol{\theta})}{p(z \mid \boldsymbol{X}; \boldsymbol{\theta})}\right)\right] \\
&= \mathbb{E}_{z\sim q}\left[\log\left(\frac{p(\boldsymbol{X}, z; \boldsymbol{\theta})}{p(z \mid \boldsymbol{X}; \boldsymbol{\theta})} \frac{q(z)}{q(z)}\right)\right] \\
&= \underbrace{\mathbb{E}_{z\sim q}\left[\log \frac{p(\boldsymbol{X}, z; \boldsymbol{\theta})}{q(z)}\right]}_{\doteq M(q,\boldsymbol{\theta})} + \underbrace{\mathbb{E}_{z\sim q}\left[\log \frac{q(z)}{p(z \mid \boldsymbol{X}; \boldsymbol{\theta})}\right]}_{\doteq E(q,\boldsymbol{\theta})}.
\end{aligned}$$

These terms have the following two properties,

$$\begin{aligned}
&\log p(\boldsymbol{X}; \boldsymbol{\theta}) \geq M(q, \boldsymbol{\theta}), \quad \forall q, \boldsymbol{\theta} \\
&\log p(\boldsymbol{X}; \boldsymbol{\theta}) = M(q^\star, \boldsymbol{\theta}), \quad q^\star = p(\cdot \mid \boldsymbol{X}; \boldsymbol{\theta}), \quad \forall \boldsymbol{\theta}.
\end{aligned}$$

Hence, we can use $M(q^\star, \boldsymbol{\theta})$ as an approximation of $\log p(\boldsymbol{X}; \boldsymbol{\theta})$ around $\boldsymbol{\theta}$.

Theorem 2.1 (EM algorithm convergence)**.** Using the expectation-maximization algorithm, $\{\log p(\boldsymbol{x}; \boldsymbol{\theta}_t)\}_{t=0}^{T}$ is non-decreasing.

Proof. Given a data point $\boldsymbol{x}$ and current parameters $\boldsymbol{\theta}$, we have the following update,

$$\boldsymbol{\theta}' \in \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} M(q^\star, \boldsymbol{\theta}).$$

Hence, we have

$$\log p(\boldsymbol{x}) = M(q^\star, \boldsymbol{\theta}) \leq M(q^\star, \boldsymbol{\theta}') \leq \log p(\boldsymbol{x}; \boldsymbol{\theta}').$$

Thus, $\{\log p(\boldsymbol{x}; \boldsymbol{\theta}_t)\}_{t=0}^{T}$ is non-decreasing. ■

Summary. In conclusion, given a set of data points $\mathcal{X}$ with normal points $\mathcal{N} \subseteq \mathcal{X}$, we train an anomaly detector as follows,

1. Fit a projector $\pi : \mathbb{R}^d \to \mathbb{R}^{d^-}$ using PCA;
2. Fit a probability density function $p(\cdot \mid \boldsymbol{\theta})$ with k components to $\{\pi(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}$ using the EM algorithm;
3. For a point $\boldsymbol{x} \in \mathcal{X}$, its "anomaly score" is computed by $-\log p(\pi(\boldsymbol{x}); \boldsymbol{\theta})$.

References