Aalto University
School of Science

Zero-shot classifier based on robust ellipsoid optimization

Kasper Rantamäki

May 12, 2023

0 What is zero-shot learning?

- Zero-shot classifier = Classifier (machine learning model) capable of predicting labels it hasn't seen in training

0 What is zero-shot learning?

- Zero-shot classifier = Classifier (machine learning model) capable of predicting labels it hasn't seen in training
- Potentially useful when number of labels is very large or acquiring training data is very expensive

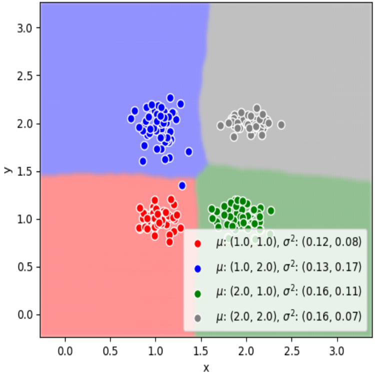
0 What is zero-shot learning?

- Zero-shot classifier = Classifier (machine learning model) capable of predicting labels it hasn't seen in training
- Potentially useful when number of labels is very large or acquiring training data is very expensive
- Used e.g. in neuroimaging studies of language, where the datapoints (grand averaged evoked responses to stimuli) are expensive to gather in large quantities and the number of labels (words or concepts) is vast

0 Why aren't most classifiers capable of zero-shot?

- Classifiers (generally) find a mapping:

$$\mathcal{F} : F^d \to L \qquad (f : X^d \to Y)$$

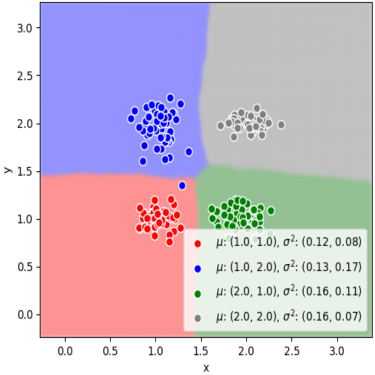


0 Why aren't most classifiers capable of zero-shot?

- Classifiers (generally) find a mapping:

$$\mathcal{F} : F^d \to L \qquad (f : X^d \to Y)$$

- i.e. divide the feature space into regions for each label with *decision boundaries*

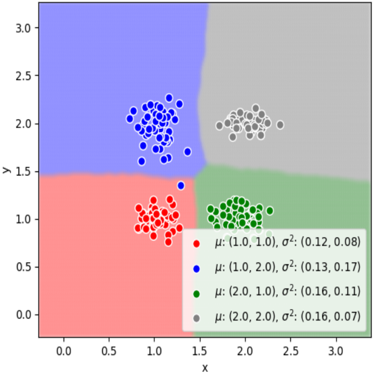


0 Why aren't most classifiers capable of zero-shot?

- Classifiers (generally) find a mapping:

$$\mathcal{F} : F^d \rightarrow L \qquad (f : X^d \rightarrow Y)$$

- i.e. divide the feature space into regions for each label with *decision boundaries*
- Not well suited for classifying labels not seen in training



0 Approach for zero-shot learning

- (Palatucci et al. 2009)[1] propose a *semantic output code classifier* as a composition of two different mapping functions:

$$
\begin{aligned}
\mathcal{H} &= (\mathcal{S} \circ \mathcal{L})(\cdot) \\
\mathcal{S} &: F^d \rightarrow S^p \\
\mathcal{L} &: S^p \rightarrow L
\end{aligned}
$$

[1] Palatucci, M., Pomerleau, D., Hinton, G. E., Mitchell, T. M. (2009). *Zero-shot learning with semantic output codes*. Advances in Neural Information Processing Systems

0 Approach for zero-shot learning

- (Palatucci et al. 2009)[1] propose a *semantic output code classifier* as a composition of two different mapping functions:

$$
\begin{aligned}
\mathcal{H} &= (\mathcal{S} \circ \mathcal{L})(\cdot) \\
\mathcal{S} &: F^d \rightarrow S^p \\
\mathcal{L} &: S^p \rightarrow L
\end{aligned}
$$

- S^p is a semantic space containing *one-hot encodings* of labels

[1] Palatucci, M., Pomerleau, D., Hinton, G. E., Mitchell, T. M. (2009). *Zero-shot learning with semantic output codes*. Advances in Neural Information Processing Systems

0 Approach for zero-shot learning

- (Palatucci et al. 2009)[1] propose a *semantic output code classifier* as a composition of two different mapping functions:

$$
\begin{aligned}
\mathcal{H} &= (\mathcal{S} \circ \mathcal{L})(\cdot) \\
\mathcal{S} &: F^d \to S^p \\
\mathcal{L} &: S^p \to L
\end{aligned}
$$

- S^p is a semantic space containing *one-hot encodings* of labels
- One-hot encodings make the mapping $\mathcal{L}$ trivial

[1] Palatucci, M., Pomerleau, D., Hinton, G. E., Mitchell, T. M. (2009). *Zero-shot learning with semantic output codes*. Advances in Neural Information Processing Systems

0 Approach for zero-shot learning

- (Palatucci et al. 2009)[1] propose a *semantic output code classifier* as a composition of two different mapping functions:

$$
\begin{aligned}
\mathcal{H} &= (\mathcal{S} \circ \mathcal{L})(\cdot) \\
\mathcal{S} &: F^d \to S^p \\
\mathcal{L} &: S^p \to L
\end{aligned}
$$

- S^p is a semantic space containing *one-hot encodings* of labels
- One-hot encodings make the mapping $\mathcal{L}$ trivial
- Real effort is in finding the mapping $\mathcal{S}$

[1] Palatucci, M., Pomerleau, D., Hinton, G. E., Mitchell, T. M. (2009). *Zero-shot learning with semantic output codes*. Advances in Neural Information Processing Systems

0 Outlined approach for finding mapping $\mathcal{S}$

- The mapping $\mathcal{S}$ can be considered as a conditional expectation:

$$\mathbb{E}[\mathbf{s}|\mathbf{f}] = \mathcal{S}(\mathbf{f}), \quad \text{where } \mathbf{s} \in S^p \text{ and } \mathbf{f} \in F^d$$

0 Outlined approach for finding mapping $\mathcal{S}$

- The mapping $\mathcal{S}$ can be considered as a conditional expectation:

$$\mathbb{E}[\mathbf{s}|\mathbf{f}] = \mathcal{S}(\mathbf{f}), \quad \text{where } \mathbf{s} \in S^p \text{ and } \mathbf{f} \in F^d$$

- Can be approximated as the weighted average

$$\frac{\sum_i K(d_i(\mathbf{f}))\mathbf{s}_i}{\sum_i K(d_i(\mathbf{f}))} = \mathbf{s}_{\text{approx}}$$

0 Outlined approach for finding mapping $\mathcal{S}$

- The mapping $\mathcal{S}$ can be considered as a conditional expectation:

$$\mathbb{E}[\mathbf{s}|\mathbf{f}] = \mathcal{S}(\mathbf{f}), \quad \text{where } \mathbf{s} \in S^p \text{ and } \mathbf{f} \in F^d$$

- Can be approximated as the weighted average

$$\frac{\sum_i K(d_i(\mathbf{f}))\mathbf{s}_i}{\sum_i K(d_i(\mathbf{f}))} = \mathbf{s}_{\text{approx}}$$

- Notice connection to kernel regression in statistics!

0 Outlined approach for finding mapping $\mathcal{S}$

- The mapping $\mathcal{S}$ can be considered as a conditional expectation:

$$\mathbb{E}[\mathbf{s}|\mathbf{f}] = \mathcal{S}(\mathbf{f}), \quad \text{where } \mathbf{s} \in S^p \text{ and } \mathbf{f} \in F^d$$

- Can be approximated as the weighted average

$$\frac{\sum_i K(d_i(\mathbf{f}))\mathbf{s}_i}{\sum_i K(d_i(\mathbf{f}))} = \mathbf{s}_{\text{approx}}$$

- Notice connection to kernel regression in statistics!
- Function K is the kernel function

0 Outlined approach for finding mapping $\mathcal{S}$

- The mapping $\mathcal{S}$ can be considered as a conditional expectation:

$$\mathbb{E}[\mathbf{s}|\mathbf{f}] = \mathcal{S}(\mathbf{f}), \quad \text{where } \mathbf{s} \in S^p \text{ and } \mathbf{f} \in F^d$$

- Can be approximated as the weighted average

$$\frac{\sum_i K(d_i(\mathbf{f}))\mathbf{s}_i}{\sum_i K(d_i(\mathbf{f}))} = \mathbf{s}_{\text{approx}}$$

- Notice connection to kernel regression in statistics!
- Function K is the kernel function
- Function d_i gives the distance from label i to point $\mathbf{f}$

0 Properties of the semantic space

- Good choice of semantic vectors maintain the semantic and syntactic relationships between the underlying concepts

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., (2013). *Distributed representations of words and phrases and their compositionality*. Advances in Neural Information Processing Systems

[3] Mikolov, T., Chen, K., Corrado, G., Dean, J., (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv:1301.3781

0 Properties of the semantic space

- Good choice of semantic vectors maintain the semantic and syntactic relationships between the underlying concepts
- In neuroimaging studies of language common choice of vectors are *word2vec* vectors

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., (2013). *Distributed representations of words and phrases and their compositionality*. Advances in Neural Information Processing Systems

[3] Mikolov, T., Chen, K., Corrado, G., Dean, J., (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv:1301.3781

0 Properties of the semantic space

- Good choice of semantic vectors maintain the semantic and syntactic relationships between the underlying concepts
- In neuroimaging studies of language common choice of vectors are *word2vec* vectors
- word2vec = Neural network based algorithm developed at Google that finds word embeddings based on large text corpus (Mikolov et al. 2013)[2][3]

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., (2013). *Distributed representations of words and phrases and their compositionality*. Advances in Neural Information Processing Systems

[3] Mikolov, T., Chen, K., Corrado, G., Dean, J., (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv:1301.3781

0 Properties of the semantic space

- Good choice of semantic vectors maintain the semantic and syntactic relationships between the underlying concepts
- In neuroimaging studies of language common choice of vectors are *word2vec* vectors
- word2vec = Neural network based algorithm developed at Google that finds word embeddings based on large text corpus (Mikolov et al. 2013)[2,3]
- Works very intuitively: ”brother” $-$ ”man” $+$ ”woman” $=$ ”sister”

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., (2013). *Distributed representations of words and phrases and their compositionality*. Advances in Neural Information Processing Systems

[3] Mikolov, T., Chen, K., Corrado, G., Dean, J., (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv:1301.3781

0 Probabilistic interpretation

- Let X be a set of datapoints $\mathbf{x} \in \mathbb{R}^d$ with label l_x

0 Probabilistic interpretation

- Let X be a set of datapoints $\mathbf{x} \in \mathbb{R}^d$ with label l_x
- We model the probability distribution of points $\mathbf{x} \in X$ with random vector $\mathbf{X} = [X_1 \; ... \; X_d]^\top$, where X_i is random variable for feature i

0 Probabilistic interpretation

- Let X be a set of datapoints $\mathbf{x} \in \mathbb{R}^d$ with label l_x
- We model the probability distribution of points $\mathbf{x} \in X$ with random vector $\mathbf{X} = [X_1 \ ... \ X_d]^\top$, where X_i is random variable for feature i
- Assume that $\mathbf{X}$ follows Gaussian distribution or something very close to it

0 Probabilistic interpretation

- Let X be a set of datapoints $\mathbf{x} \in \mathbb{R}^d$ with label l_x
- We model the probability distribution of points $\mathbf{x} \in X$ with random vector $\mathbf{X} = [X_1 \ldots X_d]^\top$, where X_i is random variable for feature i
- Assume that $\mathbf{X}$ follows Gaussian distribution or something very close to it
- Then $\mathbf{X}$ is defined by mean $\mathbf{m}$ and covariance matrix $K_{\mathbf{XX}}$

0 Mahalanobis distance

- Mean and covariance matrix define an ellipsoid

$$\mathcal{E}_A(\mathbf{m}) = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \mathbf{m})^T A(\mathbf{x} - \mathbf{m}) \leq 1\}, \quad \text{where } A = K_{\mathbf{XX}}^{-1}$$

0 Mahalanobis distance

- Mean and covariance matrix define an ellipsoid

$$\mathcal{E}_A(\mathbf{m}) = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \mathbf{m})^T A(\mathbf{x} - \mathbf{m}) \leq 1\}, \quad \text{where } A = K_{\mathbf{XX}}^{-1}$$

- Metric induced by the ellipsoid is known as *Mahalanobis distance*

$$d_M(\mathbf{x}, \mathbf{X}) = \sqrt{(\mathbf{x} - \mathbf{m})^T A(\mathbf{x} - \mathbf{m})}$$

0 Mahalanobis distance

- Mean and covariance matrix define an ellipsoid

$$\mathcal{E}_A(\mathbf{m}) = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \mathbf{m})^T A(\mathbf{x} - \mathbf{m}) \leq 1\}, \quad \text{where } A = K_{\mathbf{XX}}^{-1}$$

- Metric induced by the ellipsoid is known as *Mahalanobis distance*

$$d_M(\mathbf{x}, \mathbf{X}) = \sqrt{(\mathbf{x} - \mathbf{m})^T A(\mathbf{x} - \mathbf{m})}$$

- Main goal finding good *robust* approximations of the covariance matrix

0 Visualizations