1.11.5. Tree-based feature selection

XIAO LIYING

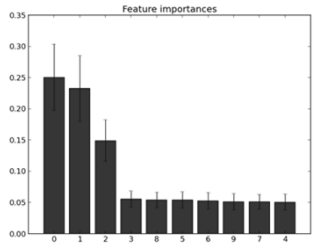
1.11.5. Tree-based feature selection

- Tree-based estimators can be used to compute feature importances, which in turn can be used to discard irrelevant features:
- >>>>>> from sklearn.ensemble import ExtraTreesClassifier
- >>> from sklearn.datasets import load_iris

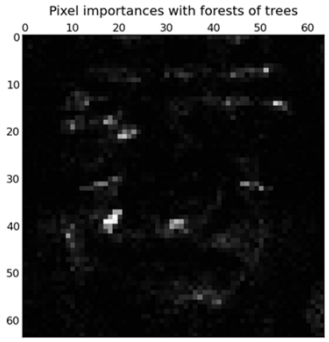
```
>>> iris = load_iris()
>>> X, y = iris.data, iris.target
>>> X.shape (150, 4)
>>> clf = ExtraTreesClassifier()
>>> X_new = clf.fit(X, y).transform(X) >>>
clf.feature_importances_ array([ 0.04..., 0.05..., 0.4..., 0.4...])
>>> X_new.shape (150, 2)
```

Feature importances with forests of trees

- This examples shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.



Pixel importances with a parallel forest of trees



1.11.6. Feature selection as part of a pipeline

- Feature selection is usually used as a pre-processing step before doing the actual learning.

```
>clf = Pipeline([ ('feature_selection',
>LinearSVC(penalty="l1")),
  ('classification', RandomForestClassifier()) ])
>clf.fit(X, y)
```

- You can perform similar operations with the other feature selection methods and also classifiers that provide a way to evaluate feature importances of course.

1.11.5. Tree-based feature selection

XIAO LIYING

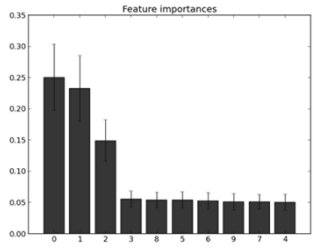
1.11.5. Tree-based feature selection

- Tree-based estimators can be used to compute feature importances, which in turn can be used to discard irrelevant features:
- >>>>>> from sklearn.ensemble import ExtraTreesClassifier
- >>> from sklearn.datasets import load_iris

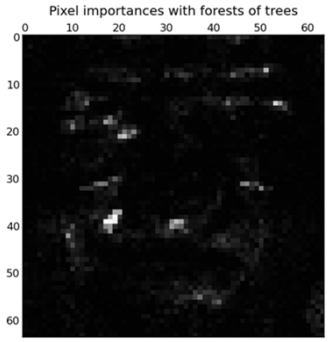
```
>>> iris = load_iris()
>>> X, y = iris.data, iris.target
>>> X.shape (150, 4)
>>> clf = ExtraTreesClassifier()
>>> X_new = clf.fit(X, y).transform(X) >>>
clf.feature_importances_ array([ 0.04..., 0.05..., 0.4..., 0.4...])
>>> X_new.shape (150, 2)
```

Feature importances with forests of trees

- This examples shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.



Pixel importances with a parallel forest of trees



1.11.6. Feature selection as part of a pipeline

- Feature selection is usually used as a pre-processing step before doing the actual learning.

```
>clf = Pipeline([ ('feature_selection',
>LinearSVC(penalty="l1")),
  ('classification', RandomForestClassifier()) ])
>clf.fit(X, y)
```

- You can perform similar operations with the other feature selection methods and also classifiers that provide a way to evaluate feature importances of course.

1.11.5. Tree-based feature selection

XIAO LIYING

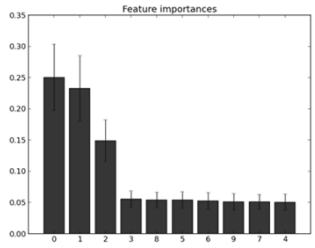
1.11.5. Tree-based feature selection

- Tree-based estimators can be used to compute feature importances, which in turn can be used to discard irrelevant features:
- >>>>>> from sklearn.ensemble import ExtraTreesClassifier
- >>> from sklearn.datasets import load_iris

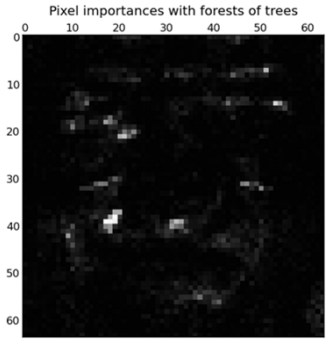
```
>>> iris = load_iris()
>>> X, y = iris.data, iris.target
>>> X.shape (150, 4)
>>> clf = ExtraTreesClassifier()
>>> X_new = clf.fit(X, y).transform(X) >>>
clf.feature_importances_ array([ 0.04..., 0.05..., 0.4..., 0.4...])
>>> X_new.shape (150, 2)
```

Feature importances with forests of trees

- This examples shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.



Pixel importances with a parallel forest of trees



1.11.6. Feature selection as part of a pipeline

- Feature selection is usually used as a pre-processing step before doing the actual learning.

```
>clf = Pipeline([ ('feature_selection',
>LinearSVC(penalty="l1")),
  ('classification', RandomForestClassifier()) ])
>clf.fit(X, y)
```

- You can perform similar operations with the other feature selection methods and also classifiers that provide a way to evaluate feature importances of course.

1.11.5. Tree-based feature selection

XIAO LIYING

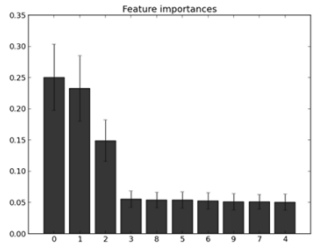
1.11.5. Tree-based feature selection

- Tree-based estimators can be used to compute feature importances, which in turn can be used to discard irrelevant features:
- >>>>>> from sklearn.ensemble import ExtraTreesClassifier
- >>> from sklearn.datasets import load_iris

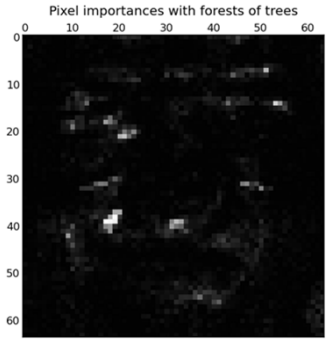
```
>>> iris = load_iris()
>>> X, y = iris.data, iris.target
>>> X.shape (150, 4)
>>> clf = ExtraTreesClassifier()
>>> X_new = clf.fit(X, y).transform(X) >>>
clf.feature_importances_ array([ 0.04..., 0.05..., 0.4..., 0.4...])
>>> X_new.shape (150, 2)
```

Feature importances with forests of trees

- This examples shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.



Pixel importances with a parallel forest of trees



1.11.6. Feature selection as part of a pipeline

- Feature selection is usually used as a pre-processing step before doing the actual learning.

```
>clf = Pipeline([ ('feature_selection',
>LinearSVC(penalty="l1")),
  ('classification', RandomForestClassifier()) ])
>clf.fit(X, y)
```

- You can perform similar operations with the other feature selection methods and also classifiers that provide a way to evaluate feature importances of course.

1.11.5. Tree-based feature selection

XIAO LIYING

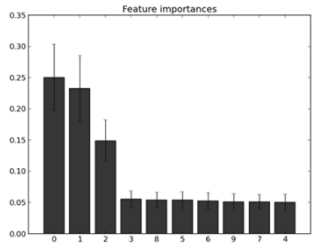
1.11.5. Tree-based feature selection

- Tree-based estimators can be used to compute feature importances, which in turn can be used to discard irrelevant features:
- >>>>>> from sklearn.ensemble import ExtraTreesClassifier
- >>> from sklearn.datasets import load_iris

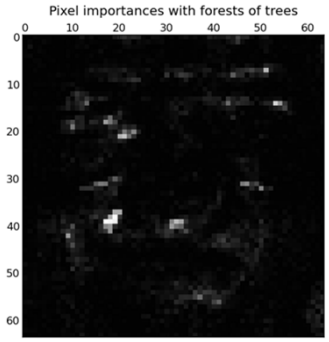
```
>>> iris = load_iris()
>>> X, y = iris.data, iris.target
>>> X.shape (150, 4)
>>> clf = ExtraTreesClassifier()
>>> X_new = clf.fit(X, y).transform(X) >>>
clf.feature_importances_ array([ 0.04..., 0.05..., 0.4..., 0.4...])
>>> X_new.shape (150, 2)
```

Feature importances with forests of trees

- This examples shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.



Pixel importances with a parallel forest of trees



1.11.6. Feature selection as part of a pipeline

- Feature selection is usually used as a pre-processing step before doing the actual learning.

```
>clf = Pipeline([ ('feature_selection',
>LinearSVC(penalty="l1")),
  ('classification', RandomForestClassifier()) ])
>clf.fit(X, y)
```

- You can perform similar operations with the other feature selection methods and also classifiers that provide a way to evaluate feature importances of course.

1.11.5. Tree-based feature selection

XIAO LIYING

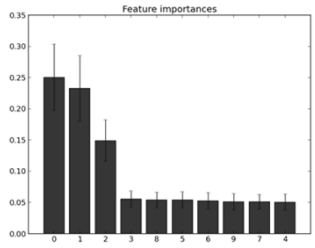
1.11.5. Tree-based feature selection

- Tree-based estimators can be used to compute feature importances, which in turn can be used to discard irrelevant features:
- >>>>>> from sklearn.ensemble import ExtraTreesClassifier
- >>> from sklearn.datasets import load_iris

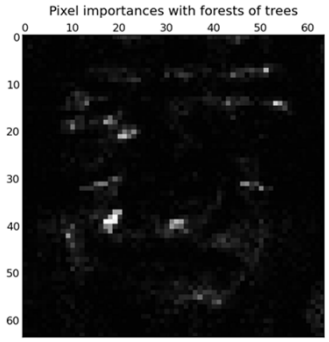
```
>>> iris = load_iris()
>>> X, y = iris.data, iris.target
>>> X.shape (150, 4)
>>> clf = ExtraTreesClassifier()
>>> X_new = clf.fit(X, y).transform(X) >>>
clf.feature_importances_ array([ 0.04..., 0.05..., 0.4..., 0.4...])
>>> X_new.shape (150, 2)
```

Feature importances with forests of trees

- This examples shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.



Pixel importances with a parallel forest of trees



1.11.6. Feature selection as part of a pipeline

- Feature selection is usually used as a pre-processing step before doing the actual learning.

```
>clf = Pipeline([ ('feature_selection',
>LinearSVC(penalty="l1")),
  ('classification', RandomForestClassifier()) ])
>clf.fit(X, y)
```

- You can perform similar operations with the other feature selection methods and also classifiers that provide a way to evaluate feature importances of course.