Semantic versioning for CVXPY

Developer Call - December 5, 2021

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	
Quasi-convex programming	
ECOS_BB removed	
DPP	
ECOS_BB reintroduced	
Power cones	
Verbose logging	
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	
ECOS_BB removed	
DPP	
ECOS_BB reintroduced	
Power cones	
Verbose logging	
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	1.0.21
ECOS_BB removed	
DPP	
ECOS_BB reintroduced	
Power cones	
Verbose logging	
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	1.0.21
ECOS_BB removed	1.1.0
DPP	1.1.0
ECOS_BB reintroduced	
Power cones	
Verbose logging	
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	1.0.21
ECOS_BB removed	1.1.0
DPP	1.1.0
ECOS_BB reintroduced	1.1.6
Power cones	
Verbose logging	
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	1.0.21
ECOS_BB removed	1.1.0
DPP	1.1.0
ECOS_BB reintroduced	1.1.6
Power cones	1.1.8
Verbose logging	
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	1.0.21
ECOS_BB removed	1.1.0
DPP	1.1.0
ECOS_BB reintroduced	1.1.6
Power cones	1.1.8
Verbose logging	1.1.11
Support for SCS 2.0	

The state of CVXPY's versioning

Good versioning should

1. help projects which have CVXPY as a dependency
2. signal to users when new features are introduced
3. Motivate developers and contributors

Change to CVXPY	Version
Reduction system	1.0.0
Geometric programming	1.0.11
Quasi-convex programming	1.0.21
ECOS_BB removed	1.1.0
DPP	1.1.0
ECOS_BB reintroduced	1.1.6
Power cones	1.1.8
Verbose logging	1.1.11
Support for SCS 2.0	1.1.17

Semantic versioning

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

https://semver.org/

Minor releases

- MUST be incremented if new, backwards compatible functionality is introduced to the public API.
- MUST be incremented if any public API functionality is marked as deprecated.
- MAY be incremented if substantial new functionality or improvements are introduced within the private code.
- MAY include patch level changes.

Moving forward

- we’ll need to collect more enhancements / features into a single minor release.
- We’ll need a release schedule.

NumPy is on a six-month cadence

“Bugfix releases (only the `z` changes in the `x.y.z` version number) have no new features; minor releases (the `y` increases) do.”

- NumPy 1.21.0 (release notes) – *22 Jun 2021*.
- NumPy 1.20.3 (release notes) – *10 May 2021*.
- NumPy 1.20.0 (release notes) – *30 Jan 2021*.
- NumPy 1.19.5 (release notes) – *5 Jan 2021*.
- NumPy 1.19.0 (release notes) – *20 Jun 2020*.
- NumPy 1.18.4 (release notes) – *3 May 2020*.
- NumPy 1.17.5 (release notes) – *1 Jan 2020*.
- NumPy 1.18.0 (release notes) – *22 Dec 2019*.
- NumPy 1.17.0 (release notes) – *26 Jul 2019*.
- NumPy 1.16.0 (release notes) – *14 Jan 2019*.
- NumPy 1.15.0 (release notes) – *23 Jul 2018*.
- NumPy 1.14.0 (release notes) – *7 Jan 2018*.

Development logistics

- Master is used as it is today
  - Accumulates all features and bugfixes
- Each minor release (e.g., 1.1, 1.2) gets a branch
  - Bugfixes are pulled into release branches as appropriate; triggers a patch.
- Tagging commits with versions
  - Do this on the release branches
  - No version tagging on master
- CI runs on master and supported release branches